

UACM

Universidad Autónoma
de la Ciudad de México

Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA
LICENCIATURA EN INGENIERÍA EN SISTEMAS
ELECTRÓNICOS Y DE TELECOMUNICACIONES

**“Algoritmos de súper-resolución
aplicados a imagen y video”**

TESIS

PARA OBTENER EL TÍTULO DE
**LICENCIADO EN INGENIERÍA EN SISTEMAS
ELECTRÓNICOS Y DE TELECOMUNICACIONES**

PRESENTA:

Minerva Liliana Osorio Mendoza

Director

Dr. Eduardo Ramos Díaz

Ciudad de México, abril 2018

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS ©

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

Resumen.

En este trabajo recepcional se describe la implementación de cinco algoritmos de interpolación en el contexto de la súper-resolución en imágenes y video. La interpolación es una etapa muy importante en el proceso de súper-resolución y tiene como objetivo incrementar la dimensión espacial de imágenes de baja resolución. Se estudian específicamente dos técnicas clásicas de interpolación, una del tipo polinomial y otra usando transformación de dominio. En el primer caso, se consideran las técnicas del vecino más cercano, bilineal, bicúbica y O-MOMS. En el segundo caso se considera la interpolación basada en la Transformada Coseno Discreta. El propósito de trabajo es analizar los métodos antes descritos con el fin de evaluar su desempeño para ser incorporados en el futuro en el proceso completo de súper-resolución. La evaluación de los algoritmos se realiza vía simulación para calcular la PSNR y el SSIM, las cuales sirven para comparar la calidad visual de las imágenes resultantes. Este trabajo, además de su carácter de trabajo recepcional, se realiza como investigación preliminar para realizar estudios de maestría en el campo de procesamiento de imágenes con súper-resolución.

AGRADECIMIENTOS.

Antes que nada le doy gracias a Dios por darme la oportunidad de concluir una más de mis metas.

También agradezco a la Universidad Autónoma de la Ciudad de México por permitirme formar parte de ella y darme las herramientas para prepararme profesionalmente, así como el apoyo para la impresión y empastado de esta tesis.

A mi director Dr. Eduardo Ramos Díaz por el apoyo que me brindó y los conocimientos compartidos para la realización de este trabajo.

Al Dr. Daniel Tapia Sánchez por la confianza y paciencia que me brindó durante mi formación académica. Gracias por compartir sus conocimientos durante este proceso, como en mi formación académica y el apoyo en la lectura y corrección del trabajo.

Al profesor Omar Nieto Crisóstomo, por los conocimientos compartidos y el apoyo en la lectura y corrección del trabajo.

Al profesor Sergio Iván Pérez Teniers, por el apoyo y corrección del trabajo.

Finalmente a cada uno de los profesores que formaron parte de mi formación académica, por su confianza, dedicación, consejos y paciencia que me brindaron.

DEDICATORIA.

A ti hermano Alfonso Osorio (papi), porque sé que desde donde estas me has guiado, protegido y cuidado, mil gracias porque por ti he llegado hasta aquí, por esos consejos y esas grandes pláticas que me diste y nunca lo olvidare siempre te llevare, en mi alma y mi corazón te amo.

Agradezco por la familia que tengo, porque por ellos he caminado con pasos fuertes y decisivos, han sabido guiarme en las decisiones que he tomado a lo largo de mi vida, por darme la suficiente fuerza para continuar en este largo camino.

A mis padres Micaela y Ernesto por ese amor infinito que me dan día a día, por el apoyo incondicional durante todo este tiempo, pero sobre todo por la confianza que depositaron en mi para lograr lo que tanto he anhelado, aunque no ha sido fácil el camino, pero con su apoyo lo he logrado, así que este triunfó considérelo suyo. Y nunca olviden que son mi mayor motivación para seguir en el camino, los amo y siempre están presentes en mi vida.

A mis hermanos y hermanas Ernesto, Araceli, Adriana, Jorge y Nahomi, por el cariño, el amor y paciencia, sin olvidar que en algún momentos hubo desconfianza por creer que no lo lograría, pero les demostré que a pesar de todo los tropiezos que hubieron en este camino lo logre y gracias a eso fue mi mayor reto el demostrar que si se pudo.

Y a todo aquellos amigos y amigas que me acompañaron en los mejores momentos y en los peores, gracias a los que se quedaron junto a mí, para seguir compartiendo este camino de logros, éxitos y lo que dios me ponga en el andar de la vida y a los que ya no están pero que me dieron lecciones importantes para saber que es una amistad y valorar, gracias que dios los llene de miles de bendiciones.

Donde ir.

Creía haber recorrido kilómetros de lava
cuando giro me doy cuenta,
sólo había caminado unos cuantos centímetros.

Los cuales me llevaron a estar
en donde estoy, ahora me toca subir.

Escalar
el llano de sangre.

Para llegar a la cima de lo esperado.

Y no me importa si recorro un metro de lava o medio kilómetro de sangre.

Oh, estar aquí,
oh, estar del otro lado del mundo.

Sólo es
cuestión de esperar
y llegar donde quiero estar.

Osme.

RESUMEN

AGRADECIMIENTOS

DEDICATORIA

CONTENIDO

INTRODUCCION	1
PLANTEAMIENTO DEL PROBLEMA	3
SOLUCIÓN PROPUESTA	3
OBJETIVOS	3
JUSTIFICACIÓN	4
ORGANIZACIÓN DEL DOCUMENTO	5
Capitulo 1. Estado del arte: Súper-resolución.....	6
1.1 El concepto de resolución.....	7
1.2 Imágenes de baja resolución e imágenes de alta resolución.....	9
1.3 Origen de la súper-resolución.	9
1.4 Súper-resolución e Interpolación.	12
1.5 El proceso de súper-resolución.....	14
Capitulo 2. Métodos de Interpolación	18
2.1 El proceso de Interpolación.	19
2.2 Interpolación ideal.	21
2.3 Convolución sobre una imagen.....	23
2.4 Interpolación polinomial y el kernel de interpolación.	24
2.5 Los coeficientes de interpolación.....	28
2.6 Técnicas de interpolación.	30
2.6.1 Técnica de interpolación “del vecino más cercano”	31
2.6.2 Técnica de Interpolación “Bilineal”	32
2.6.3 Técnica de interpolación Bicúbica.....	34
2.6.4 Técnica de interpolación O-MOMS.....	36

2.7	Técnicas de interpolación basadas en transformadas	36
2.7.1	La Transformada Coseno Discreta	36
2.7.2	Método de interpolación Transformada Coseno Discreta	38
Capítulo 3. Evaluación de algoritmos sobre imágenes.		40
3.1	Obtención de la imagen de baja resolución.	41
3.2	Imágenes de prueba.	45
3.3	Procedimiento de evaluación de los algoritmos.	47
3.3.1	Relación pico señal a ruido (PSNR).....	49
3.3.2	Índice de Similitud Estructural (SSIM).....	50
3.4	Evaluación de resultados.	51
Capítulo 4. Evaluación de algoritmos en secuencias de video.....		105
4.1	Procedimiento de evaluación.	106
4.2	Evaluación de los algoritmos sobre las regiones de interés.....	109
Capítulo 5. Conclusiones y trabajo futuro.		110
5.1	Conclusiones.....	111
5.2	Trabajo a futuro.....	111
Anexo 1. Códigos de programación ejecutados sobre MatLab.		113
REFERENCIAS		143

ÍNDICE DE FIGURAS

FIGURA 1.1 DIAGRAMA DE PROCESO PARA CAPTURAR UNA IMAGEN.....	10
FIGURA 1.2 EFECTOS NEGATIVOS DE LA INTERPOLACIÓN.....	14
FIGURA 1.3 PROCESO DE CAPTURA, RESTAURACIÓN Y SÚPER-RESOLUCIÓN.....	15
FIGURA 1.4 TRANSFORMACIÓN DE UNA IMAGEN DE BAJA RESOLUCIÓN EN UNA IMAGEN DE ALTA RESOLUCIÓN USANDO UNA REJILLA DE PROYECCIÓN.	16
FIGURA 1.5 IMAGEN COMPUESTA QUE MUESTRA UNA ESTRUCTURA SEMI-UNIFORME.	16
FIGURA 2.1 EJEMPLO DE INTERPOLACIÓN DE UNA SEÑAL.....	19
FIGURA 2.2 USO DE LA INTERPOLACIÓN EN IMÁGENES DIGITALES.....	20
FIGURA 2.3 DISTANCIA DE INTERPOLACIÓN A PARTIR DE PÍXELES CONOCIDOS.	20
FIGURA 2.4 FUNCIÓN SINC	22
FIGURA 2.5 CONVOLUCIÓN APLICADA AL PÍXEL (2,4).....	24
FIGURA 2.6 FUNCIÓN BASE $\beta_0(x)$	26
FIGURA 2.7 FUNCIÓN BASE $\beta_1x = \beta_0x * \beta_0(x)$	26
FIGURA 2.8 FUNCIÓN BASE $\beta_2x = \beta_1x * \beta_0(x)$	27
FIGURA 2.9 FUNCIÓN BASE $\beta_3x = \beta_2x * \beta_0(x)$	27
FIGURA 2.10 MÉTODO DE REJILLA.....	29
FIGURA 2.11 MODELO GENERAL DE INTERPOLACIÓN.	30
FIGURA 2.12 TÉCNICA DE INTERPOLACIÓN DEL VECINO MÁS CERCANO.	31
FIGURA 2.13 TÉCNICA DE INTERPOLACIÓN BILINEAL.	33
FIGURA 2.14 MATRIZ DE 8X8 CON LOS 64 VALORES BASE PARA LA TRANSFORMADA COSENO DISCRETO.	37
FIGURA 2.15 PROCESO DE INTERPOLACIÓN DCT.	39
FIGURA 3.1 DIAGRAMAS DE FLUJO CON EL PROCEDIMIENTO DE INTERPOLACIÓN Y EVALUACIÓN DE LOS ALGORITMOS.	41
FIGURA 3.2 PROCESO DE OBTENCIÓN DE LA IMAGEN DE BAJA RESOLUCIÓN (LR).....	42
FIGURA 3.3 PROCESO DE OBTENCIÓN DE LA IMAGEN DE BAJA RESOLUCIÓN (LR) A PARTIR DE UNA IMAGEN DE PRUEBA DE ALTA RESOLUCIÓN (HR).	43
FIGURA 3.4 DEMOSTRACIÓN GRÁFICA DEL MÉTODO DE OBTENCIÓN DE LA IMAGEN LR.	44
FIGURA 3.5 IMÁGENES DE PRUEBA.	47
FIGURA 3.6 PROCEDIMIENTO DE EVALUACIÓN DE LOS ALGORITMOS.....	48
FIGURA 3.7 ESTIMACIÓN MEDIANTE EL MÉTODO DEL VECINO MÁS CERCANO (1:2).....	54
FIGURA 3.8 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BILINEAL (1:2).....	55
FIGURA 3.9 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BICÚBICA (1:2).....	56
FIGURA 3.10 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN O-MOMS (1:2).	57
FIGURA 3.11 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN DCT (1:2).....	58
FIGURA 3.12 ESTIMACIÓN MEDIANTE EL MÉTODO DEL VECINO MÁS CERCANO (1:2).	59
FIGURA 3.13 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BILINEAL (1:2).	60
FIGURA 3.14 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BICÚBICA (1:2)	61
FIGURA 3.15 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN O-MOMS (1:2).	62
FIGURA 3.16 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN DCT (1:2).....	63
FIGURA 3.17 ESTIMACIÓN MEDIANTE EL MÉTODO DE VECINO MÁS CERCANO (1:2)	64
FIGURA 3.18 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BILINEAL (1:2).	65

FIGURA 3.19 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BICÚBICA (1:2).	66
FIGURA 3.20 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN O-MOMS (1:2).	67
FIGURA 3.21 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN DCT (1:2).	68
FIGURA 3.22 ESTIMACIÓN MEDIANTE EL MÉTODO DEL VECINO MÁS CERCANO (1:2).	69
FIGURA 3.23 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BILINEAL (1:2).	70
FIGURA 3.24 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BICÚBICA (1:2).	71
FIGURA 3.25 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN O-MOMS (1:2).	72
FIGURA 3.26 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN DCT (1:2).	73
FIGURA 3.27 ESTIMACIÓN MEDIANTE EL MÉTODO DEL VECINO MÁS CERCANO (1:2).	74
FIGURA 3.28 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BILINEAL (1:2).	75
FIGURA 3.29 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BICÚBICA (1:2).	76
FIGURA 3.30 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN O-MOMS (1:2).	77
FIGURA 3.31 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN DCT (1:2).	78
FIGURA 3.32 ESTIMACIÓN MEDIANTE EL MÉTODO DEL VECINO MÁS CERCANO (1:4).	80
FIGURA 3.33 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BILINEAL (1:4).	81
FIGURA 3.34 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BICÚBICA (1:4).	82
FIGURA 3.35 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN O-MOMS (1:4).	83
FIGURA 3.36 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN DCT (1:4).	84
FIGURA 3.37 ESTIMACIÓN MEDIANTE EL MÉTODO DEL VECINO MÁS CERCANO (1:4).	85
FIGURA 3.38 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BILINEAL (1:4).	86
FIGURA 3.39 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BICÚBICA (1:4).	87
FIGURA 3.40 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN O-MOMS (1:4).	88
FIGURA 3.41 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN DCT (1:4).	89
FIGURA 3.42 ESTIMACIÓN MEDIANTE EL MÉTODO DEL VECINO MÁS CERCANO (1:4).	90
FIGURA 3.43 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BILINEAL (1:4).	91
FIGURA 3.44 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BICÚBICA (1:4).	92
FIGURA 3.45 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN O-MOMS (1:4).	93
FIGURA 3.46 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN DCT (1:4).	94
FIGURA 3.47 ESTIMACIÓN MEDIANTE EL MÉTODO DEL VECINO MÁS CERCANO (1:4).	95
FIGURA 3.48 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BILINEAL (1:4).	96
FIGURA 3.49 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BICÚBICA (1:4).	97
FIGURA 3.50 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN O-MOMS (1:4).	98
FIGURA 3.51 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN DCT (1:4).	99
FIGURA 3.52 ESTIMACIÓN MEDIANTE EL MÉTODO DEL VECINO MÁS CERCANO (1:4).	100
FIGURA 3.53 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BILINEAL (1:4).	101
FIGURA 3.54 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN BICÚBICA (1:4).	102
FIGURA 3.55 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN O-MOMS (1:4).	103
FIGURA 3.56 ESTIMACIÓN MEDIANTE EL MÉTODO DE INTERPOLACIÓN DCT (1:4).	104
FIGURA 4.1 TRAMAS DEL VIDEO PARA LAS PRUEBAS	106
FIGURA 4.2 INCREMENTO DE LA DIMENSIÓN ESPACIAL DE LA ESCENA COMPLETA.	107
FIGURA 4.3 PROCESO DE EXTRACCIÓN DE LA REGIÓN DE INTERÉS PARA SU INTERPOLACIÓN.	108
FIGURA 4.4 MÉTODO DEL VECINO MÁS CERCANO SOBRE LA REGIÓN DE INTERÉS DE LA PRIMERA IMAGEN DE PRUEBA.	109

ÍNDICE DE TABLAS

TABLA 3.1 CONCENTRADO DE VALORES PSNR Y SSIM PARA AUMENTAR EL DOBLE DE RESOLUCIÓN DE CADA MÉTODO	52
TABLA 3.2 CONCENTRADO DE VALORES PSNR Y SSIM PARA AUMENTAR EL CUÁDRUPLE DE RESOLUCIÓN DE CADA MÉTODO.....	79

INTRODUCCION

Desde la invención de la computadora, ha existido un notable interés por procesar, transmitir y mostrar imágenes de mejor calidad a través de una gran diversidad de sistemas electrónicos y de telecomunicaciones. La mejora en la capacidad de procesamiento y visualización de imágenes en diversos sistemas electrónicos representa un tema de especial interés para la investigación, lo cual ha sido motivado en gran parte por el hecho de que la mayoría de la información que utiliza el ser humano es visual. En efecto, el campo del procesamiento de imágenes es bastante vasto en relación a métodos y algoritmos, los cuales se han aplicado a una gran variedad de áreas, tales como la medicina, ciencias genómicas, visión artificial, inspección industrial y videos, entre otras. En el contexto del procesamiento de imágenes, existen aplicaciones tales como video-vigilancia, sensores médicos, radares, televisiones de alta definición, percepción remota, microscopios electrónicos, rayos x, entre otras, para las cuales resulta indispensable generar imágenes con una alta resolución, dado que una mayor resolución conduce a una mayor exactitud en el análisis visual de la imagen. Sin embargo, el tamaño de una imagen es proporcional a su resolución y depende en gran medida de la calidad del dispositivo mediante el cual la imagen es adquirida. Pero no únicamente que la resolución de la imagen es importante, también lo es la velocidad con que esta es adquirida. El tamaño y la velocidad con que se procesan las imágenes están determinados en primer lugar por el dispositivo de captura, típicamente un dispositivo microelectrónica que realiza la función de sensor de imagen. Uno de los sensores más comunes es el CCD (Dispositivo de cargas eléctricas interconectadas), el cual está formado por un conjunto de diminutas células fotoeléctricas llamados fotositos que capturan la imagen bidimensional. La capacidad de resolución dependerá del número de células que tenga el sensor. Un alto número de fotositos posibilita una resolución espacial alta de los sistemas de imagen. La frecuencia de muestreo determina las componentes de alta frecuencia (detalles de la imagen) que pueden ser captadas. Esto es, a menor resolución del sensor menor será la frecuencia de

muestreo y menor será el número de componentes de alta frecuencia que puedan ser captadas.

A pesar de que en la actualidad es posible diseñar dispositivos de captura de gran capacidad, no siempre pueden ser integrados en todas las aplicaciones debido a restricciones físicas del sistema o aplicación. Por otra parte, en la mayoría de los casos, el ancho de banda de sistema de comunicación que conecta el dispositivo de captura con la unidad de procesamiento es insuficiente para transmitir el tamaño de las imágenes a la velocidad de captura, puesto que por lo general la velocidad del canal es muy lento en comparación con la velocidad a la cual se captura la información. Por esta razón, resulta conveniente adquirir imágenes de baja resolución y transmitir las sin mayores requerimientos de ancho de banda a la unidad de procesamiento, y entonces, mediante el procesamiento adecuado, transformarlas en imágenes equivalentes de alta resolución.

Las técnicas de súper-resolución realizan un proceso que permite reconstruir información faltante una imagen, y son empleadas en aquellos casos donde se desea obtener una imagen de mayor nitidez y resolución a partir de un conjunto de imágenes de baja resolución y poco nítidas, degradadas por la diferencia de velocidad entre el objeto a capturar en pantalla y la velocidad del obturador de la cámara.

Otra interesante aplicación, es tener la capacidad de transmitir un video con baja calidad y al recibirlo tener la posibilidad de reproducirlo con una mejor definición, si esto puede hacer eficientemente, se optimizaría la velocidad de transmisión. Además, en las aplicaciones mencionadas anteriormente, el poder contar con imágenes de mayor resolución permite realizar análisis visuales más objetivos por parte de los especialistas.

PLANTEAMIENTO DEL PROBLEMA

El problema que se plantea en este trabajo es el siguiente: ¿Cómo mejorar la resolución de imágenes obtenidas a través de un dispositivo de captura de baja resolución?

SOLUCIÓN PROPUESTA

Para contribuir al estado del arte que aborda el problema de la baja resolución de imágenes y video, se propone comprobar la efectividad de algoritmos de súper-resolución. Específicamente, el propósito es evaluar la efectividad de los algoritmos para recuperar los píxeles perdidos durante la adquisición, almacenamiento o transmisión de imágenes, con el fin de obtener una representación de más alta resolución que pueda ser interpretada de manera óptima, y que al ser analizada por terceros, pueda ser visualizada como si ésta hubiera sido capturada mediante un dispositivo de súper-resolución.

OBJETIVOS

El objetivo general es implementar la segunda etapa del proceso de súper-resolución, evaluando y comparando la capacidad para mejorar la dimensión espacial en imágenes y video de cinco de los principales métodos de interpolación, considerándolos índices de calidad correspondientes.

Objetivos específicos:

1. Adquirir el conocimiento del proceso de súper-resolución en imágenes y video y sus técnicas principales.
2. Comprender sus principales modelos de interpolación polinomial y con transformación de dominio.

3. Implementar los algoritmos de interpolación usando imágenes de prueba y secuencias de videos reales.
4. Evaluar la calidad de los resultados obtenidos con los diferentes algoritmos mediante los índices de calidad correspondientes.

JUSTIFICACIÓN

La propuesta de enfocar la atención del trabajo solamente a la etapa de interpolación se hace con el fin de lograr resultados concretos en el tiempo destinado a la realización de la tesis, mismos que puedan servir como fundamento para trabajar en un futuro cercano sobre el proceso completo de resolución durante mis estudios de maestría. La razón de acudir a la simulación para verificar el desempeño de los algoritmos obedece a que la idea principal del trabajo se orienta a comparar la mayor cantidad de algoritmos más que a su realización en tiempo real, lo cual será considerado también dentro del trabajo a futuro. Sin embargo, no por ello el trabajo desarrollado en esta tesis pierde relevancia, ya que generalmente el tema de la súper-resolución se estudia a partir del nivel de maestría debido a la alta complejidad de las técnicas actuales utilizadas. Más bien, se pretende que la investigación realizada sirva para comprender los fundamentos de la súper-resolución y desarrollar una mayor habilidad en la codificación de las técnicas y métodos utilizados en el campo. Por otra parte, la medición de la calidad en imágenes representa por sí misma un reto muy grande, por lo que el trabajo también servirá para comprender el funcionamiento de dos de los índices de calidad más comunes actualmente.

ORGANIZACIÓN DEL DOCUMENTO

El trabajo se encuentra dividido en cinco capítulos. En el primer capítulo se presenta el estado del arte, el cual sirve al mismo tiempo de marco para describir el concepto de la súper-resolución, el proceso de súper-resolución, sus fundamentos y aplicaciones.

El segundo capítulo se enfoca en el proceso de interpolación, se describen sus fundamentos matemáticos y se explica el funcionamiento de las técnicas clásicas. También se introduce el concepto de kernel de interpolación, el cual sirve para describir los métodos de interpolación considerados en el trabajo.

El tercer capítulo describe los resultados obtenidos considerando imágenes de prueba usadas comúnmente en el campo del procesamiento de imágenes. Los resultados se presentan en la forma de la tabla con los valores PSNR y SSIM obtenidos para cada método, así como también en forma gráfica mediante las imágenes obtenidas en cada simulación.

El cuarto capítulo es similar al capítulo anterior, excepto por dos aspectos. Primero, el hecho de que los algoritmos se evalúan considerando ahora imágenes reales provenientes de cámaras de video y segundo, que los algoritmos no se aplican sobre la totalidad de píxeles de la imagen, sino únicamente sobre cierta región de interés. En este caso, debido a que no se cuenta con ninguna imagen de referencia, no es posible calcular ninguna métrica de calidad, por lo que la evaluación de los algoritmos se realiza en forma más subjetiva mediante su apreciación visual.

Finalmente, en el quinto capítulo se concluye el trabajo y se mencionan las tareas pendientes para realizar.

Capítulo

1

Estado del arte: Súper-resolución

El propósito de este capítulo es exponer las generalidades de la técnica de súper-resolución. Para comprender la diferencia entre una imagen de baja resolución y una de alta resolución, resulta imprescindible comprender en primer lugar el concepto de resolución en términos de la cantidad de información contenida en sus componentes espaciales. Esto nos servirá para describir la forma en la cual una técnica de súper-resolución puede mejorar la resolución de una imagen a partir de la escasa información contenida en una imagen de baja resolución. Finalmente, se presentan los principales métodos usados para implementar la súper-resolución en términos de sus modelos matemáticos, los cuales servirán para detallar su implementación en el capítulo siguiente.

1.1 El concepto de resolución.

El término **resolución**¹ se define como el nivel más pequeño de detalle que puede ser medido o distinguido en una representación visual [1] y está relacionado con los conceptos de nitidez y calidad visual. En el caso digital, la resolución está determinada por número de píxeles distintos que tiene una imagen por unidad de longitud. En efecto, una de las unidades de medida más utilizada es la del número de píxeles por pulgada (ppp), parámetro que nos permite cuantificar la calidad de una imagen en relación con la capacidad de discriminación del observador. Algunos otros parámetros como el desenfoque, nivel de detalle y nivel de nitidez, entre otros, son analizados usando esta medida. En resumen, se puede decir que a mayor resolución de la imagen, mayor número de detalle y calidad visual de la misma.

La resolución puede estudiarse desde dos perspectivas distintas: desde el campo de la óptica y desde el punto de vista del procesamiento de señales. En óptica, los investigadores la definen en términos de una función de transferencia de modulación (MTF), la cual está definida como el módulo de la función de transferencia (OTF) óptica. La OTF representa los efectos de la pérdida por difracción debidos a la geometría circular de la pupila del ojo. En este caso, las limitaciones en la resolución se deben al espacio finito de la pupila a través del cual se concentra la cantidad de luz incidente. La MTF [22,25] se utiliza no solamente para determinar el límite de resolución en un punto dado, sino también para caracterizar la respuesta del sistema óptico para una entrada arbitraria. Esta misma idea se utilizará más adelante para cuando se describan los métodos de interpolación algunos métodos de interpolación en términos de la convolución entre la imagen capturada de baja resolución y un kernel de interpolación, el cual, en el dominio de la frecuencia, representa precisamente la MTF.

Por otra parte, en el campo del procesamiento digital de imágenes y visión artificial, la resolución se utiliza de tres diferentes formas [24]:

¹ El concepto de resolución resulta muy importante para determinar la calidad de la impresión de una imagen digital. A mayor resolución de una imagen digital mejor es la calidad de dicha imagen.

a) La **resolución espacial** se refiere a la distribución espacial de los píxeles por unidad de área y es medido en píxeles por pulgada (ppp). Entre mayor sea la resolución espacial, mayor será el número de píxeles, lo cual permite lograr un mayor nivel de detalle y transiciones muy sutiles entre los colores de la imagen. La resolución espacial de un dispositivo de visualización se expresa por lo general en puntos por pulgada (dpi) y se refiere al tamaño de los puntos individuales creados por el dispositivo.

b) La **resolución de brillantez** se refiere a la cantidad de niveles de brillo que pueden ser registrados para un píxel determinado. Esto se relaciona con la cuantización de la energía luminosa contenida en un elemento de dispositivo de carga acoplada (CCD). Un nombre más apropiado para la resolución de brillantez es el de nivel de cuantización. La resolución de brillantez para una imagen que usa dos niveles (1 bit por píxel) o en escala de grises es comúnmente de 256, lo que implica representar cada nivel usando 8 bits. Para imágenes a color, se requieren al menos 24 bits para representar cada nivel, es decir, 8 bits para cada plano RGB.

c) La **resolución temporal** se utiliza en el procesamiento de video y se refiere al número de cuadros capturados por segundo y se conoce también como tasa de tramas. Está relacionada con la cantidad de movimiento perceptible entre tramas. Mayores tasas de tramas resultan en menos distorsión debida a movimientos en la escena. El límite inferior de la resolución temporal es directamente proporcional al movimiento esperado durante dos tramas consecutivas. El valor típico nunca debe ser menor de 25 tramas por segundo [7].

En este trabajo, el término resolución se refiere específicamente a la resolución espacial y por lo tanto, el proceso de obtener una imagen de mayor resolución a partir de observaciones de baja resolución espacial, es a lo que se refiere la súper-resolución.

1.2 Imágenes de baja resolución e imágenes de alta resolución.

Una forma común de clasificar las imágenes a partir de su resolución es aquella que las divide en imágenes de alta resolución (HR) e imágenes de baja resolución (LR). Una imagen de alta resolución está prevista para la impresión, teniendo generalmente 300ppp o más. Una imagen de baja resolución está prevista solamente para su exhibición en pantalla, teniendo generalmente una resolución de 100ppp o menos.

Una impresión fotográfica de 200ppp podría ser aceptable, sin embargo, es totalmente necesario alcanzar por lo menos 300ppp debido a factores presentes durante el proceso de impresión que pueden reducir la calidad de la imagen, tales como la óptica de la cámara, la limpieza de la lente o el procesador de imágenes de la cámara digital. Para lograr una resolución de 300ppp, se requiere imprimir 17.3×17.3 ppp. Por ejemplo, en caso de tener una imagen de 128×128 pixeles, el tamaño máximo de la impresión en alta resolución debería ser de 18.5 cm^2 , debido a que $128/17.3=7.3$ pulgadas, lo que equivale 18.5 cm^2 . Si se requiere entonces aumentar el tamaño de impresión de una imagen sin afectar su resolución, es necesario utilizar algún algoritmo de súper-resolución.

1.3 Origen de la súper-resolución.

Como se ha comentado en secciones anteriores, la súper-resolución busca representar detalles que en un principio no son apreciables en las imágenes originales captadas mediante una cámara fotográfica o de video digital. Para comprender las razones por las cuales la mayoría de dispositivos de captura comerciales no pueden producir imágenes con la calidad requerida, es necesario explicar el proceso de adquisición de la imagen y las restricciones físicas y de ingeniería presentes durante el proceso [10].

La captura de una imagen de una escena del mundo real sigue los siguientes pasos:

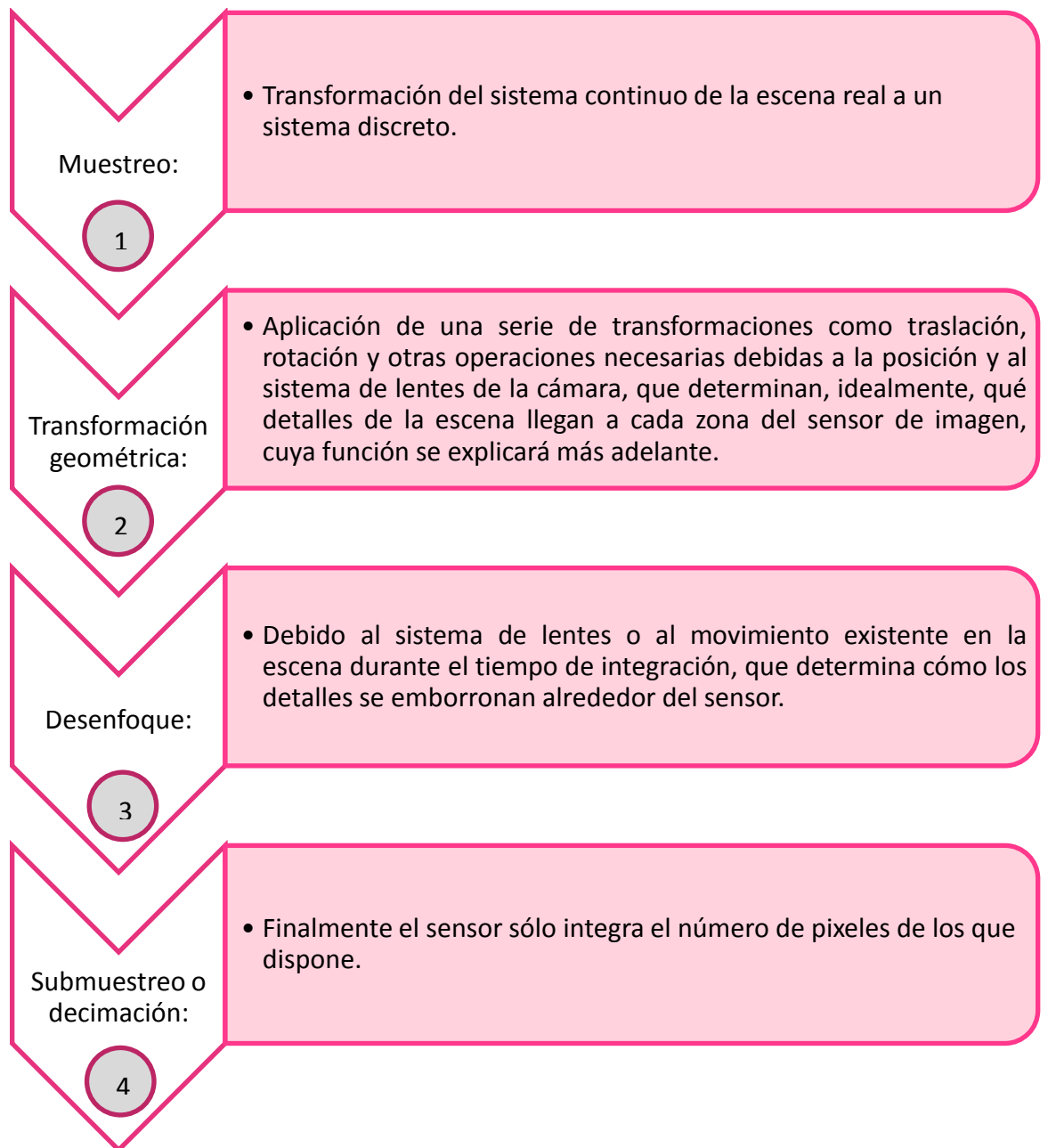


Figura 1.1 Diagrama de proceso para capturar una imagen.

El sensor de imagen es el elemento más importante de una cámara digital o de video. Se trata de un circuito integrado formado por millones de componentes sensibles a la luz que al ser expuestos capturan la luz proyectada por un objetivo. Durante el proceso de captura, los detalles de la escena son integrados por los diferentes píxeles del sensor, de manera que en cada captura cada píxel recoge una

información diferente. Por ello, algunas técnicas de súper-resolución se basan en tratar de encontrar la relación entre que pixeles de diferente captura se han muestreado los distintos detalles de la escena.

Los sensores de imagen modernos están basados en el principio de los dispositivos de cargas acopladas (CCD) [25], los cuales responden a fuentes de luz. Un sensor con una alta densidad de foto-detectores captura imágenes con una alta resolución espacial, pero un sensor con un menor número de foto-detectores produce una imagen de baja resolución, generando un fenómeno de pixelización, el cual puede ser observado incluso a simple vista. Esto se puede explicar a partir del teorema del muestreo, a partir del cual la resolución espacial está limitada por la tasa de muestreo espacial, es decir, el número de foto-detectores por unidad de longitud a lo largo de una dirección particular. Otro factor que limita la resolución es el tamaño del foto detector. Al respecto, se podría pensar que reduciendo el tamaño del foto detector se podría incrementar la cantidad de pixeles, no obstante, conforme la dimensión del pixel se reduce, la calidad de la imagen también se ve degradada debido al aumento del ruido impulsivo. Se ha estimado que el tamaño mínimo de un foto detector debería ser aproximadamente $50 \mu m^2$ y desde hace tiempo los dispositivos CCD comerciales ya han superado este límite. Por otra parte, si la tasa de muestreo es demasiado baja, la imagen se ve distorsionada debido al efecto escalonado.

La solución más simple para mejorar la resolución espacial es reducir el tamaño del pixel, es decir, incrementar el número de pixeles por unidad de área. Sin embargo, al reducir el tamaño del pixel, la cantidad de luz también se reduce la cantidad de luz disponible, lo cual genera ruido impulsivo, el cual degrada la calidad de la imagen severamente. Para reducir el tamaño del pixel sin sufrir los efectos del ruido impulsivo, existe una limitación en la reducción del tamaño del pixel, el cual se ha estimado como 40 mm^2 .

Otra forma de mejorar la resolución espacial aumentando el tamaño del circuito integrado en el dispositivo de captura, lo que ocasionaría un incremento en la capacitancia. Debido a que una mayor capacitancia hace que sea más difícil incrementar la tasa de transferencia de carga, este enfoque no se considera

adecuado debido a la degradación en la resolución temporal. Por lo tanto, es necesario un enfoque alternativo para mejorar la resolución espacial que no dependa de las limitaciones sobre los sensores ni de los dispositivos de captura [10].

El enfoque más adecuado es utilizar técnicas de procesamiento de señales para generar una imagen de alta resolución (HR) a partir de una o más imágenes de baja resolución (LR). Sin embargo, es importante mencionar que la simple acción de aumentar el tamaño de una imagen no necesariamente corresponde a mejorar su resolución. De hecho, el redimensionamiento de la imagen debe estar acompañado de aproximaciones a frecuencias mayores que las representadas en el tamaño original, así como una mayor relación señal a ruido, lo cual se logra a través de técnicas de sobre-muestreo tales como el uso de funciones de interpolación. Recientemente, esta ha sido una de las áreas de investigación más activas en el campo del procesamiento de imágenes y se ha desarrollado bajo el nombre de súper-resolución. La principal ventaja de esta técnica es su bajo costo y su capacidad de ser aplicada a todo tipo de imágenes digitales. Una de las aplicaciones más comunes es la reconstrucción de imágenes de alta calidad a partir de imágenes LR tomadas por una cámara de bajo costo. Otra aplicación es el acercamiento sintético de una región de interés (ROI), el cual es muy importante en áreas de vigilancia, medicina, ciencias forenses e imágenes satelitales. Para vigilancia o propósitos forenses. Son precisamente estas dos aplicaciones las que se utilizarán en tercer capítulo de este trabajo para demostrar el desempeño de los algoritmos desarrollados en el capítulo dos.

El objetivo ideal de la súper-resolución es regenerar la escena continua del mundo real de donde se ha capturado la imagen. Sin embargo, dado que este objetivo resulta inalcanzable todavía con la tecnología y técnicas actuales, lo que se espera es poder regenerar una escena discretizada intermedia con mayor resolución.

1.4 Súper-resolución e Interpolación.

En el proceso de obtención de imágenes de alta resolución existen métodos, los cuales se encuentran estrechamente relacionados y se complementan entre sí para lograr el objetivo principal de la súper-resolución, el que además de transformar la

imagen de baja resolución en una de alta resolución, se preserven las características visuales de la imagen original, más allá de simplemente incrementar su dimensión espacial. Sin embargo, ambos métodos tienen su propia finalidad en el proceso. La diferencia más importante entre los métodos, es que la interpolación [9] se utiliza específicamente para sobre-muestrear la imagen de baja resolución para mejorar su dimensión espacial. Debido a que el proceso de sobre-muestreo utilizado por la interpolación no es capaz de recuperar los componentes de alta frecuencia de la imagen, la estimación obtenida puede presentar cierta degradación causada principalmente por aliasing en los bordes de la imagen o puede ser que esta se vea ligeramente borrosa o difuminada [4]. Por ejemplo, la figura 1.2a muestra una imagen de baja resolución y su equivalente con una mayor resolución espacial usando dos diferentes métodos de interpolación. La imagen mostrada en la figura 1.2b presenta visualmente el efecto de aliasing y la imagen 1.2c se ve borrosa. Estos son dos de los efectos no deseables introducidos durante el proceso de interpolación.

Por otra parte, la súper-resolución incluye, además de procesos de sobre-muestro (interpolación), otros procedimientos para eliminar los efectos negativos que provocan la degradación de la imagen y preservar las características visuales de la imagen original [6]. La súper-resolución también se utiliza en la literatura para identificar algunos métodos de interpolación los cuales utilizan una secuencia de imágenes en lugar de una sola imagen. Estos métodos aprovechan la información distribuida en las distintas imágenes para estimar los valores óptimos y mejorar la resolución espacial. En la siguiente sección se ilustra esta variante de la interpolación.



Figura 1.2 Efectos negativos de la interpolación.

1.5 El proceso de súper-resolución.

El proceso de captura de la imagen produce una representación discreta de una escena continua. Esta discretización, tanto en espacio como intensidad, es un proceso de muestreo que introduce un efecto denominado técnicamente *aliasing*, el cual se muestra visiblemente como discontinuidades en forma de sierra o escalón sobre los bordes y líneas de la imagen, esto debido a la información que se pierde en frecuencias por encima de la tasa de Nyquist. Es precisamente esta pérdida de información la que se busca reconstruir mediante las técnicas de súper-resolución. Actualmente existen distintos métodos y técnicas para lograrlo los cuales pueden clasificarse en dos grandes categorías: algoritmos en el dominio del tiempo y algoritmos en el dominio de frecuencia. Dentro de estos, existen algoritmos que requieren de una sola imagen de baja resolución y otros que requieren de múltiples tomas de baja resolución de la misma escena. En este caso, la tesis describe métodos en el dominio del tiempo sobre una sola imagen de baja resolución.

Los procesos de captura de la imagen y el de súper-resolución puede resumirse en tres diferentes etapas, las cuales se muestran en la figura 1.3. La primera etapa

corresponde al proceso de captura descrito en la sección 1.3. La segunda etapa depende de funciones específicas del modelo de la cámara y su capacidad para mejorar la calidad de la imagen. Hasta este punto la resolución de la imagen está en relación con las características del sensor de imagen. La etapa 3 corresponde a la etapa donde se aplican los métodos de súper-resolución para compensar la pérdida de información de la imagen de baja resolución.

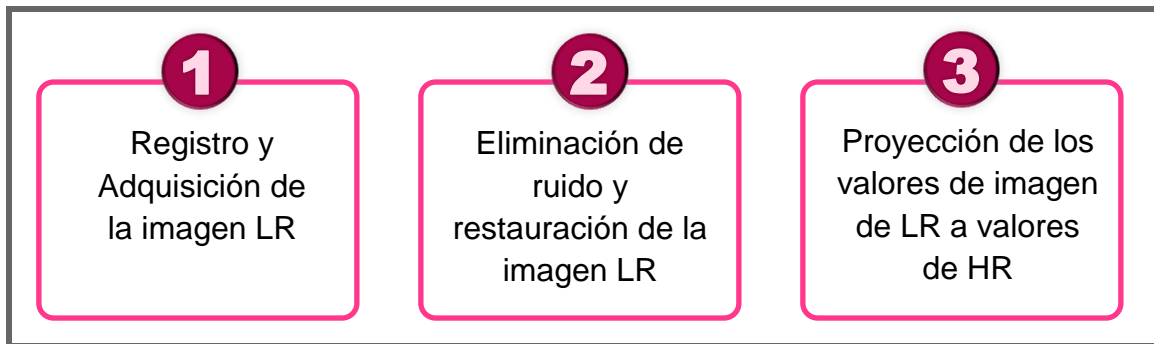


Figura 1.3 Proceso de Captura, restauración y súper-resolución

Una vez que la imagen ha sido registrada y pre-procesada para reducir el ruido adherido inevitablemente durante el proceso de captura, es posible aplicar alguna técnica de súper-resolución para incrementar y mejorar su dimensión espacial. Para realizar esta labor existen diversos métodos, pero casi todos ellos emplean un procedimiento en común que consiste en proyectar los pixeles disponibles en la imagen de baja resolución de dimensión $M \times N$ sobre una “rejilla” vacía de dimensión $nM \times nN$, la cual claramente tendrá una resolución espacial n veces mayor que la imagen registrada mediante el dispositivo de captura [17]. Una vez proyectados los valores, el siguiente paso es completar estimar la información faltante usando una operación matemática en particular del método de súper-resolución empleado. Este proceso se ilustra en la figura 1.4.

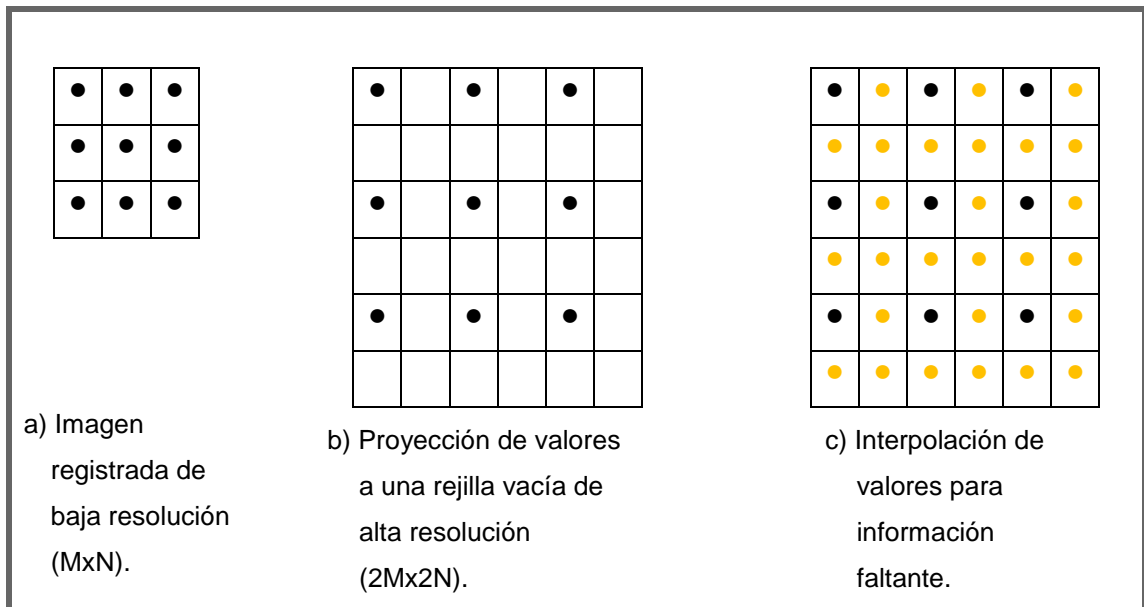


Figura 1.4 Transformación de una imagen de baja resolución en una imagen de alta resolución usando una rejilla de proyección.

El proceso descrito en la figura 1.4 se conoce comúnmente como interpolación. En el caso de la súper-resolución a partir de múltiples imágenes consecutivas de una misma escena, el proceso es similar, aunque la imagen resultante ya no es muestreada sobre una rejilla rectangular uniforme, sino que debido al movimiento tiene una estructura semi-uniforme, como se puede apreciar en la figura 1.5

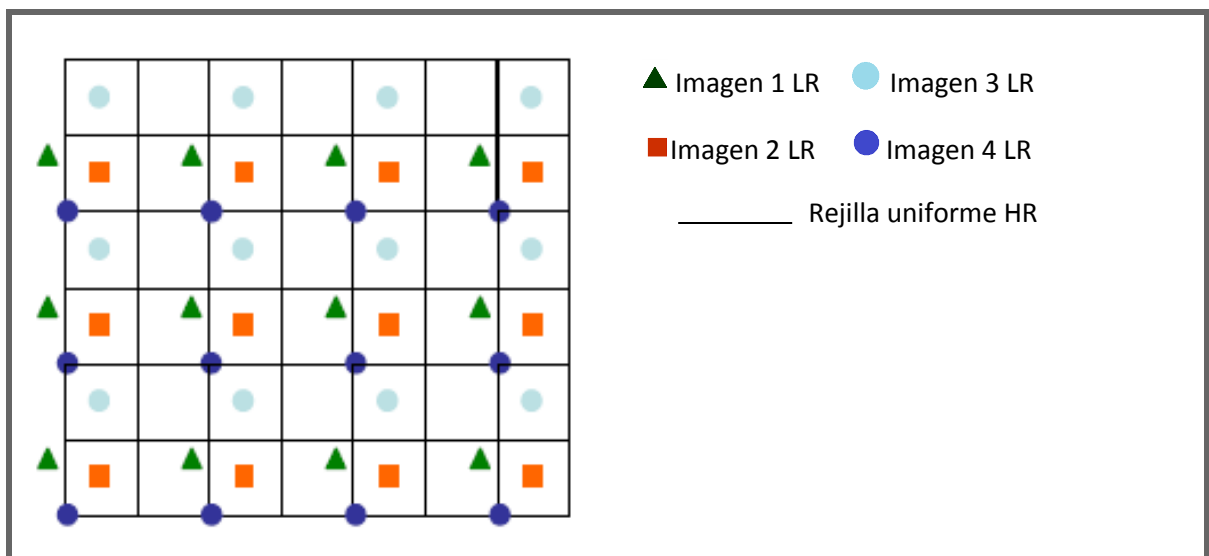


Figura 1.5 Imagen compuesta que muestra una estructura semi-uniforme.

El objetivo principal de los métodos de súper-resolución no sólo consiste en recuperar la imagen de alta resolución de una o más imágenes de entrada de baja resolución. Otro objetivo es magnificar una imagen mientras mantiene la nitidez de los bordes y los detalles de la misma. En contraste con esto, en la súper-resolución, es recuperar detalles perdidos en la nueva imagen de alta resolución que no son explícitos y fácil de encontrar en la correspondiente imagen de baja resolución.

En este trabajo nosotros damos por hecho la adquisición y registro, para enfocarnos únicamente en la estimación de valores así como en su proyección en la rejilla de alta resolución. Específicamente nos concentraremos a partir de ahora en los procesos y algoritmos para generar la imagen de HR a partir de la imagen de LR. En principio, pareciera que interpolación y súper-resolución son equivalentes. Sin embargo, ambos conceptos son totalmente distintos. Más bien, como se mencionó anteriormente, la interpolación es solamente una etapa dentro del proceso de súper-resolución.

Capítulo

2

Métodos de Interpolación

La interpolación es una estimación informada de algo desconocido. Básicamente, la interpolación y todas sus técnicas se basan en la idea que a partir de la información conocida de una señal, es posible estimar y recuperar información importante que quizá se haya perdido durante el proceso de captura. Entre mayor sea esta pérdida, la resolución de la señal, de la imagen o del video será menor y su calidad se degradará de manera importante. Por esta razón, las técnicas de interpolación son fundamentales dentro del proceso de súper-resolución, ya que, si bien, la obtención de imágenes de alta resolución no implica solamente incrementar su resolución espacial, este es el primer paso en cualquier proceso de súper-resolución [6].

2.1 El proceso de Interpolación.

La pérdida de información ocasionada por las limitaciones físicas de los dispositivos de captura, es un fenómeno que afecta tanto a señales unidimensionales como bidimensionales. Precisamente en el caso unidimensional, un concepto relacionado con la interpolación es la operación de ajuste de curva, cuyo objetivo es estimar una función que sea lo más cercana posible a los datos conocidos o experimentales. En este caso, la interpolación [27] se refiere a la construcción de nuevos puntos a partir de un conjunto de puntos discretos conocidos. En la imagen 2.1 se muestra un ejemplo de interpolación de una señal.

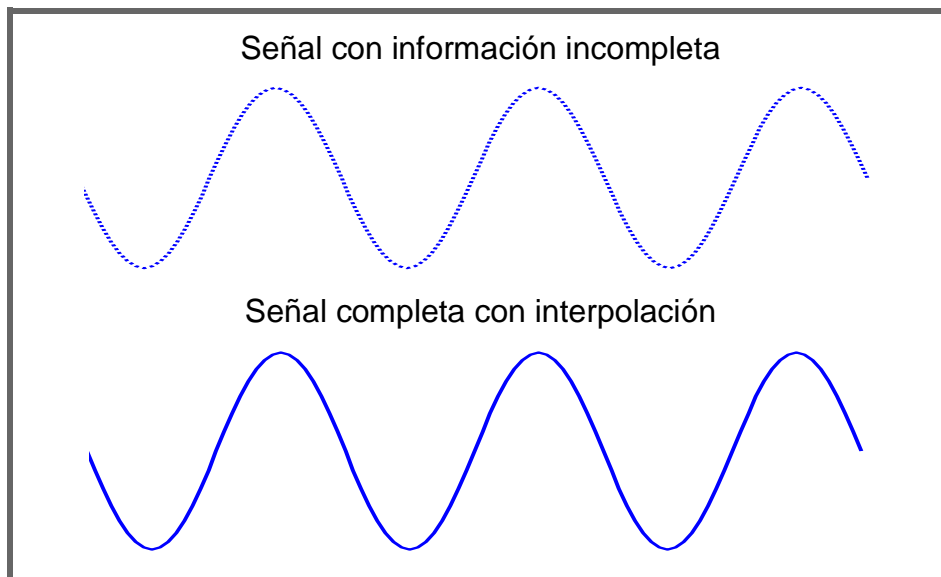


Figura 2.1 Ejemplo de interpolación de una señal.

En el caso de señales bidimensionales en la forma imágenes digitales, el uso más común de la interpolación es modificar el tamaño de una imagen, ya sea para aumentarlo o reducirlo como se muestra en la figura 2.2. De esta forma, cuando se aumenta el tamaño de una imagen discreta, los puntos originales se completan con nuevos píxeles interpolados [28]. En el caso contrario, cuando se reduce el tamaño de una imagen, también se necesita una técnica de interpolación, ya que, aunque hay pérdida de píxeles, los que quedan deben “representar” a los que desaparecen.

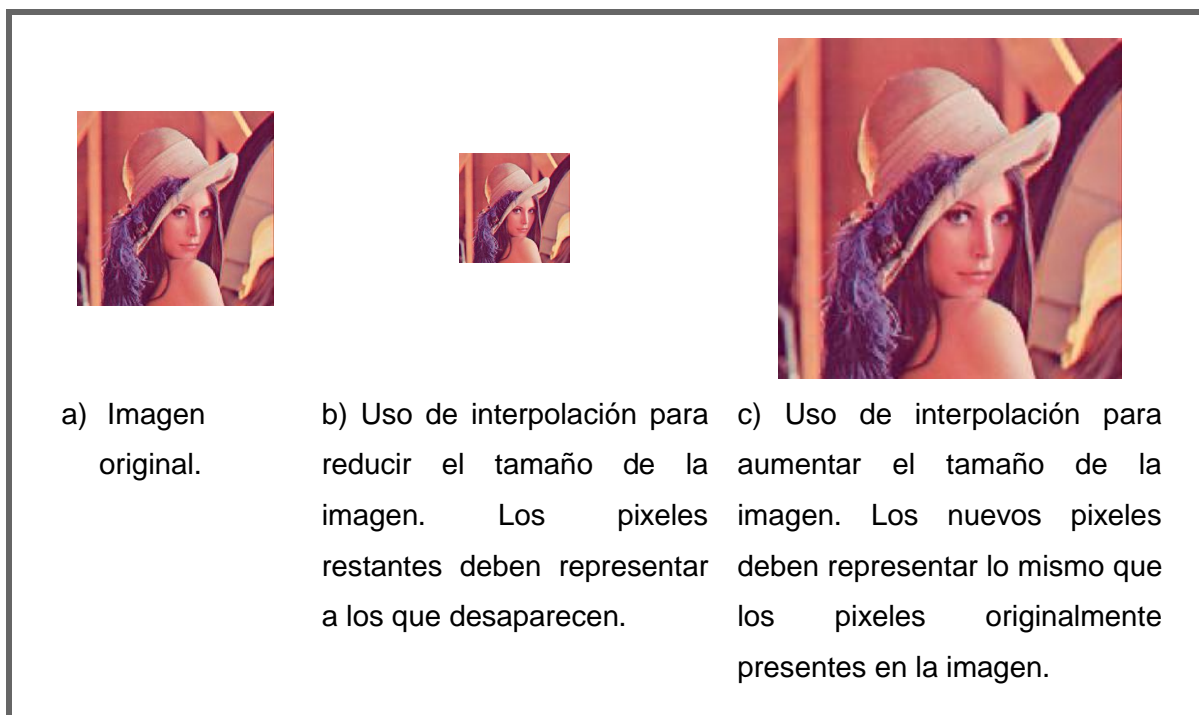


Figura 2.2 Uso de la interpolación en imágenes digitales.

El proceso de interpolación de una imagen tiene como objetivo estimar pixeles intermedios entre pixeles conocidos como se ilustra en la figura 2.3.

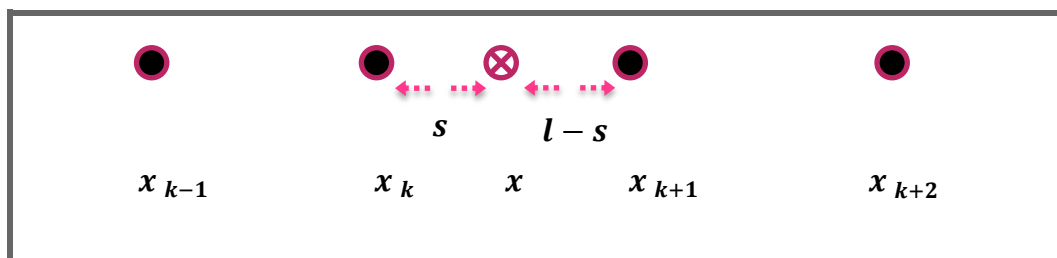


Figura 2.3 Distancia de interpolación a partir de pixeles conocidos.

De acuerdo con la figura 2.3, para estimar el valor de pixel intermedio x entre dos pixeles cualquier conocidos de la imagen, se utilizan como referencia los valores de los pixeles vecinos anterior y posterior x_k y x_{k+1} , respectivamente, así como la distancia s que los separa de la posición a interpolar. La ecuación general de interpolación usada para estimar el valor $f_{HR}(x)$, puede escribirse en la forma [40-42]:

$$f_{HR}(x) = \sum_{k=-\infty}^{\infty} c(x_k)\beta(x - x_k) \quad (2.1)$$

Donde $\beta(x)$ es la función base de interpolación o kernel de interpolación. Los valores $c(x)$ se denominan coeficientes de interpolación y se obtienen a partir de los píxeles de la imagen original proyectados sobre una rejilla de alta resolución, como se describió en la sección 1.5 y tal como se muestra en la figura 1.2.

De acuerdo con la ecuación 2.1, la interpolación de una imagen digital es simplemente una convolución discreta entre los píxeles conocidos de la imagen y el kernel de interpolación. En el contexto del procesamiento digital de imágenes la convolución se implementa como un filtro digital, renglón por renglón y posteriormente, columna por columna.

2.2 Interpolación ideal.

El teorema de muestreo señala que es posible la reconstrucción (aproximadamente) exacta de una señal continua en el tiempo en banda base, a partir de sus muestras, si la señal es limitada en banda y la frecuencia de muestreo es mayor que dos veces el ancho de banda de la señal. Si las condiciones de muestreo no se satisfacen, entonces las frecuencias se pueden llegar a traslapar; es decir, las frecuencias superiores a la mitad de la frecuencia de muestreo serán reconstruidas y aparentarán ser frecuencias por debajo de la frecuencia de muestreo. El resultado sería una distorsión visual denominada *aliasing*.

Entonces, para una señal de banda limitada, si los instantes de muestreo son lo suficientemente cercanos, entonces la señal se puede reconstruir exactamente: en otras palabras, mediante el uso de un filtro pasa bajas se puede efectuar la interpolación exacta entre los puntos de muestra. Para ello se utiliza la función Sinc, la cual representa un filtro ideal que elimina todas las frecuencias por encima de ancho de banda dado y deja pasar bajas frecuencias, perfectamente cortando altas frecuencias.

Con fundamento en la teoría clásica del muestreo, si $f(x)$ está limitada en banda dentro de $[-\pi, \pi]$, entonces:

$$\hat{f}(x) = \sum_{k=-\infty}^{\infty} f(x_k) \text{sinc}(x - x_k) \quad (2.2)$$

Esta se conoce como interpolación ideal, sin embargo, su complejidad computacional impone como restricción el diseño de filtros ideales en su implementación, lo que la hace imposible de realizar. El principal problema con la también denominada interpolación *sin* [29], es que el kernel de interpolación $\text{sin}(x)$ mostrado en la figura 2.4, presenta una caída muy lenta, lo cual vuelve ineficientes los cálculos en el dominio del tiempo.

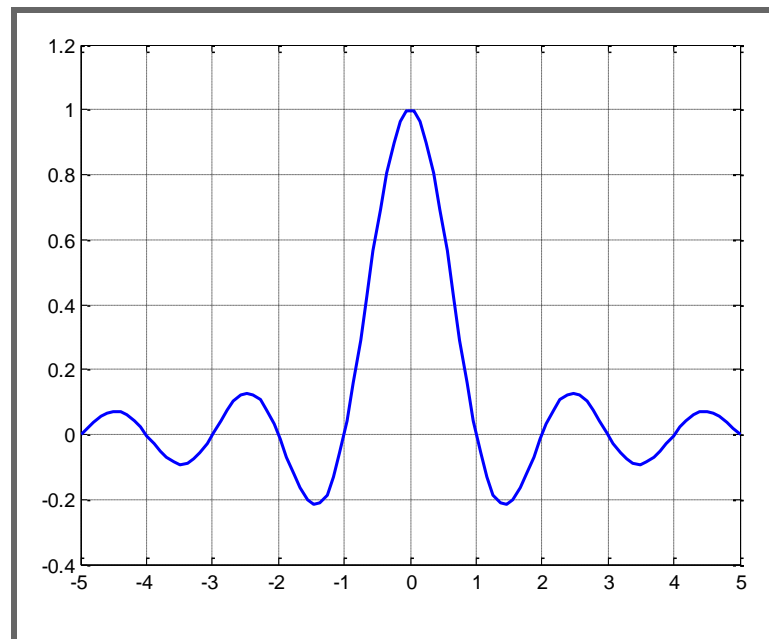


Figura 2.4 Función Sin

Dado que la función *Sin* es una función infinita, mientras que las imágenes digitales se limitan siempre dentro de un campo finito, la interpolación resulta ser meramente una limitada aproximación, por esta razón, como alternativa, es preferible utilizar otro tipo de funciones acotadas en el tiempo tales como las denominadas funciones Spline o B-splines [30,31].

2.3 Convolución sobre una imagen.

En términos generales, la convolución es una operación matemática que combina dos señales para producir una tercera. La convolución aplicada al procesamiento digital de imágenes es un proceso en el que obtenemos como resultado (salida) una imagen con algún efecto que la hace distinta a la imagen original de entrada.

Este proceso está basado en las propiedades de la convolución respecto de la transformada de Fourier, para simplificar el cálculo matemático y reducir los tiempos de procesamiento.

Un filtrado de convolución, para una imagen digital, es un plano bidimensional (x,y), puede representarse como una matriz cuadrada (matriz de convolución), de dimensiones (de 3x3, 5x5 píxeles en la mayoría de los casos) denominada kernel de convolución. El filtrado debe ser de tamaño impar, para que tenga un centro ya que dicho centro se hará con cada uno de los píxeles de la imagen.

Una imagen digital puede verse como una cuadrícula en el que cada cuadro (pixel) tiene asignado un color concreto específico por medio de su código de color. Aplicar un filtro de convolución implica superponer simbólicamente la matriz de convolución sobre la cuadrícula que conforma la imagen. El proceso aritmético de aplicación de filtro consiste en multiplicar el valor de cada cuadrícula de la matriz por el situado debajo suyo en esa superposición imaginaria. Los resultados de todos esos productos se suman obteniendo un valor que sustituirá al valor del pixel central del área que abarca la superposición en la imagen resultante. En otras palabras, la matriz de convolución se desplaza sobre la imagen original haciendo que el elemento central de la matriz de convolución coincida con cada uno de los píxeles de la imagen, que coincide en posición con un elemento de la matriz de convolución, por el valor de este. El pixel de la imagen, que coincide con el elemento central de la matriz de convolución, es substituido por la suma de los productos.

El valor anterior puede ser corregido por medio de un factor multiplicador del resultado y también por medio un parámetro de desplazamiento que permitiría incrementar (o reducir) en un valor entero el resultado anterior.

La intensidad de color del pixel resultado puede ser negativo o mayor que 255, si eso sucede debemos truncarlo para que valores menores de cero sean 0 y valores mayores de 255 sean 255.

En la práctica la operación de convolución requiere de 4 ciclos anidados, por lo que no es muy rápido, a menos que se utilicen filtros pequeños. Usualmente se utilizan filtros de tamaño 3x3 o 5x5.

La operación debe aplicarse sobre cada uno de los canales RGB en imágenes a color, por lo que para agilizar el proceso se puede aplicar sobre una imagen en escala de grises, evitando hacer tres veces el proceso (El código de color de un tono de gris tiene la misma intensidad en sus canales RGB).

En la figura 2.5 se muestra el proceso de convolución que se le aplica a un pixel.

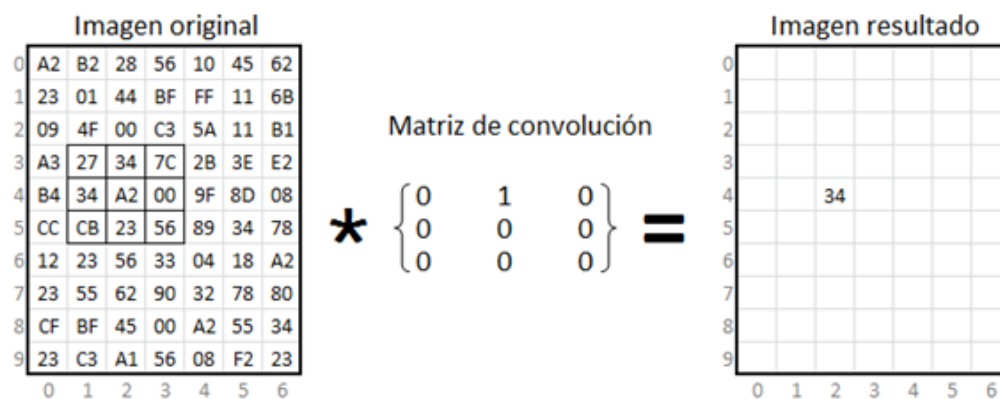


Figura 2.5 Convolución aplicada al pixel (2,4)

2.4 Interpolación polinomial y el kernel de interpolación.

Una labor fundamental en el método de interpolación descrito por la ecuación 2.3 es la adecuada selección de las funciones base de interpolación $\beta(x)$. En la interpolación polinomial, el kernel de interpolación [30], como también se denomina a las funciones base de interpolación, se elige entre un conjunto de n polinomios relacionados entre sí mediante n operaciones de convolución. De esta forma, la ecuación 2.1 puede reescribirse de la siguiente manera:

$$\hat{f}(x) = \sum_{k \in Z} c(x_k) \beta^n(x - x_k) \quad (2.3)$$

En donde Z representa una vecindad finita alrededor de x .

La función base $\beta^n(x)$ se conoce como B-spline [33] y representa una función en forma de campana simétrica obtenida mediante $n + 1$ convoluciones de un pulso rectangular β^0 definido por:

$$\beta^0(x) = \begin{cases} 1 & -\frac{1}{2} < x < \frac{1}{2} \\ \frac{1}{2} & |x| = \frac{1}{2} \\ 0 & \text{en otro caso} \end{cases} \quad (2.4)$$

De esta forma, una función base cualquiera puede obtenerse realizando una serie de convoluciones entre $\beta^0(x)$ con sí misma, esto es:

$$\beta^n(x) = \underbrace{\beta^0(x) * \beta^0(x) * \dots * \beta^0(x)}_{n+1 \text{ veces}} \quad (2.5)$$

Utilizando la ecuación anterior se pueden obtener distintas funciones base para realizar la interpolación de imágenes.

En la práctica del procesamiento de señales, los valores de $\beta^n(x)$, $c(x)$ y $\hat{f}(x)$ son vectores de números reales, los cuales se pueden procesar y graficar usando lenguajes de programación. Por ejemplo, si se define el vector $\beta^0(x)$ como se indica en la ecuación 2.4, se obtiene la primera función base, la cual tiene la forma de un pulso rectangular como el que se muestra en la figura 2.6.

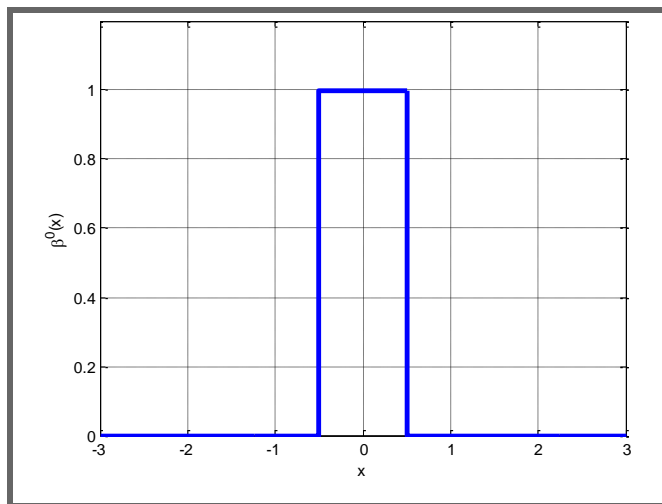


Figura 2.6 Función base $\beta^0(x)$

Si se toma esta primera función base $\beta^0(x)$ y se calcula su convolución con sí misma $\beta^0(x) * \beta^0(x)$, se obtiene la segunda función base de la serie, la cual tiene la forma de un pulso triangular como se muestra en la figura 2.7.

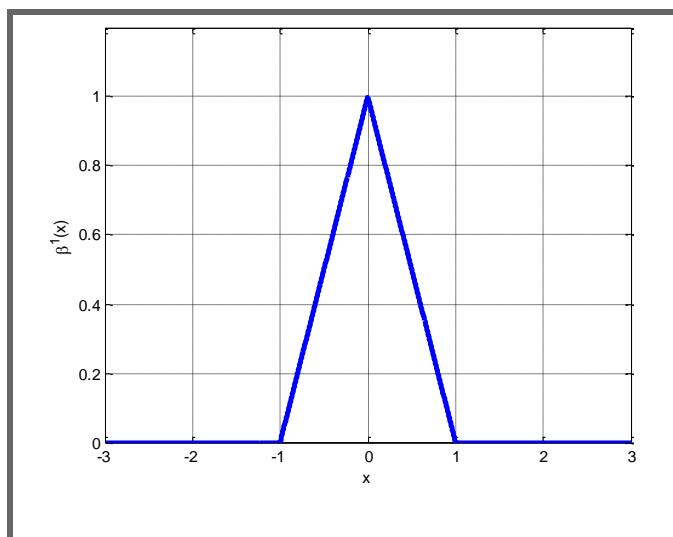


Figura 2.7 Función base $\beta^1(x) = \beta^0(x) * \beta^0(x)$

De la misma forma, si se toma el resultado anterior y se realiza su convolución con $\beta^0(x)$, es decir $\beta^2(x) = \beta^1(x) * \beta^0(x)$, se obtiene la tercer función base, la cual presenta una forma de campana. De hecho, a partir de esta función en adelante, todas las funciones base de la serie presentan una forma de campana.

En las figuras 2.8 y 2.9 se muestran la tercera y cuarta función base de la serie correspondientes a $\beta^2(x) = \beta^1(x) * \beta^0(x)$ y $\beta^3(x) = \beta^2(x) * \beta^0(x)$, respectivamente.

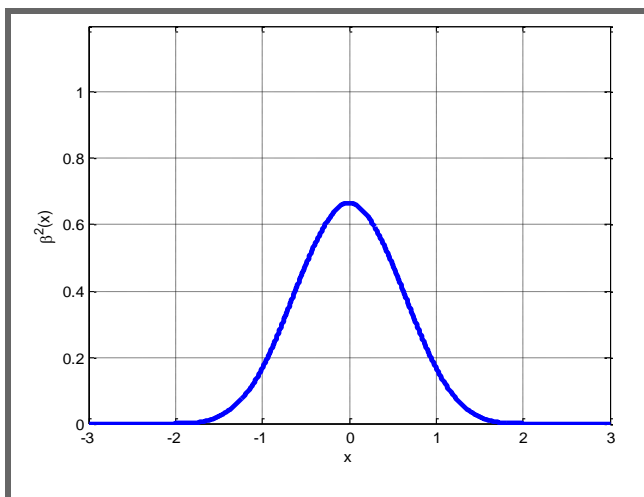


Figura 2.8 Función base $\beta^2(x) = \beta^1(x) * \beta^0(x)$

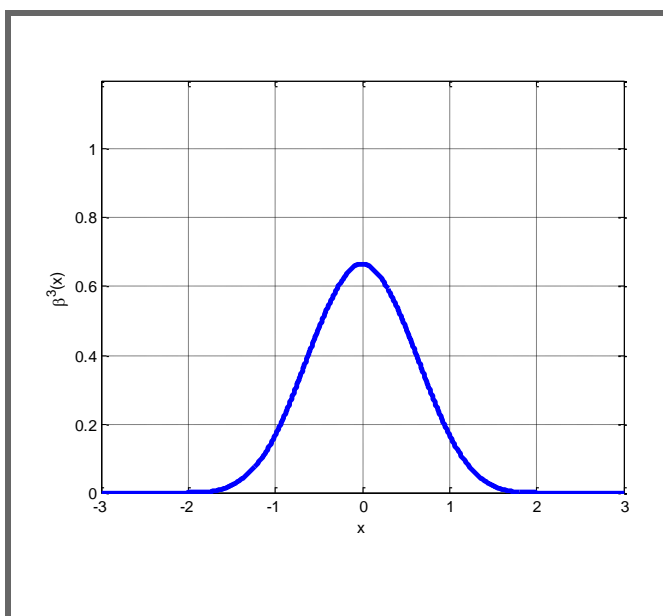


Figura 2.9 Función base $\beta^3(x) = \beta^2(x) * \beta^0(x)$

La selección de una función base específica $\beta^n(x)$ en la ecuación 2.3, define un kernel de interpolación específico y, por lo tanto, un método de interpolación específico. Cada uno de los distintos métodos ofrece una manera distinta de estimar

los valores desconocidos de la imagen y, como se demostrará posteriormente, la selección del kernel tiene un efecto significativo en la calidad visual de la imagen interpolada. Existen otras clases de métodos que no necesariamente utilizan representaciones polinomiales de funciones para estimar los valores de alta resolución. Entre estos se encuentran los que utilizan algún tipo de transformación como por ejemplo, la transformada coseno o la transformada wavelet. Las técnicas consideradas en este trabajo están basadas por una parte en funciones de interpolación polinomial, pero también se aborda el caso de la transformada coseno. A pesar de que se estudió también el uso de la transformada wavelet, los resultados no se incluyen en este trabajo debido a que forman parte del trabajo a futuro de la tesis y serán incluidos en el trabajo de tesis de maestría.

En las siguientes secciones se describen los modelos de interpolación polinomial usados comúnmente en la súper-resolución de imágenes, los cuales se distinguen entre sí solamente por la selección de $\beta^n(x)$ en la ecuación de interpolación 2.3.

2.5 Los coeficientes de interpolación.

El parámetro $c(x)$ en las ecuaciones 2.1 y 2.3 es la función con la cual se realizará la convolución con el kernel de interpolación. Este parámetro es una función que se aplica sobre la imagen para obtener un conjunto de valores que incluyen los pixeles conocidos de esta más los pixeles que deberán ser estimados en la interpolación. En un principio, dado que se desconoce el valor de los pixeles, estos toman el valor de cero, lo que equivale a intercalar ceros entre los pixeles de la imagen. Al hacerlo fila por fila y columna por columna, se obtiene una proyección de los valores conocidos de la imagen sobre una especie de “rejilla” con una dimensión espacial aumentada. Por ejemplo, en la figura 2.10 se muestra una imagen de entrada adquirida por un dispositivo de baja resolución, la cual es mapeada sobre una rejilla con el doble de pixeles que la imagen original. Con el fin de apreciar mejor la distribución de los pixeles cero en la rejilla, a propósito se ha cambiado la dimensión real, al doble de la imagen de entrada. Los pixeles de la imagen son entonces proyectados sobre la rejilla, quedando intercalados con los pixeles de valor cero. Al resolver la ecuación 2.3, cada fila y cada columna de la imagen serán precisamente

los coeficientes de interpolación $c(x)$ que serán convolucionados con la función base $\beta^n(x)$.

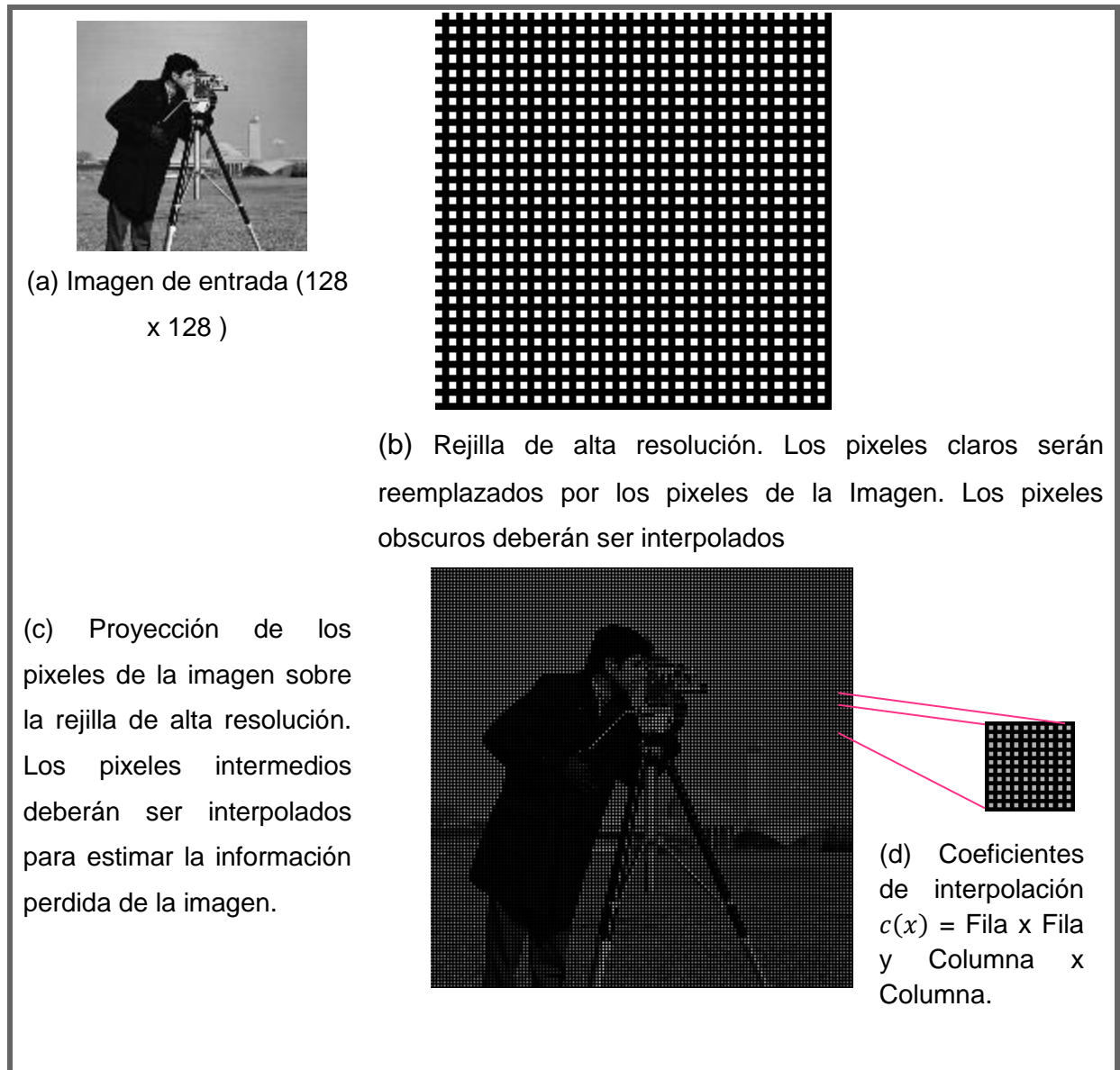


Figura 2.10 Método de rejilla

2.6 Técnicas de interpolación.

La ecuación 2.3 representa el modelo general de interpolación y puede representarse gráficamente como en la figura 2.11.

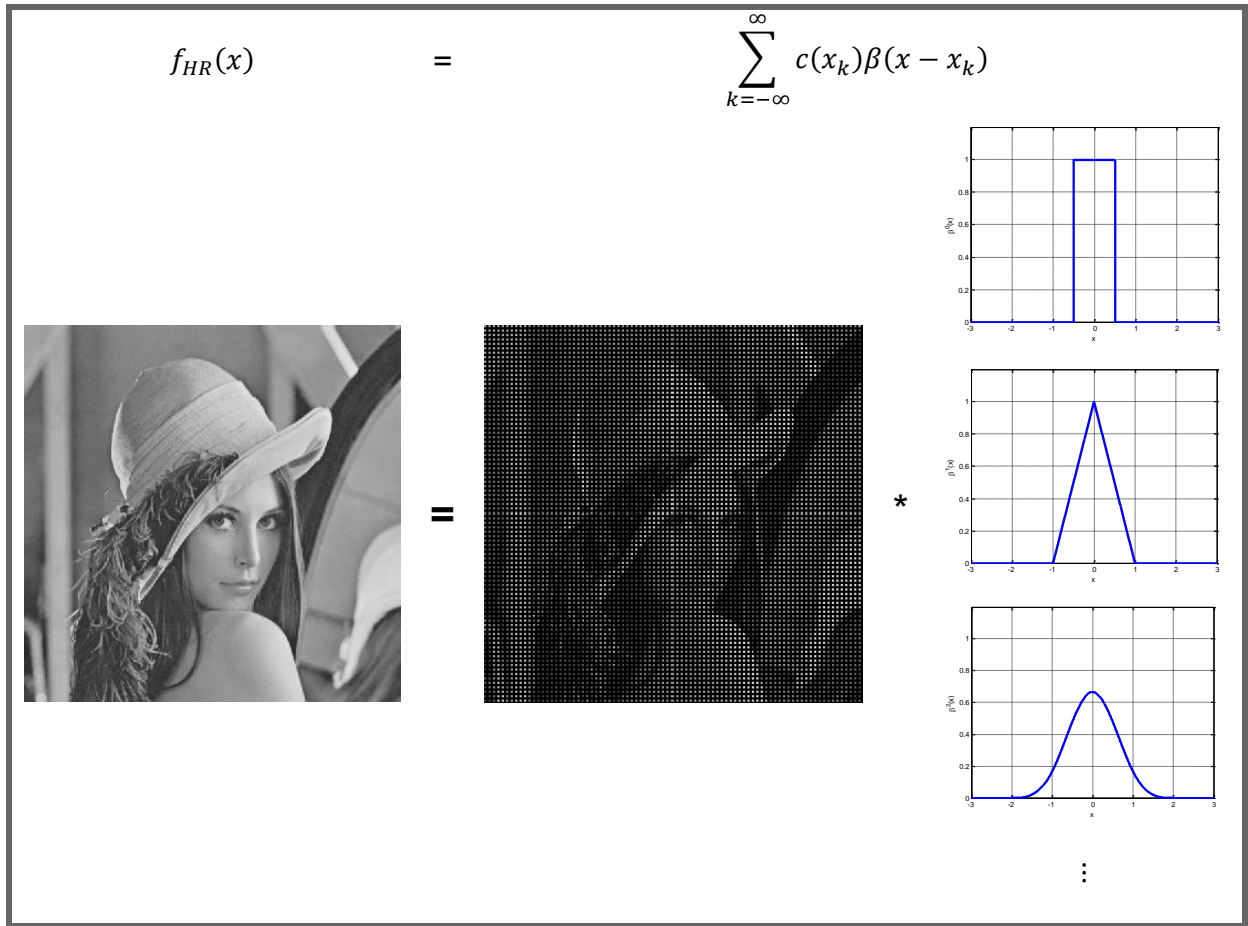


Figura 2.11 Modelo general de interpolación.

En el lado derecho de la figura, aparecen algunas de las distintas formas que puede tomar el kernel de interpolación. A continuación se describen algunas de las técnicas de interpolación más comunes las cuales se distinguen entre sí solamente por la selección de la función base de interpolación $\beta^n(x)$ que representa el kernel de interpolación.

2.6.1 Técnica de interpolación “del vecino más cercano”

La función base de interpolación para este método es precisamente la que se define como β^0 en la ecuación 2.4 y su forma gráfica es la que se muestra en la figura 2.6. Esta función se sustituye en la ecuación 2.3 para estimar los pixeles desconocidos de la imagen interpolada [16]. Realizando la convolución correspondiente, la imagen de salida resultante es la representación en alta resolución de la imagen registrada de baja resolución.

El método del vecino más cercano es la solución más simple para el problema de interpolación. Este consiste en interpolar el valor del pixel de la imagen de salida a partir del pixel vecino más cercano con relación a la posición de los pixeles en la imagen original, tal como se ilustra en la figura 2.12.

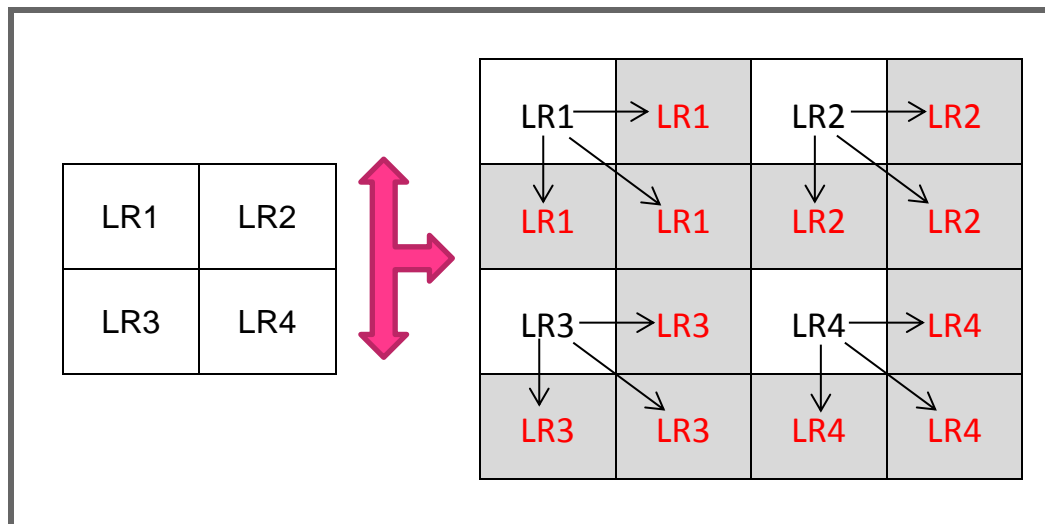


Figura 2.12 Técnica de interpolación del vecino más cercano.

Los píxeles faltantes LR1, LR2, LR3 y LR4 se estiman de la siguiente manera:

Para implementar este método es anticipando el resultado de la convolución con el pulso rectangular, que en este caso es la repetición de los valores de imagen de tal forma que por cada valor en la imagen de baja resolución, tendríamos cuatro valores idénticos en la imagen de alta resolución.

La ventaja de este método es simplicidad, pues en la práctica implica meramente la repetición de los píxeles de la imagen de baja resolución. Lo que hace es considerar el píxel más cercano al píxel a interpolar. Sin embargo, presenta frecuentemente ruido indeseable en las imágenes, tales como distorsión de líneas en imágenes de alta resolución, motivo por el cual no se aconseja para aplicaciones prácticas de súper-resolución. La imagen resultante es más grande que la original, y preserva todo el detalle original, pero presenta *aliasing* en los bordes o líneas diagonales, es decir, se crea la forma característica de “escalera” en líneas o bordes. Finalmente, el resultado presenta una baja calidad si la imagen tiene una gran variación en la gama de grises. A favor puede decirse que requiere un tiempo de procesamiento muy pequeño, en comparación con otros métodos.

2.6.2 Técnica de Interpolación “Bilineal”.

Al seleccionar $\beta^1(x) = \beta^0(x) * \beta^0(x)$ en la ecuación 2.3, se obtiene el método de interpolación bilineal. Esta es una forma simple de aumentar el tamaño de una imagen y es más eficiente que el método del vecino más cercano. Al igual que en una interpolación lineal, los píxeles a interpolar son estimados a partir de una superficie lineal entre los píxeles conocidos de la imagen original, pero en lugar de repetir el valor de un solo píxel, se toma el promedio de los píxeles vecinos [19].

La función base utilizada por el método de interpolación bilineal está definida [42,43] entonces como en la ecuación 2.7 y su gráfica es la curva en forma de pulso triangular mostrada en la figura 2.7.

$$\beta^1(x) = \begin{cases} 1 - |x| & |x| < 1 \\ 0 & 1 \leq |x| \end{cases} \quad (2.7)$$

A partir de las ecuaciones 2.3 y 2.7, y de acuerdo con la figura 2.3, las distancias entre los píxeles x y x_k , y entre x_{k+1} y x son, respectivamente:

$$s = x - x_k \quad \text{y} \quad l - s = x_{k+1} - x \quad (2.8)$$

Sustituyendo las ecuaciones 2.7 y 2.8 en la ecuación 2.3, obtenemos un modelo equivalente para la interpolación bilineal:

$$f_{HR}(x) = (l - s)f(x_k) + sf(x_{k+1}) \quad (2.9)$$

Usando este método, cada pixel de la figura 2.13, es interpolada considerando los cuatro pixeles que lo rodean:

LR1	$\frac{LR1}{2} + \frac{LR2}{2}$	LR2	$\frac{LR2}{2} + \frac{LR3}{2}$	LR3
$\frac{LR1}{2} + \frac{LR4}{2}$	$\frac{LR1 + LR2 + LR4 + LR5}{4}$	$\frac{LR2}{2} + \frac{LR5}{2}$	$\frac{LR2 + LR3 + LR5 + LR6}{4}$	$\frac{LR3}{2} + \frac{LR6}{2}$
LR4	$\frac{LR4}{2} + \frac{LR5}{2}$	LR5	$\frac{LR5}{2} + \frac{LR6}{2}$	LR6
$\frac{LR4}{2} + \frac{LR7}{2}$	$\frac{LR4 + LR5 + LR7 + LR8}{4}$	$\frac{LR5}{2} + \frac{LR8}{2}$	$\frac{LR5 + LR6 + LR8 + LR9}{4}$	$\frac{LR6}{2} + \frac{LR9}{2}$
LR7	$\frac{LR7}{2} + \frac{LR8}{2}$	LR8	$\frac{LR8}{2} + \frac{LR9}{2}$	LR9

Figura 2.13 Técnica de interpolación Bilineal.

De esta manera, la estimación de los píxeles en la imagen de alta resolución (HR) a partir de la imagen de baja resolución (LR) puede expresarse de la forma:

$$HR(2x - 1, 2y - 1) = LR(x, y)$$

$$HR(2x, 2y - 1) = [LR(x, y) + LR(x, y + 1)]/2$$

$$HR(2x - 1, 2y) = [LR(x, y) + LR(x + 1, y)]/2$$

$$HR(2x - 2y) = [LR(x, y) + LR(x + 1, y) + LR(x, y + 1) + LR(x + 1, y + 1)]/4$$

$$\forall x, y \in LR$$

Como puede apreciarse en la figura 2.13 y en el cuadro anterior, la interpolación bilineal crea un valor medio ponderado que utiliza para rellenar los espacios vacíos. Esto proporciona un mejor equilibrio entre la imagen, calidad y costo computacional, sin embargo, presenta un efecto de difuminación sobre la imagen interpolada. Esta clase de funciones recibe el nombre de Bilineal porque es lineal como función de x cuando y permanece fijo y a su vez lineal como función de y cuando x permanece invariante.

Este tipo de interpolación, es típicamente mejor que el método del vecino más cercano para cambiar el tamaño de una imagen, aunque la imagen resultante puede ser un poco borrosa y presenta en cierto grado el efecto de diente de sierra.

2.6.3 Técnica de interpolación Bicúbica.

La interpolación bicúbica es la más popular entre la familia de métodos de interpolación polinomial [11,20]. Esta técnica es similar a la interpolación bilineal, solo que en vez de tomar los cuatro valores de los píxeles más cercanos, tomas los valores de 16 píxeles más cercanos para promediar y encontrar el valor que se deberá asignar al píxel desconocido. Gracias al empleo de un mayor número de

valores a diferentes distancias del pixel desconocido, este método logra mayor nitidez sobre la imagen de salida.

El kernel de interpolación para la interpolación bicúbica es $\beta^3(x)$ y su forma gráfica es como se muestra en la figura 2.9. Otra forma de obtener los valores de la función base es definiéndola de la siguiente manera:

$$\beta^3(x) = \begin{cases} \frac{3}{2} - |x|^2 + \frac{|x|^3}{2} & 0 \leq |x| < 1 \\ \frac{(2-|x|)^3}{6} & 1 \leq |x| < 2 \\ 0 & 2 \leq |x| \end{cases} \quad (2.10)$$

Para estimar los pixeles desconocidos mediante interpolación bicúbica, puede sustituirse $\beta^3(x)$ en la ecuación 2.3, o puede utilizarse el siguiente modelo alternativo:

$$f_{HR}(i', j') = \sum_{m=-1}^2 \sum_{n=-1}^2 F(i+m, j+n) R(x); \quad (2.11)$$

Dónde:

$$R(x) = \frac{1}{6} [P(x+2)^3 - 4P(x+1)^3 + 6P(x)^3 - 4P(x-1)^3] \quad (2.12)$$

Para:

$$P \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (2.13)$$

Una vez simplificadas las fórmulas 2.3 y 2.11, la interpolación bicúbica resulta ser una convolución o combinación lineal entre los 16 valores del entorno, cuyos coeficientes se calculan según las distancias en vertical y en horizontal sobre las filas y columnas de la rejilla. Esto equivale a la siguiente representación para la interpolación bicúbica:

$$\begin{aligned}
f_{HR}(x) = & c(x_{k-1})[(3+s)^3 - 4(2+s)^3 + 6(1+s)^3 - 4s^3]/6 \\
& + c(x_k)[(2+s)^3 - 4(1+s)^3 + 6s^3]/6 \\
& + c(x_{k+1})[(1+s)^3 - 4s^3]/6 + c(x_{k+2})s^3/6
\end{aligned} \tag{2.14}$$

Este método produce imágenes más nítidas que los dos métodos anteriores, preserva mejor los detalles que la interpolación Bilineal, pero mantiene un efecto de escalera o dentado, aunque en menor grado. Es una técnica que da una mejor calidad de imagen, mucho mejor que las anteriores.

2.6.4 Técnica de interpolación O-MOMS.

La técnica O-MOMS se refiere a una familia de funciones base o kernels de interpolación denominada Interpolación Óptima de Máximo Orden y Mínimo Soporte. [26] Cualquiera de estas funciones se expresa como una suma ponderada de una función spline β^n y su derivada. Por ejemplo, para la interpolación O-MOMS bicúbica, se define la siguiente función base:

$$\beta_{O-MOMS}^3(x) = \beta^3(x) + \frac{1}{42} \frac{d}{dx} \beta^3(x) \tag{2.15}$$

$$= \begin{cases} \frac{1}{2}|x|^3 - |x|^2 + \frac{1}{14}|x| + \frac{13}{21}, & 0 \leq |x| < 1 \\ -\frac{1}{6}|x|^3 + |x|^2 - \frac{85}{42}|x| + \frac{29}{21}, & 1 \leq |x| < 2 \\ 0, & 2 \leq |x| \end{cases} \tag{2.16}$$

2.7 Técnicas de interpolación basadas en transformadas.

2.7.1 La Transformada Coseno Discreta

La transformada coseno discreto (DCT) representa una imagen como una suma de sinusoides de distintas amplitudes y frecuencias [14]. La figura 2.14 muestra gráficamente 64 valores base de la DCT bidimensional.

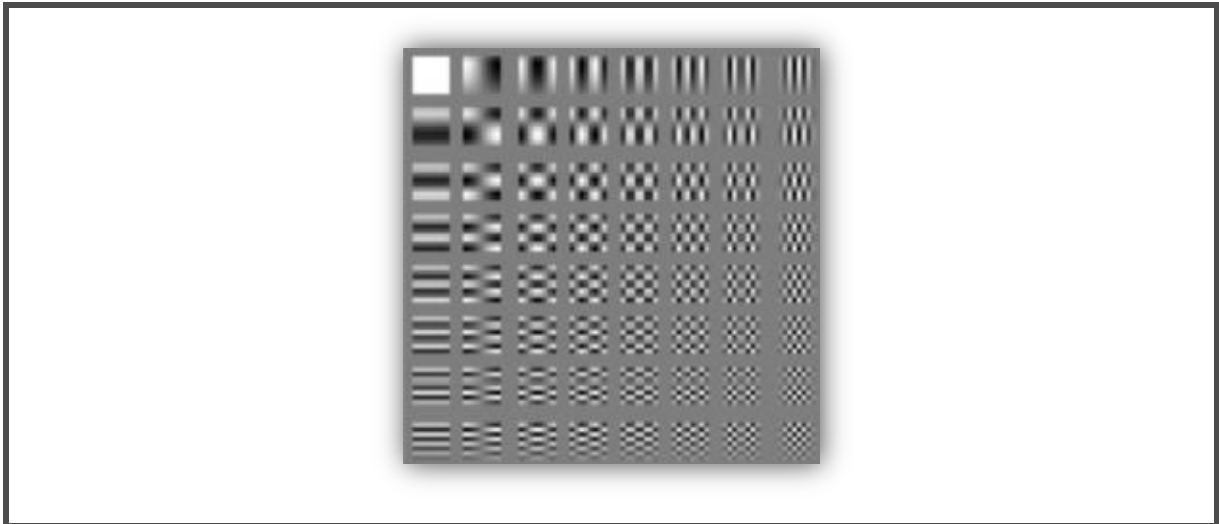


Figura 2.14 Matriz de 8x8 con los 64 valores base para la transformada coseno discreto.

La transformada coseno discreto de una imagen de tamaño $M \times N$ está definida de la siguiente manera [35-39]:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_m \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \leq p \leq M-1 \\ 0 \leq q \leq N-1 \end{matrix} \quad (2.17)$$

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M-1 \end{cases} \quad (2.18)$$

$$\alpha_q = \begin{cases} \frac{1}{\sqrt{MN}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq MN-1 \end{cases} \quad (2.19)$$

Por otra parte, la inversa de la transformada coseno discreta se define como:

$$A_{mn} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \alpha_p \alpha_q B_{pq} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \leq p \leq M-1 \\ 0 \leq q \leq N-1 \end{matrix} \quad (2.20)$$

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M-1 \end{cases} \quad (2.21)$$

$$\alpha_q = \begin{cases} \frac{1}{\sqrt{MN}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq MN-1 \end{cases} \quad (2.22)$$

Por lo general, la transformada coseno discreta se utiliza principalmente para llevar a cabo la compresión de imágenes, sin embargo, al mismo tiempo también puede ser útil para estimar la información perdida en una imagen de baja resolución.

2.7.2 Método de interpolación Transformada Coseno Discreta

En la figura 2.15 podemos observar el diagrama del algoritmo que se utilizó para el proceso del método de interpolación DCT, donde como primer paso obtenemos una imagen de baja resolución, en seguida se le aplica el método DCT, el resultado es proyectado a una rejilla de alta resolución, para después al resultado de dicha proyección se le aplicara la inversa de DCT, por lo cual se obtendrá la estimación de los pixeles faltantes y por ultimo nos muestra la imagen recupera de alta resolución.

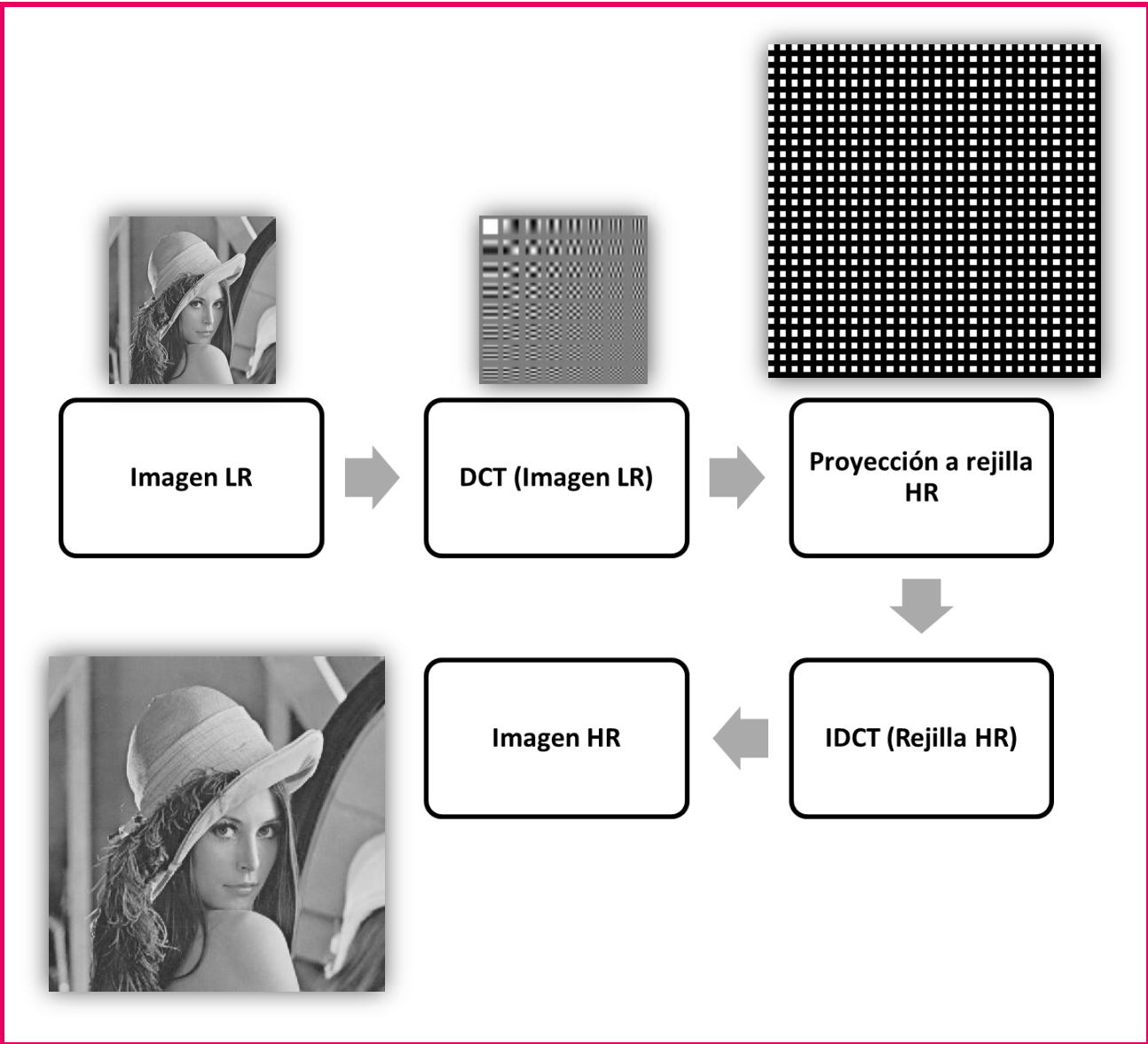


Figura 2.15 Proceso de Interpolación DCT.

Capítulo

3

Evaluación de algoritmos sobre imágenes.

En este capítulo se presentan los resultados obtenidos al implementar métodos descritos en el capítulo anterior. Para ello se desarrollaron los programas correspondientes a cada algoritmo y se utilizaron algunas imágenes de prueba en escala de grises y a color en formato RGB, evaluando su desempeño mediante la relación señal a ruido pico (PSNR) y el índice de similitud estructural (SSIM) como métricas de calidad.

En la figura 3.1 se presenta en forma de diagrama de flujo el procedimiento general utilizado para evaluar los algoritmos desarrollados. Más adelante, se mostrará con mayor detalle este procedimiento.

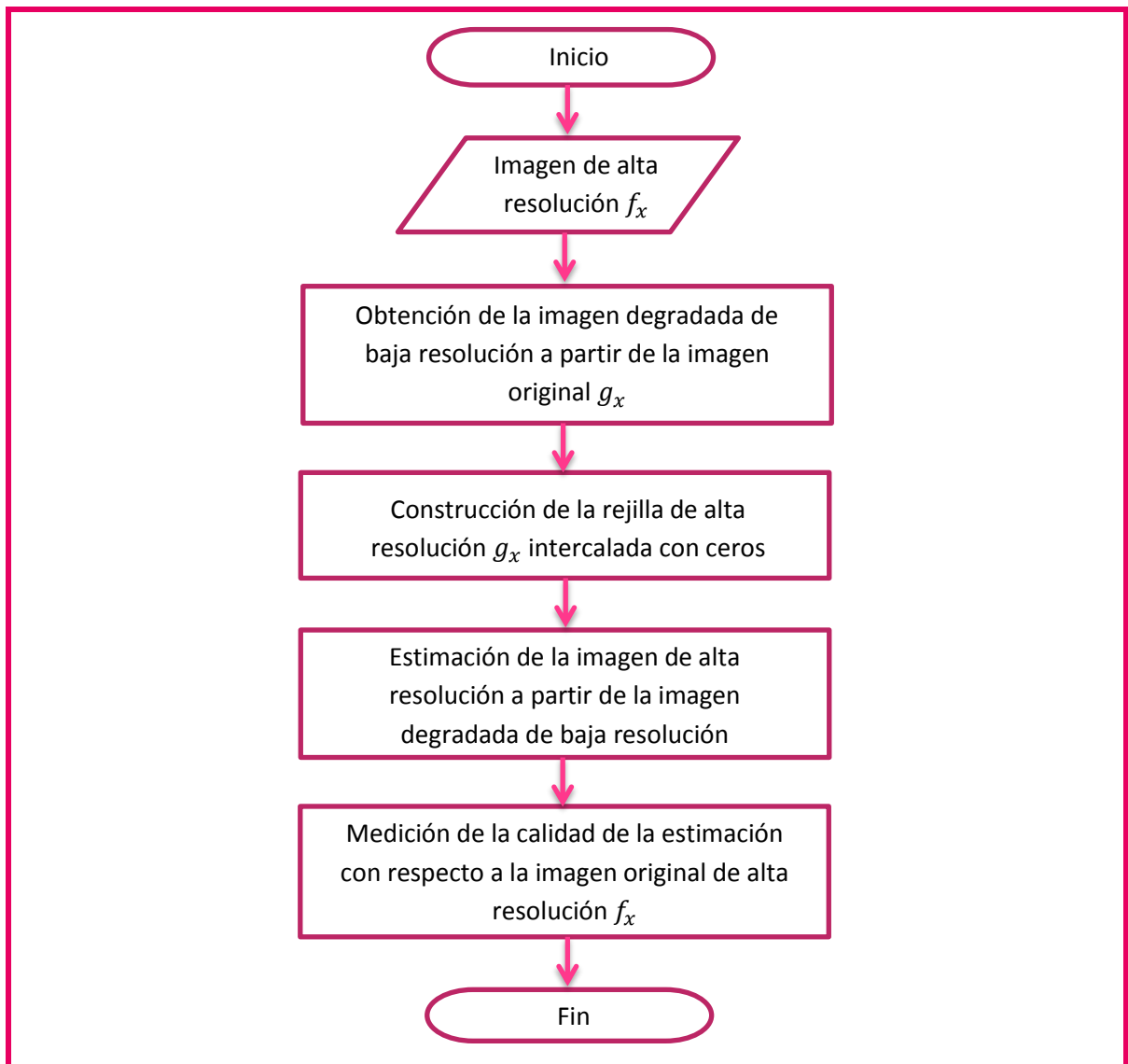


Figura 3.1 Diagramas de flujo con el procedimiento de interpolación y evaluación de los algoritmos.

3.1 Obtención de la imagen de baja resolución.

En el procesamiento de imágenes, cuando una imagen es capturada mediante una cámara de alta resolución, la imagen obtenida se puede representar como $f(n_1, n_2)$, donde $n_1, n_2 = 0, 1, 2, \dots, N - 1$, sin embargo, cuando se utiliza una cámara de baja resolución, la imagen resultante es $g(m_1, m_2)$, para $m_1, m_2 = 0, 1, 2, \dots, M - 1$. En este caso $M = N/R$, y R es la razón entre las tasas de muestreo de $f(n_1, n_2)$ y $g(m_1, m_2)$. La relación entre la imagen de baja resolución LR y la imagen de alta resolución HR puede representarse mediante el siguiente modelo matemático [33,34]:

$$\mathbf{g} = \mathbf{D}\mathbf{f} + \mathbf{v} \quad (3.1)$$

Donde \mathbf{g} representa los vectores fila y columna de la imagen de baja resolución LR, \mathbf{f} representa los vectores fila y columna de la imagen de alta resolución HR y \mathbf{v} representa el vector de ruido. La matriz \mathbf{D} representa el proceso de degradación y filtrado de la imagen HR y su función es introducir una degradación similar a la que ocurre cuando se captura la imagen mediante un dispositivo de baja resolución. Para ello, debe aplicarse el modelo 3.1 fila por fila y columna por columna sobre cualquier imagen de alta resolución, como se muestra en las figuras 3.2 y 3.3. Este procedimiento se realiza por conveniencia para facilitar la medición de la calidad de las imágenes reconstruidas usando los diferentes algoritmos, ya que todas las métricas de calidad en imágenes necesitan una imagen de referencia para poder cuantificar el error de las estimaciones. De esta forma, la imagen de alta resolución original puede ser utilizada como imagen de referencia.

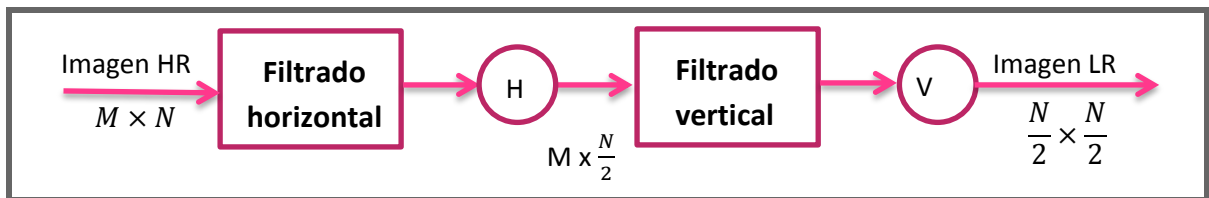


Figura 3.2 Proceso de obtención de la imagen de baja resolución (LR).

Una vez obtenida la imagen degradada LR como se muestra en la figura 3.4, es posible aplicarle los algoritmos descritos en el capítulo anterior para estimar las secuencias fila y columna \mathbf{f} a partir de las secuencias \mathbf{g} . De esta forma, los algoritmos se aplican sobre la imagen de baja resolución para tratar de reconstruir su equivalente en alta resolución, la cual será comparada con la imagen original con las mismas dimensiones para medir la calidad de los algoritmos.

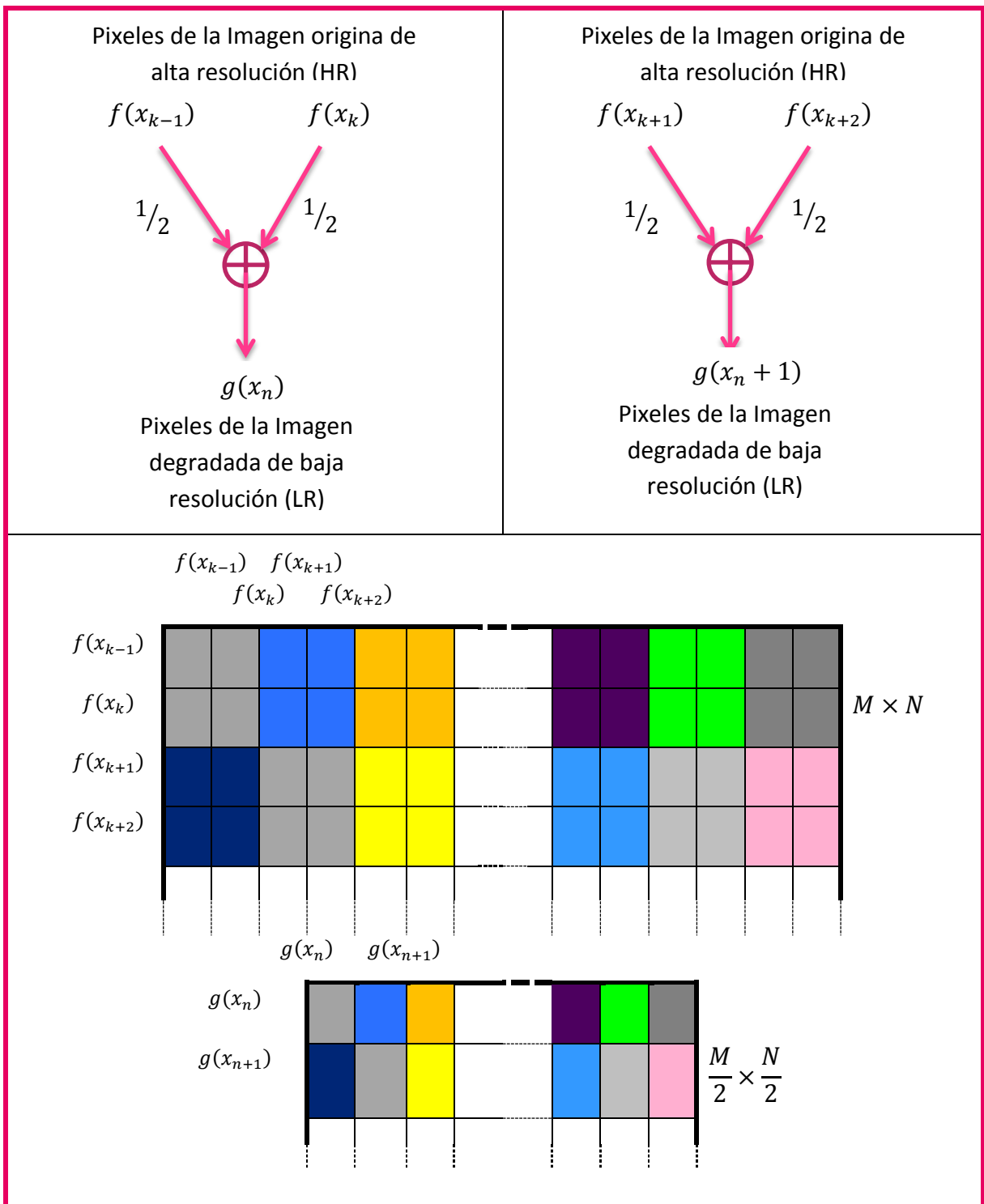


Figura 3.3 Proceso de obtención de la imagen de baja resolución (LR) a partir de una imagen de prueba de alta resolución (HR).

Imagen original de alta resolución (HR)

$f(x)$

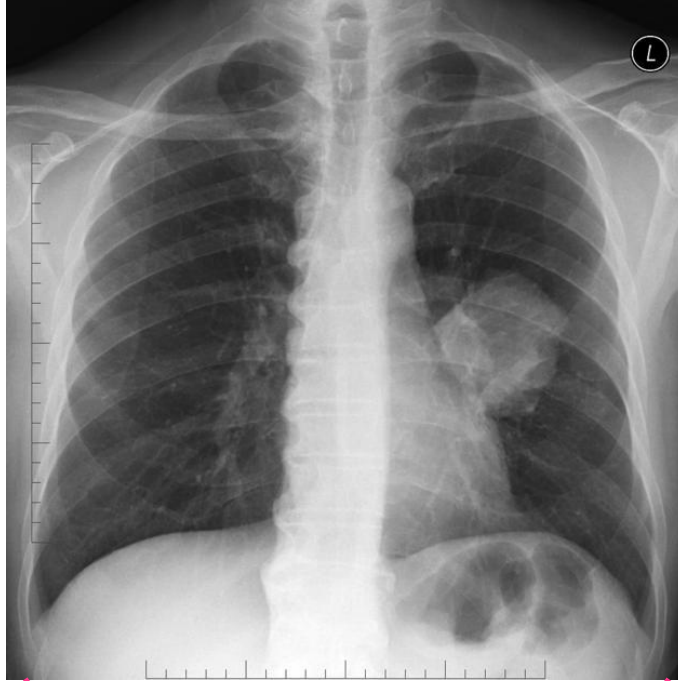


Imagen degradada de
baja resolución (LR)

$g(x)$

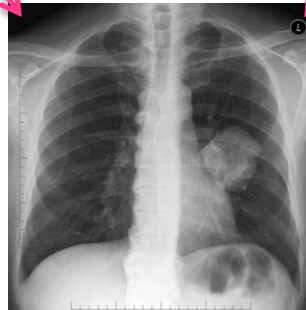


Figura 3.4 Demostración grafica del método de obtención de la imagen LR.

3.2 Imágenes de prueba.

Para el proceso de simulación de los algoritmos se eligieron cinco imágenes entre ellas existen en escala de grises y en formato RGB, las cuales se eligieron de una base de datos médica, en donde las podemos encontrar en la siguiente dirección electrónica www.medpix.nlm.nih.gov. La decisión de tomar estas imágenes es por, sus dimensiones, la definición de sus contornos, los detalle que se encuentran en cada imagen, las diferencia que existe entre cada una.

De acuerdo con el procedimiento descrito en la sección 1.5, los algoritmos que se evalúan a continuación corresponden a la fase de interpolación, y se asume que la imagen adquirida ya ha sido capturada y registrada debidamente usando alguna cámara fotográfica de baja resolución o recibida desde un canal de baja capacidad. Además, se asume que en un trabajo futuro, los resultados obtenidos y presentados en esta tesis serán procesados para completar el proceso de súper-resolución. Para ello, fue necesario realizar un pre-procesamiento sobre las imágenes de alta resolución seleccionadas similar al utilizado en [18] con el fin de obtener sus versiones degradadas de baja resolución. En las figuras 3.5 – 3.9 se presentan estas imágenes de prueba en su versión original de alta resolución.

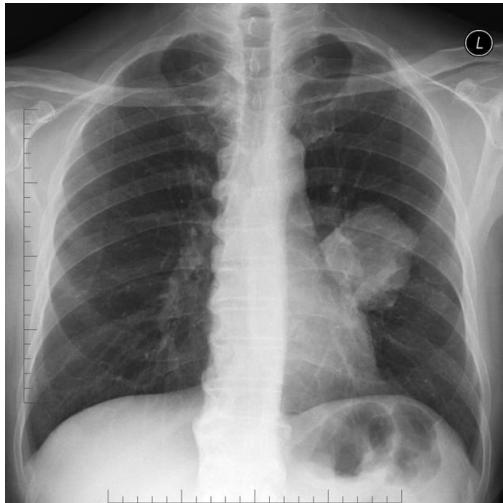


Imagen pulmón con cáncer de
512 x 512 píxeles.

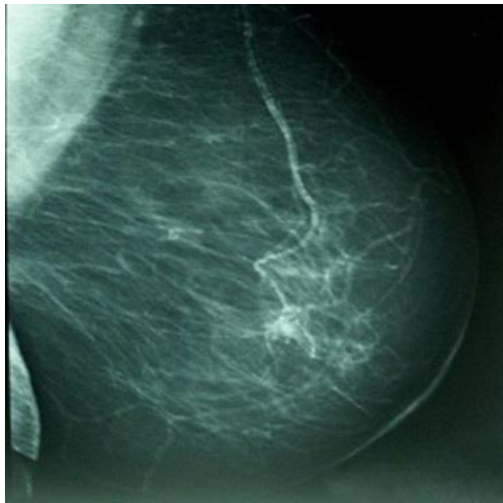


Imagen Mastografía de 512 x
512 píxeles.

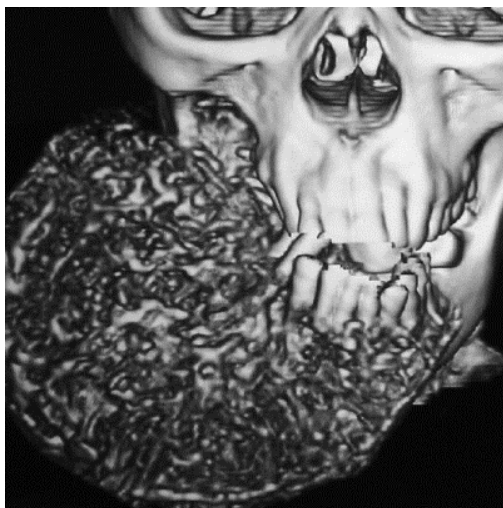


Imagen Tumor en mandíbula
404 x 404 píxeles.



Imagen Pulmones RGB de 950 x 950 pixeles.



Imagen Tumor RGB de 385 x 385 pixeles.

Figura 3.5 Imágenes de prueba.

3.3 Procedimiento de evaluación de los algoritmos.

Para evaluar los algoritmos desarrollados, se aplicó cada uno de estos sobre las diferentes imágenes de prueba y se evaluó su calidad usando la relación pico señal a ruido (PSNR) y el índice de similitud estructural (SSIM), como se muestra en la figura 3.6.

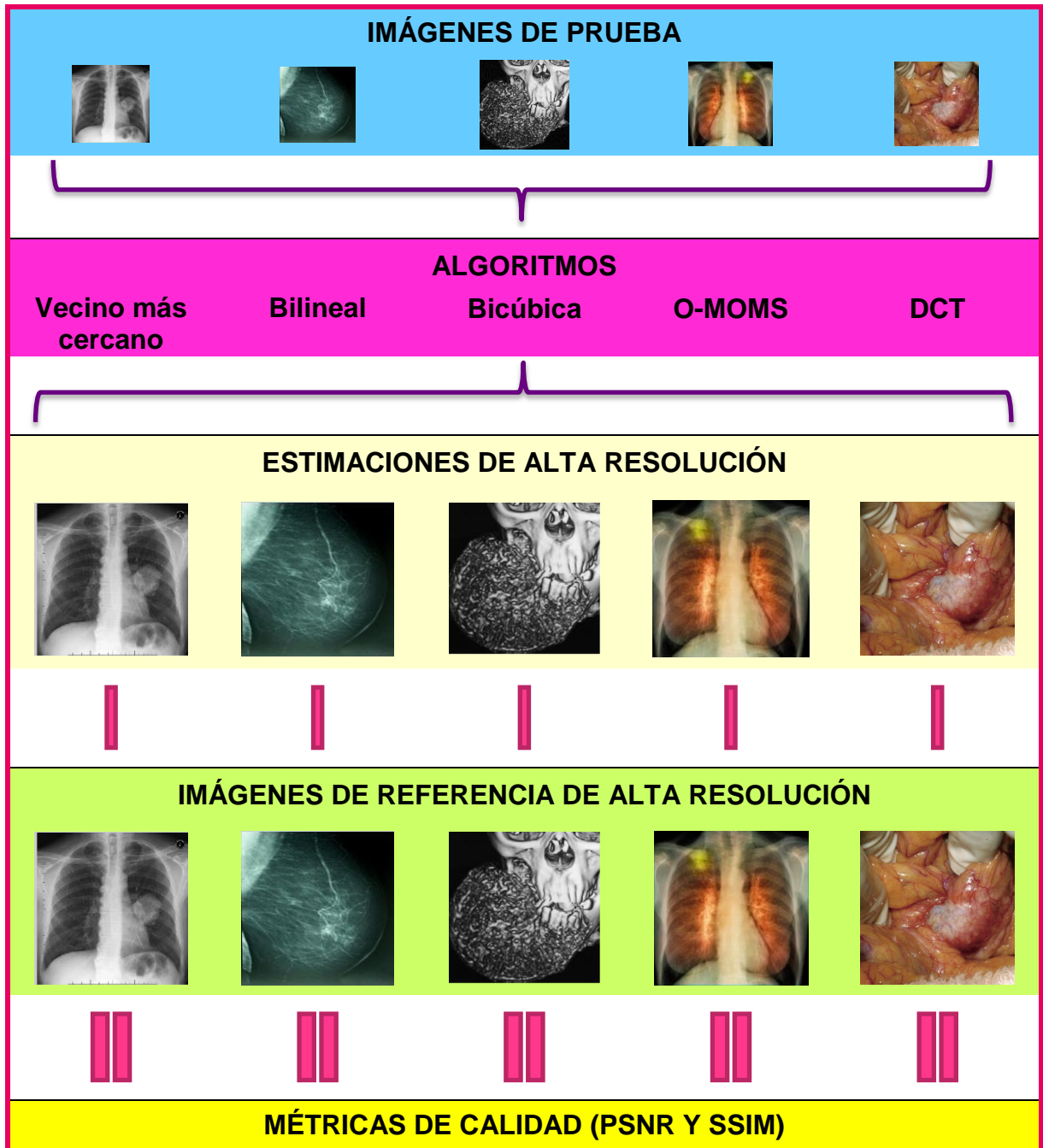


Figura 3.6 Procedimiento de evaluación de los algoritmos.

3.3.1 Relación pico señal a ruido (PSNR).

El PSNR determina la relación de la energía de una señal y el ruido que la afecta, [8] su valor en escala logarítmica está dado en dB cuya expresión matemática está dada por la ecuación 3.2, es una medida relativa de la calidad de imagen y se basa en el error cuadrático medio (MSE) de la ecuación 3.3, el cual se utiliza como medida de desviación del promedio cuadrático para hallar la estimación del valor verdadero, es entonces la medida objetiva más común para comparar el filtrado de una imagen.

$$PSNR = 10 \log_{10} \left[\frac{255^2}{MSE} \right] dB, \quad (3.2)$$

Donde

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [|R(i,j) - \hat{R}(i,j)|], \quad (3.3)$$

Para el caso de imágenes en escala de grises y:

$$\begin{aligned} & MSE \\ &= \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[\frac{|R(i,j) - \hat{R}(i,j)| + |G(i,j) - \hat{G}(i,j)| + |B(i,j) - \hat{B}(i,j)|}{3} \right] \end{aligned} \quad (3.4)$$

Para el caso de imágenes a color con formato RGB.

3.3.2 Índice de Similitud Estructural (SSIM)

El índice de similitud estructural (SSIM) es una medida que pretende cuantificar de forma numérica y automática la calidad visual de una imagen para un observador humano [3]. El SSIM consiste en comparar tres aspectos de una imagen: la luminancia, el contraste y la estructura está definido de la siguiente manera:

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(2\mu_x^2\mu_y^2 + C_1)(2\sigma_x^2\sigma_y^2 + C_2)} \quad (3.5)$$

En donde $\mu_x, \mu_y, \sigma_x^2, \sigma_y^2$ y σ_{xy} son la media, varianza y covarianza de la imagen de referencia y la imagen interpolada. C_1 y C_2 son dos constantes que dependen del rango dinámico de valores de pixel.

Donde la media en un plano (x,y) se define de la siguiente manera:

$$\mu_x = \frac{1}{N} \sum_{n=1}^N X_n \quad (3.6)$$

Para la varianza, de igual manera en un plano (x,y) se define de la siguiente manera:

$$\sigma_x = \sqrt{\frac{1}{N-1} \sum_{n=1}^N (X_n - \mu_x)^2} \quad (3.7)$$

Y para la covarianza en un plano (x,y) la podemos definir de la siguiente manera:

$$\sigma_x = \frac{X - \mu_x}{\sigma_x} \quad \sigma_y = \frac{Y - \mu_y}{\sigma_y} \quad (3.8)$$

3.4 Evaluación de resultados.

A continuación se presentan los resultados obtenidos al simular los métodos de interpolación descritos en las secciones anteriores. En un principio los algoritmos incrementan al doble la dimensión espacial de la imagen de baja resolución, pero se puede realizar un proceso iterativo para aumentarla aún más. Sin embargo, resulta evidente que entre mayor sea el factor de escala mayor será la degradación de la imagen interpolada. Para demostrarlo, a continuación se presentan los resultados obtenidos al redimensionar la imagen al doble y cuatro veces su tamaño original de baja resolución. La tabla 3.1 resume los valores de PSNR y SSIM para cada método utilizado para obtener una estimación con el doble de resolución con respecto a la imagen de baja resolución.

Tabla 3.1 Concentrado de valores PSNR y SSIM para aumentar el doble de resolución de cada método

Imagen	Método	PSNR en dB	SSIM
Pulmón con cáncer	Vecino más Cercano	29.10	0.8786
	Bilineal	29.95	0.9420
	Bicúbica	30.67	0.9173
	O-MOMS	29.32	0.9030
	DCT	31.27	0.8952
Mastografía	Vecino más Cercano	32.36	0.9026
	Bilineal	33.86	0.9752
	Bicúbica	33.97	0.9539
	O-MOMS	36.97	0.9362
	DCT	39.36	0.9701
Tumor en mandíbula	Vecino más Cercano	24.69	0.8580
	Bilineal	26.86	0.9094
	Bicúbica	27.03	0.9157
	O-MOMS	26.91	0.9004
	DCT	26.70	0.9048
Pulmones (Color)	Vecino más Cercano	37.55	0.9145
	Bilineal	34.16	0.9617
	Bicúbica	36.77	0.9567
	O-MOMS	34.62	0.9466
	DCT	40.19	0.9528
Tumor (Color)	Vecino más Cercano	34.58	0.9201
	Bilineal	41.17	0.9782
	Bicúbica	35.59	0.9609
	O-MOMS	31.29	0.9473
	DCT	34.09	0.9227

A partir de los resultados obtenidos, se pudo observar que los resultados varían en los métodos de interpolación, en algunos presentan mejores valores de PSNR, basándonos en los resultados de la tabla 3.1, sin embargo, los valores SSIM indican que las estimaciones bilineal y bicúbica son estructuralmente más exactas.

A partir de los resultados de la tabla, se puede observar que los algoritmos son eficientes si se aplican en imágenes a color, lo cual se debe a que este tipo de imágenes ofrecen dos veces más información en cada pixel en comparación con las imágenes en escala de grises.

Por otra parte, es notable la diferencia entre la imagen original y las imágenes obtenidas después del proceso, es notable el realce de cada detalle, así como el contorno de cada una de las ellas, utilizadas para simulación de los métodos mencionados. Debido a que en estas imágenes tiene una variación en los elementos visuales, los algoritmos introducen una mayor distorsión en el resultado final. Si ampliamos las imágenes cuatro veces su tamaño original de baja resolución, los resultados presentarán un menor desempeño que los que se muestran en este caso para el doble de resolución.

En las figuras 3.7 – 3.31 se muestran gráficamente cada uno de los resultados obtenidos para los cinco métodos y para las cinco imágenes de prueba.



a) Imagen de baja resolución (256 x 256).

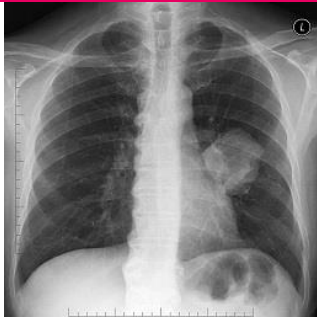


b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método del vecino más cercano (512 x 512).
PSNR = 30.4645 dB
SSIM = 0.8786

Figura 3.7 Estimación mediante el método del vecino más cercano (1:2)



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).

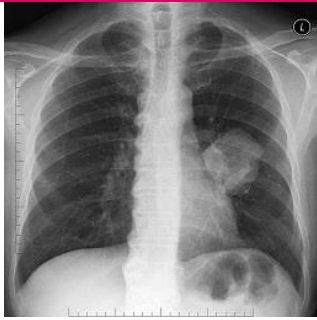


c) Estimación usando el método Bilineal (512 x 512).

PSNR = 33.4456 dB

SSIM = 0.9420

Figura 3.8 Estimación mediante el método de interpolación Bilineal (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).

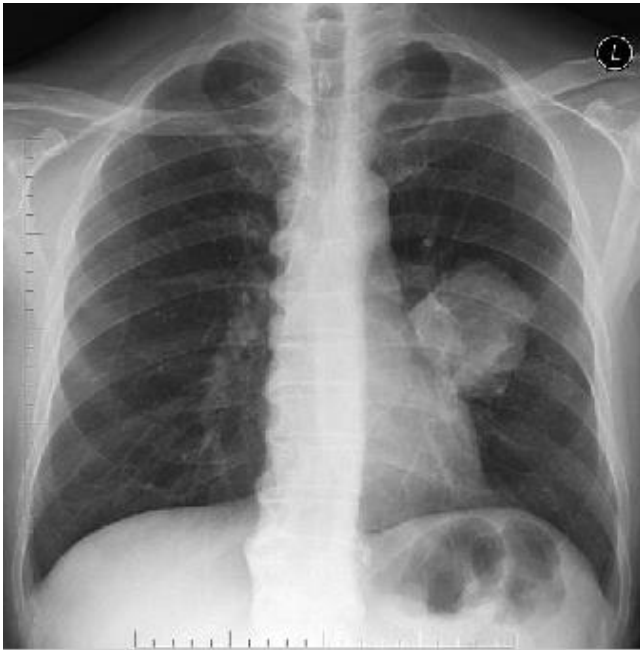


c) Estimación usando el método de interpolación Bicúbica (512 x 512).
PSNR = 30.2644 dB
SSIM = 0.9173

Figura 3.9 Estimación mediante el método de interpolación Bicubica (1:2).



a) Imagen de baja resolución (256 x 256).

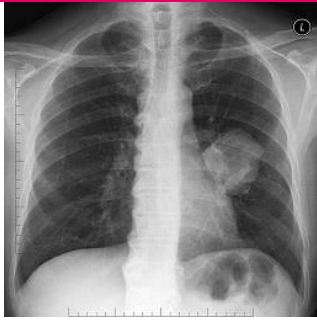


b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación O-MOMS (512 x 512).
PSNR = 30.0494 dB
SSIM = 0.9030

Figura 3.10 Estimación mediante el método de interpolación O-MOMS (1:2).



a) Imagen de baja resolución (256 x 256).

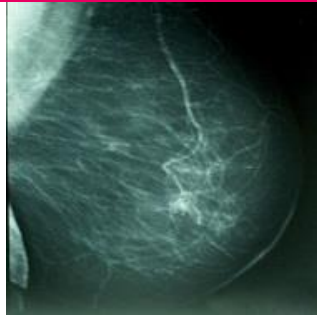


b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación DCT (512 x 512).
PSNR = 31.27 dB
SSIM = 0.8952

Figura 3.11 Estimación mediante el método de interpolación DCT (1:2).



a) Imagen de baja resolución (256 x 256).

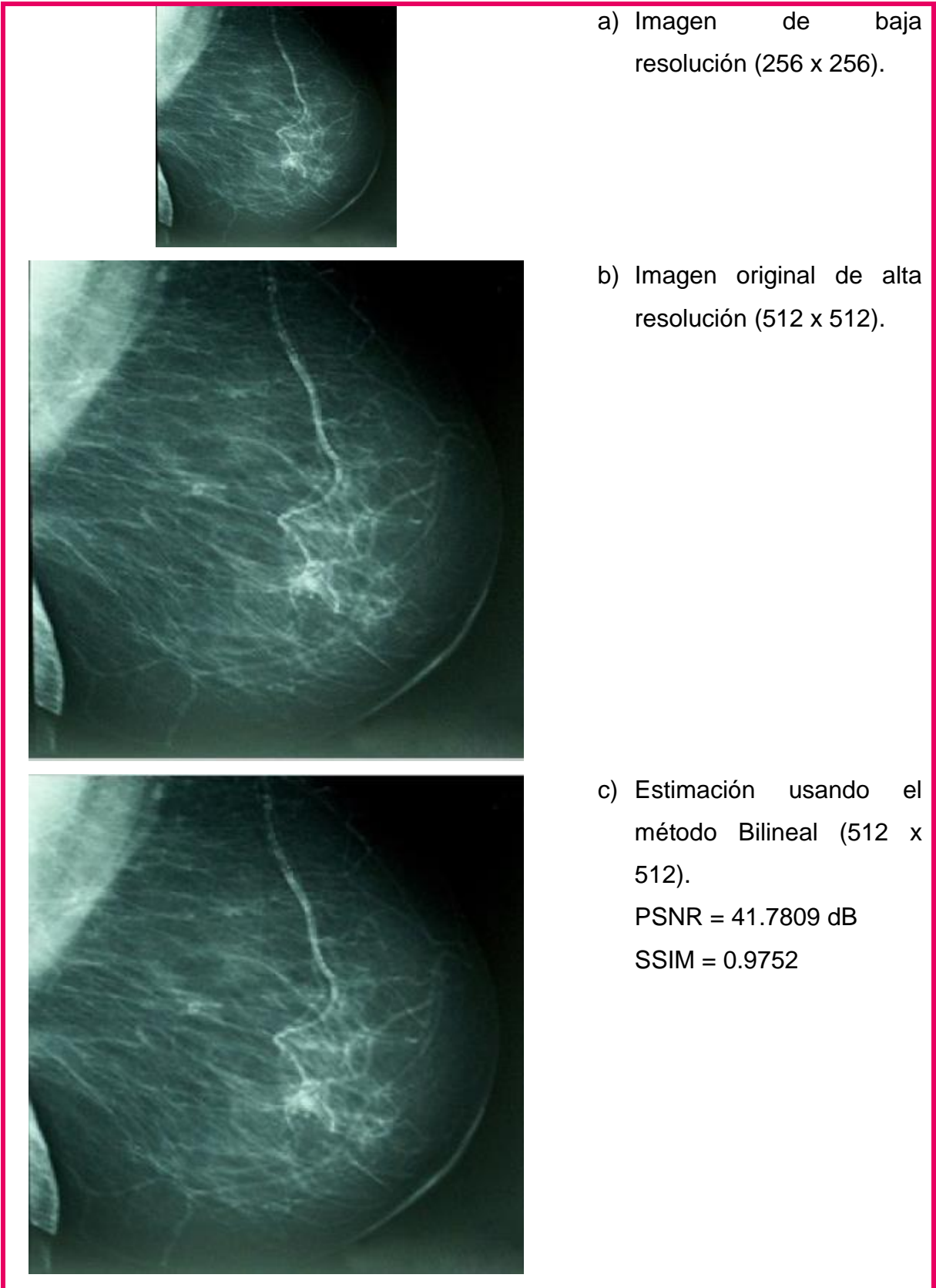


b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método del vecino más cercano (512 x 512).
PSNR = 34.1657 dB
SSIM = 0.9026

Figura 3.12 Estimación mediante el método del vecino más cercano (1:2).



a) Imagen de baja resolución (256 x 256).

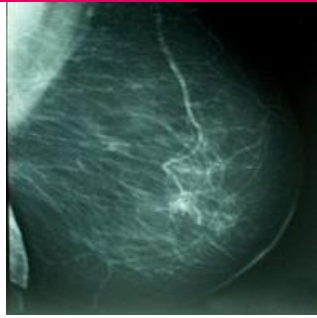
b) Imagen original de alta resolución (512 x 512).

c) Estimación usando el método Bilineal (512 x 512).

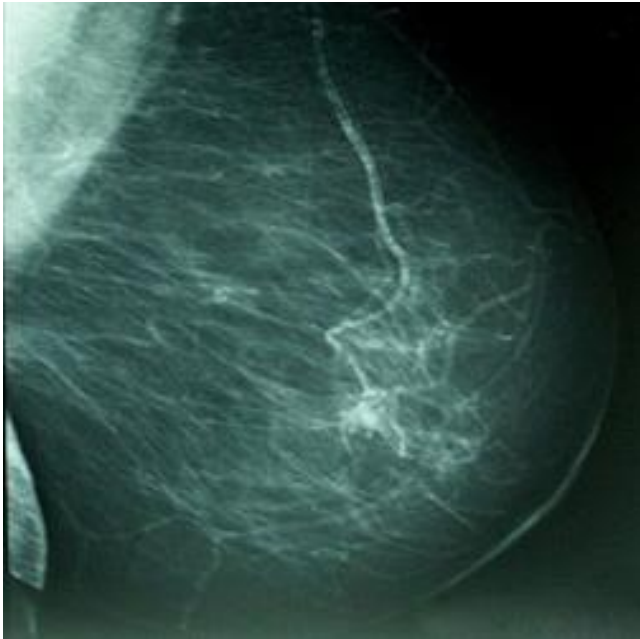
PSNR = 41.7809 dB

SSIM = 0.9752

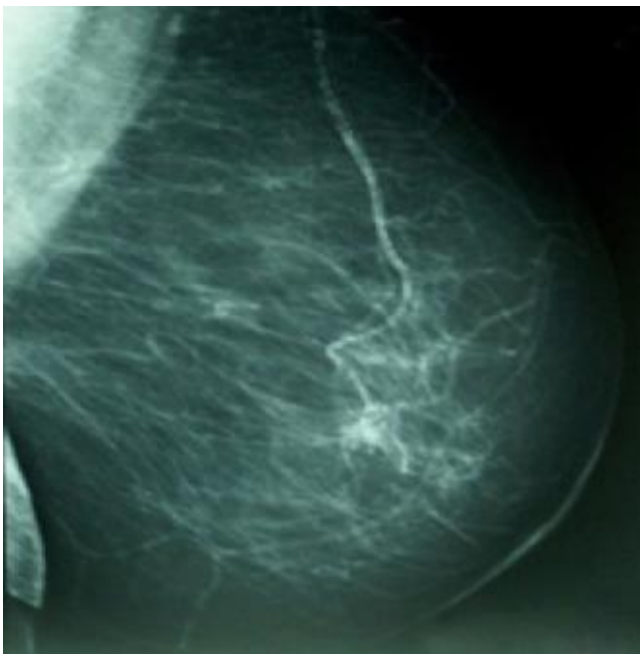
Figura 3.13 Estimación mediante el método de interpolación Bilineal (1:2).



a) Imagen de baja resolución (256 x 256).

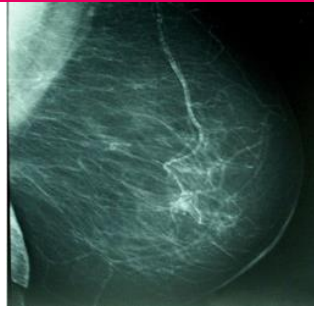


b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación Bicúbica (512 x 512).
PSNR = 37.0390 dB
SSIM = 0.9539

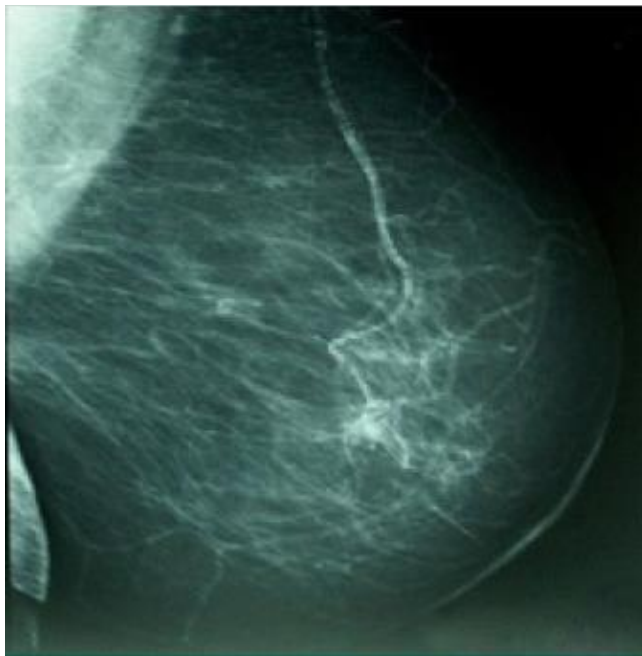
Figura 3.14 Estimación mediante el método de interpolación Bicúbica (1:2)



a) Imagen de baja resolución (256 x 256).

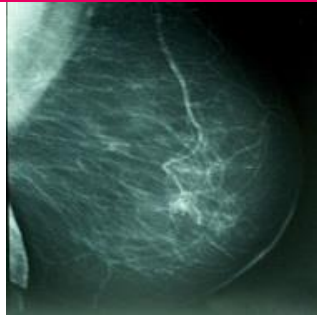


b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación O-MOMS (512 x 512).
PSNR = 36.97 dB
SSIM = 0.9362

Figura 3.15 Estimación mediante el método de interpolación O-MOMS (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación DCT (512 x 512).
PSNR = 39.36 dB
SSIM = 0.9701

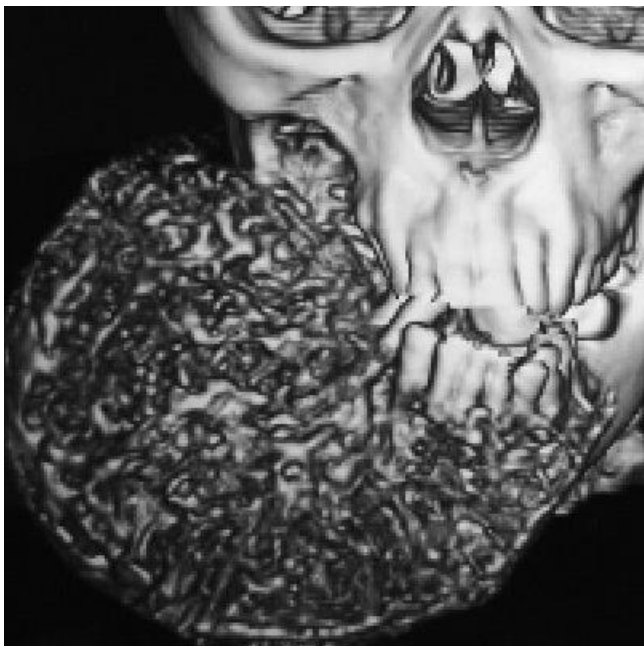
Figura 3.16 Estimación mediante el método de interpolación DCT (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método del vecino más cercano (512 x 512).
PSNR = 24.6961 dB
SSIM = 0.8580

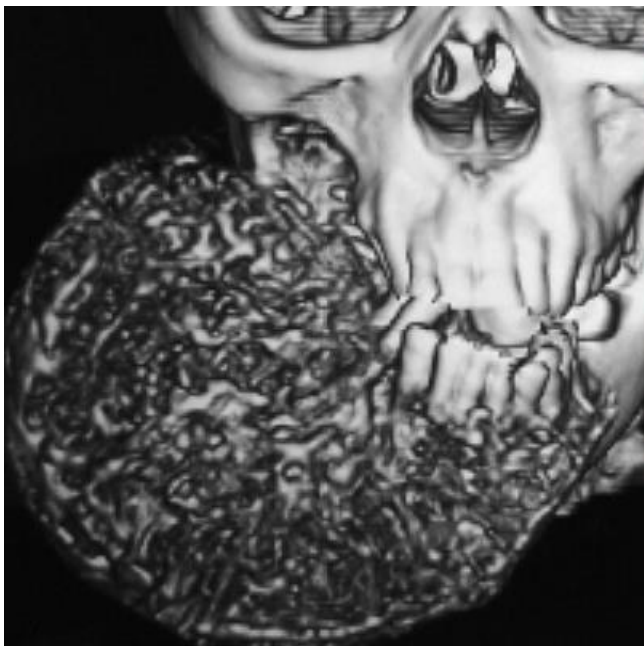
Figura 3.17 Estimación mediante el método de vecino más cercano (1:2)



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método Bilineal (512 x 512).

PSNR = 28.4277 dB

SSIM = 0.9453

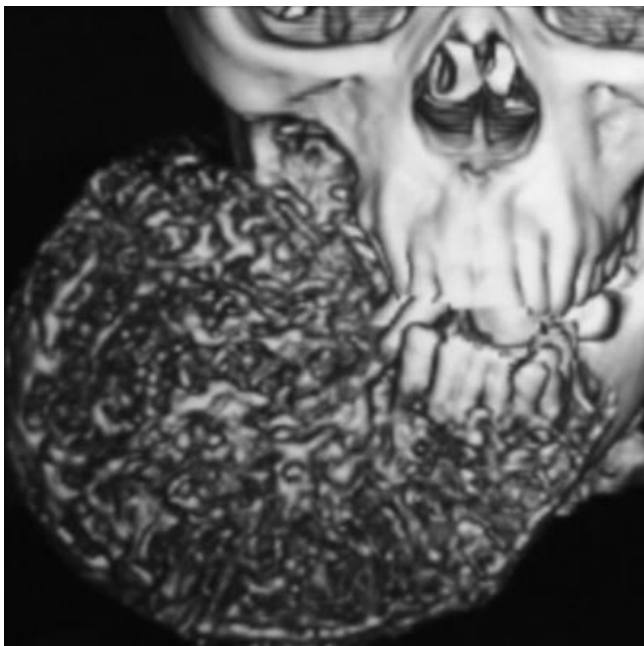
Figura 3.18 Estimación mediante el método de interpolación Bilineal (1:2).



a) Imagen de baja resolución (256 x 256).

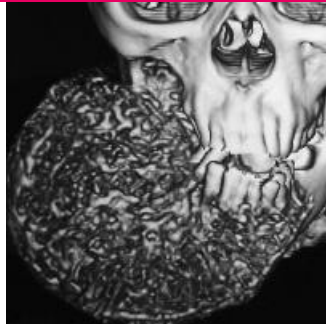


b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación Bicúbica (512 x 512).
PSNR = 27.7297 dB
SSIM = 0.9182

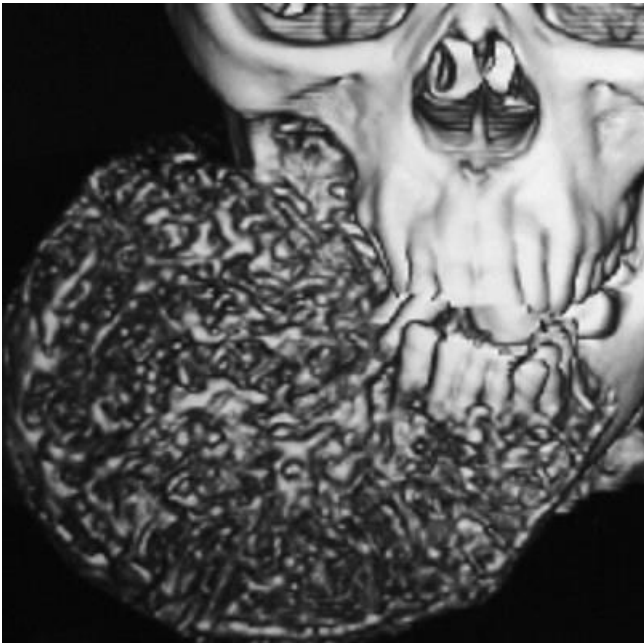
Figura 3.19 Estimación mediante el método de interpolación Bicúbica (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).

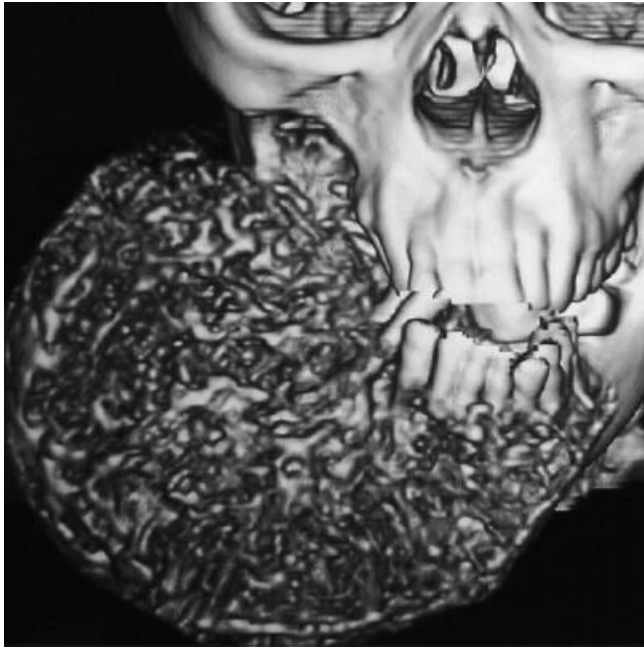


c) Estimación usando el método de interpolación O-MOMS (512 x 512).
PSNR = 26.9179 dB
SSIM = 0.9004

Figura 3.20 Estimación mediante el método de interpolación O-MOMS (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación DCT (512 x 512).
PSNR = 26.7066 dB
SSIM = 0.9048

Figura 3.21 Estimación mediante el método de interpolación DCT (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método del vecino más cercano (512 x 512).
PSNR = 37.5592 dB
SSIM = 0.9145

Figura 3.22 Estimación mediante el método del vecino más cercano (1:2).



a) Imagen de baja resolución (475 x 475x3).



b) Imagen original de alta resolución (950 x 950x3).



c) Estimación usando el método de interpolación Bilineal (512 x 512).
PSNR = 34.1636dB
(43.3694)
SSIM = 0.9617

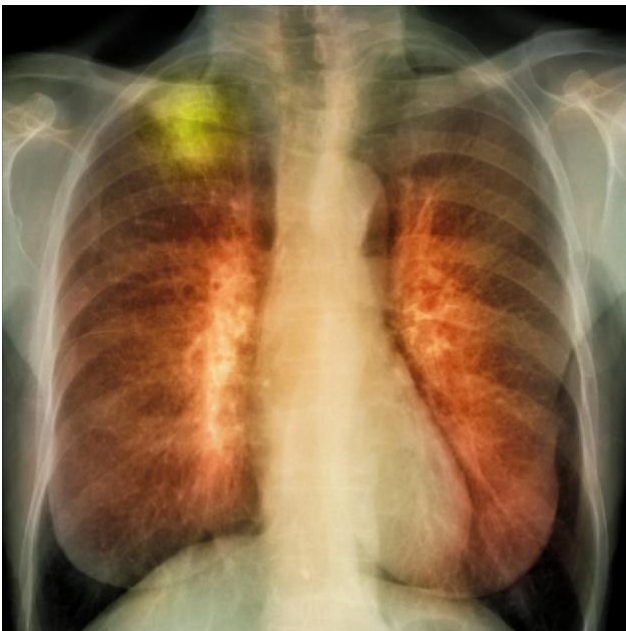
Figura 3.23 Estimación mediante el método de interpolación Bilineal (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación Bicúbica (512 x 512).
PSNR = 36.7770 dB
SSIM = 0.9567

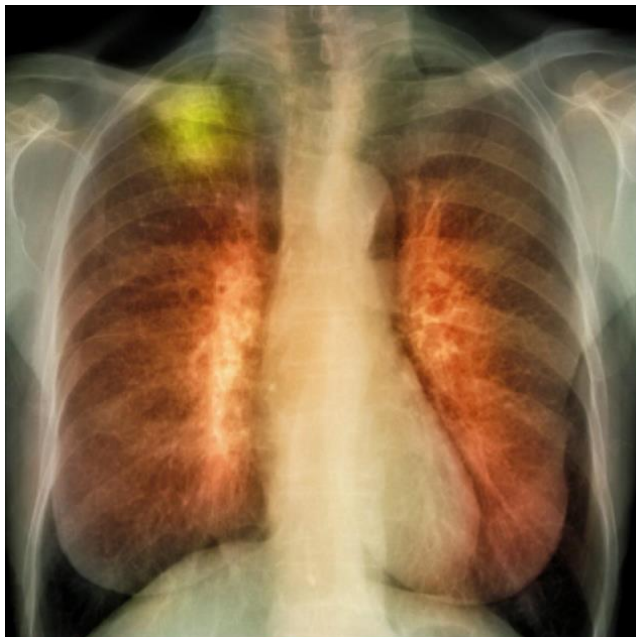
Figura 3.24 Estimación mediante el método de interpolación Bicúbica (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación O-MOMS (512 x 512).
PSNR = 34.6214 dB
SSIM = 0.9466

Figura 3.25 Estimación mediante el método de interpolación O-MOMS (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).

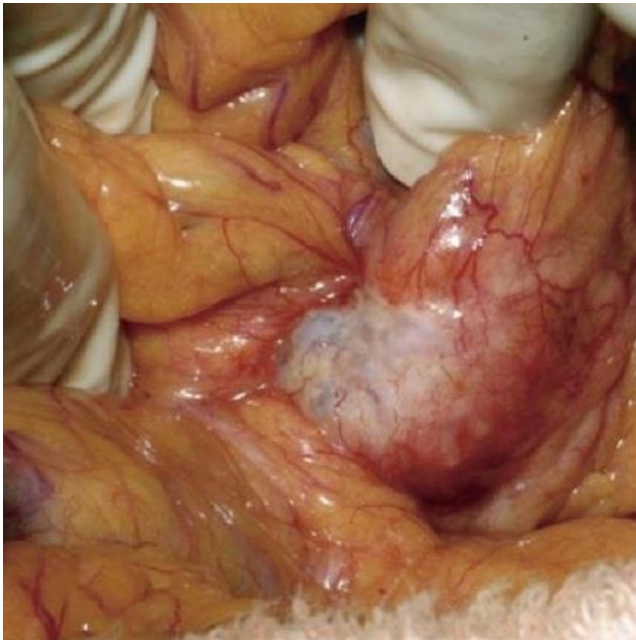


c) Estimación usando el método de interpolación DCT (512 x 512).
PSNR = 40.1902 dB.
SSIM = 0.9528

Figura 3.26 Estimación mediante el método de interpolación DCT (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método del vecino más cercano (512 x 512).
PSNR = 34.58 dB.
SSIM = 0.9201

Figura 3.27 Estimación mediante el método del vecino más cercano (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).

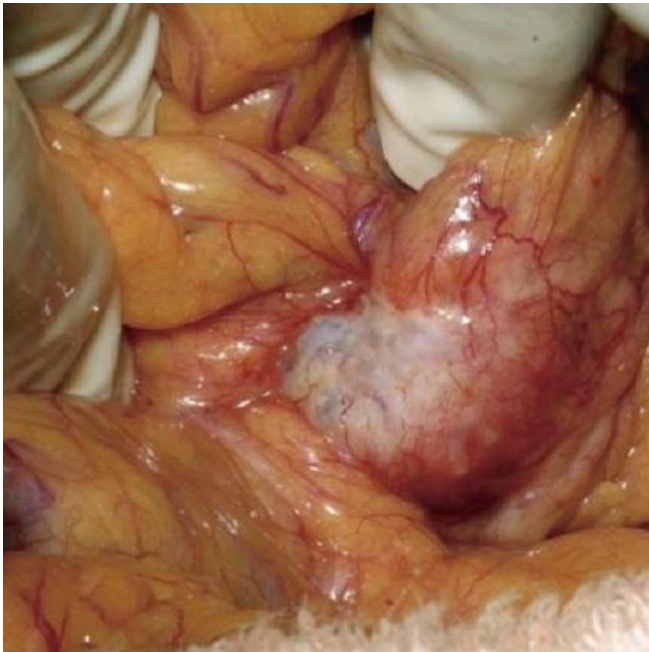


c) Estimación usando el método de interpolación Bilineal (512 x 512).
PSNR = 41.17 dB.
SSIM = 0.9782

Figura 3.28 Estimación mediante el método de interpolación Bilineal (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).

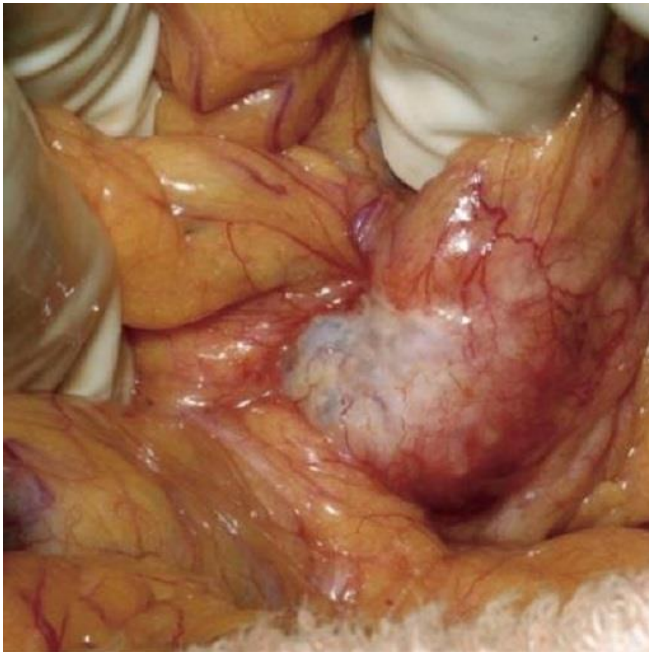


c) Estimación usando el método de interpolación Bicúbica (512 x 512).
PSNR = 35.59 dB.
SSIM = 0.9609

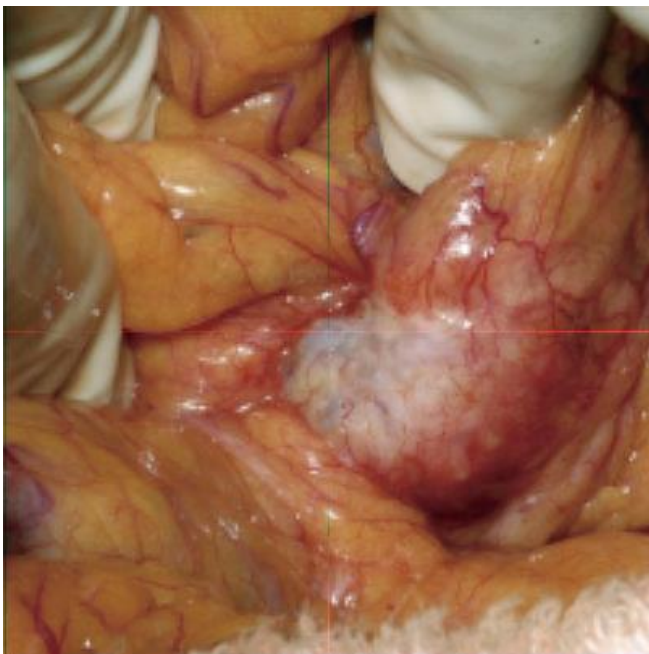
Figura 3.29 Estimación mediante el método de interpolación Bicúbica (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).

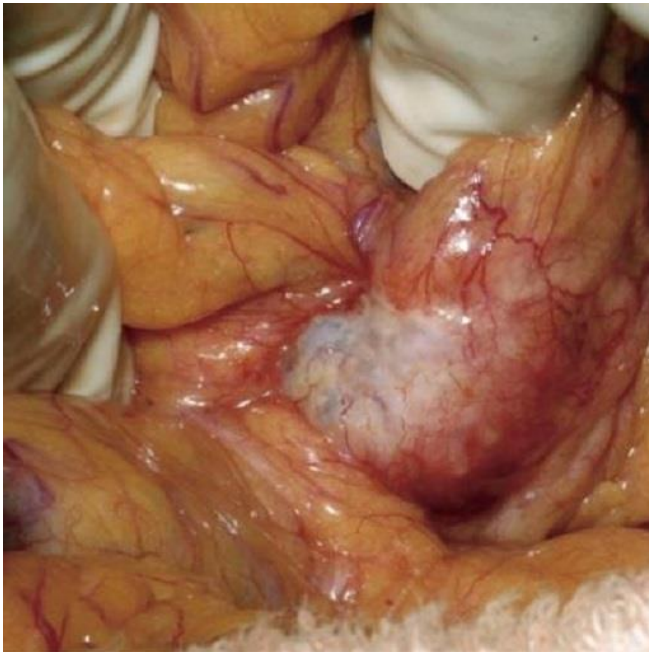


c) Estimación usando el método de interpolación O-MOMS (512 x 512).
PSNR = 31.29 dB.
SSIM = 0.9473

Figura 3.30 Estimación mediante el método de interpolación O-MOMS (1:2).



a) Imagen de baja resolución (256 x 256).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación DCT (512 x 512).
PSNR = 34.09 dB.
SSIM = 0.9227

Figura 3.31 Estimación mediante el método de interpolación DCT (1:2).

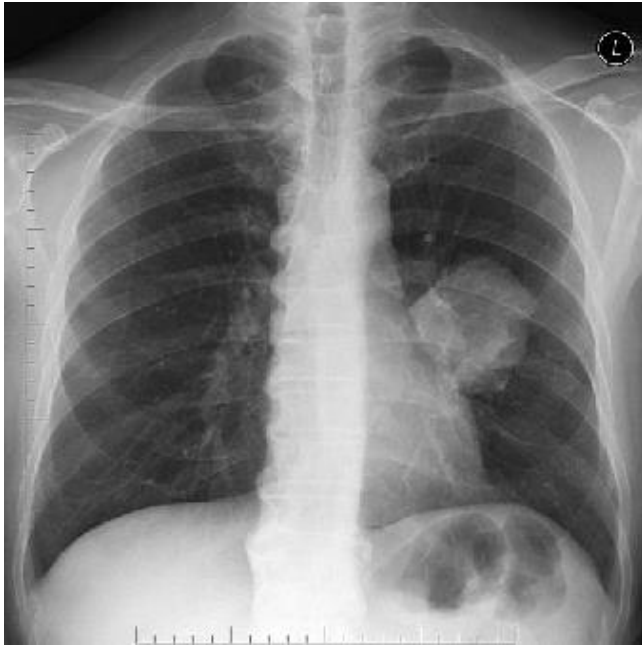
La tabla 3.2 resume los valores de PSNR y SSIM para cada método utilizado para obtener una estimación con una resolución cuatro veces mayor con respecto a la imagen de baja resolución. De la misma forma, en las imágenes 3.32 - 3.56 se muestran los resultados obtenidos para cada método.

Tabla 3.2 Concentrado de valores PSNR y SSIM para aumentar el cuádruple de resolución de cada método

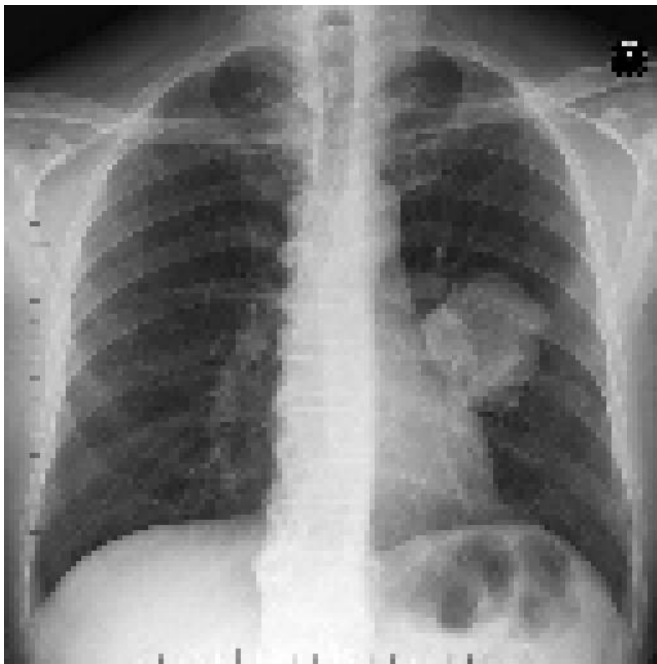
Imagen	Método	PSNR en dB	SSIM
Pulmones con cáncer	Vecino más cercano	27.05	0.7254
	Bilineal	30.17	0.8563
	Bicúbica	26.69	0.8302
	O-MOMS	24.82	0.7621
	DCT	27.52	0.7617
Mastografía	Vecino más cercano	28.75	0.7583
	Bilineal	33.94	0.8879
	Bicúbica	30.55	0.8407
	O-MOMS	29.60	0.8786
	DCT	29.68	0.8027
Tumor en mandíbula	Vecino más cercano	18.22	0.5294
	Bilineal	22.69	0.7786
	Bicúbica	21.17	0.6577
	O-MOMS	21.92	0.7654
	DCT	19.12	0.5714
Pulmones (Color)	Vecino más cercano	33.70	0.8561
	Bilineal	37.31	0.9095
	Bicúbica	33.54	0.8431
	O-MOMS	28.81	0.9050
	DCT	32.90	0.8350
Tumor (Color)	Vecino más cercano	25.66	0.6192
	Bilineal	26.52	0.6808
	Bicúbica	27.89	0.7682
	O-MOMS	25.11	0.8195
	DCT	27.28	0.7273



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).

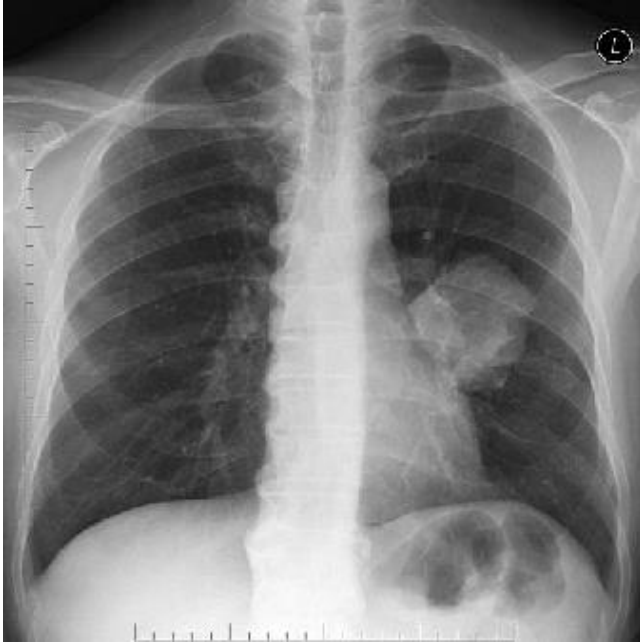


c) Estimación usando el método del vecino más cercano (512 x 512).
PSNR = 27.05 dB
SSIM = 0.7254

Figura 3.32 Estimación mediante el método del vecino más cercano (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación Bilineal (512 x 512).

PSNR = 30.17 dB

SSIM = 0.8563

Figura 3.33 Estimación mediante el método de interpolación Bilineal (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación Bicúbica (512 x 512).

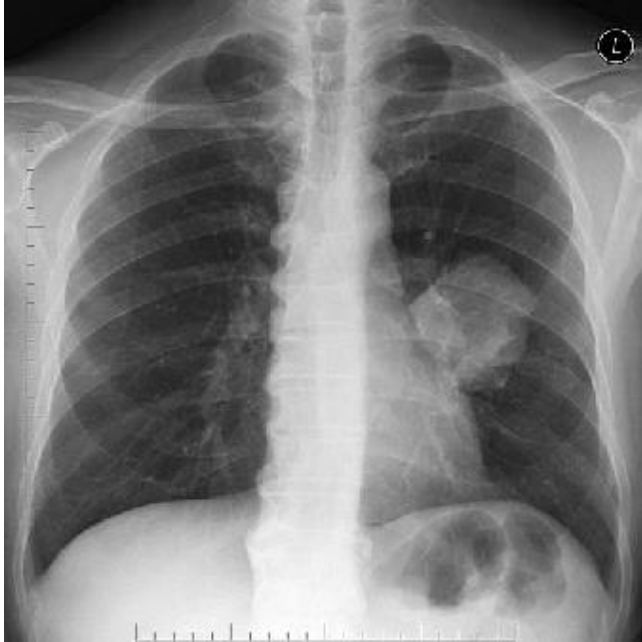
PSNR = 26.69 dB

SSIM = 0.8302

Figura 3.34 Estimación mediante el método de interpolación Bicúbica (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).

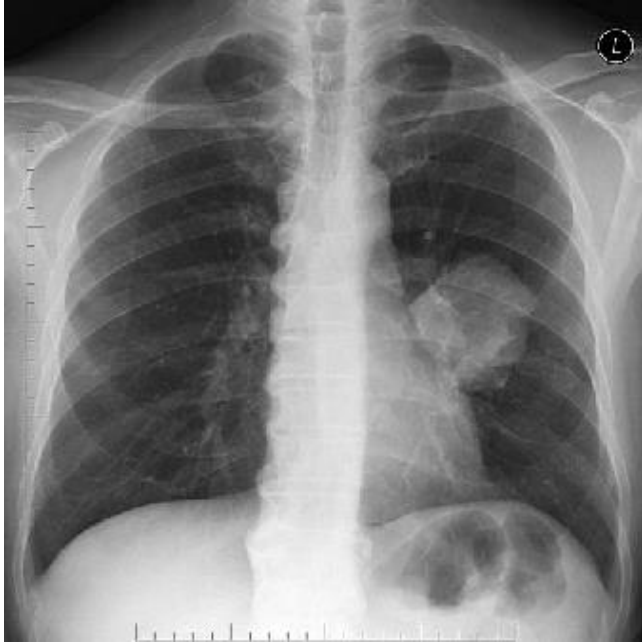


c) Estimación usando el método de interpolación O-MOMS (512 x 512).
PSNR = 24.82 dB
SSIM = 0.762

Figura 3.35 Estimación mediante el método de interpolación O-MOMS (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).

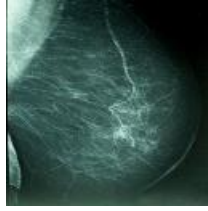


c) Estimación usando el método de interpolación DCT (512 x 512).

PSNR = 27.52 dB

SSIM = 0.7617

Figura 3.36 Estimación mediante el método de interpolación DCT (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).

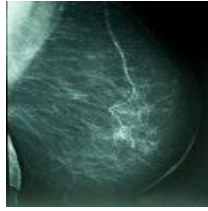


c) Estimación usando el método del vecino más cercano (512 x 512).

PSNR = 28.75 dB

SSIM = 0.7583

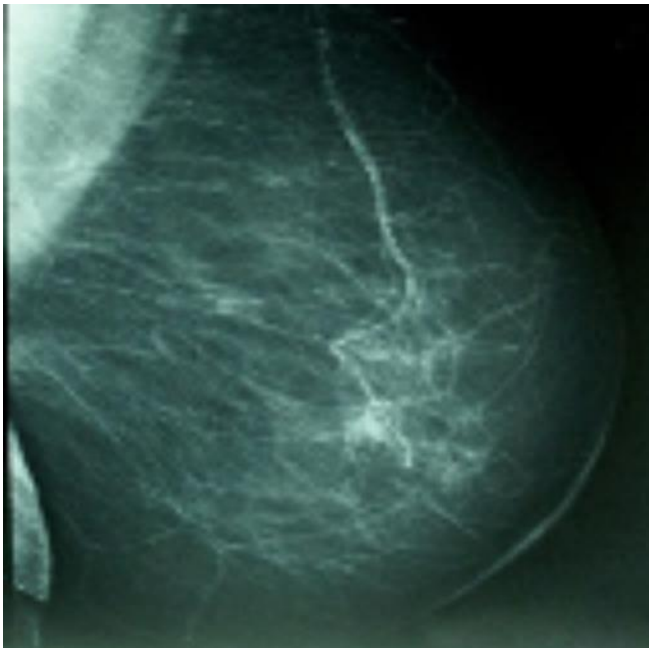
Figura 3.37 Estimación mediante el método del vecino más cercano (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).

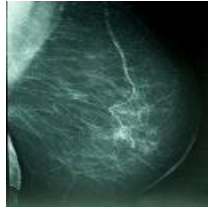


c) Estimación usando el método de interpolación Bilineal (512 x 512).

PSNR = 33.94 dB

SSIM = 0.8879

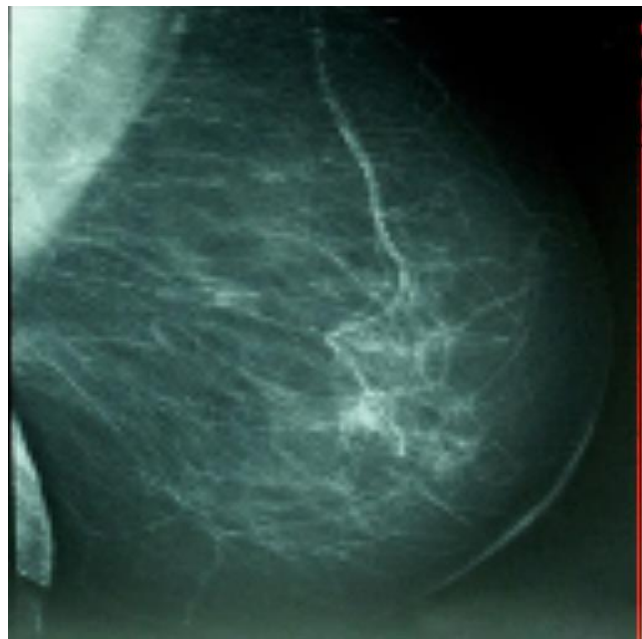
Figura 3.38 Estimación mediante el método de interpolación Bilineal (1:4).



a) Imagen de baja resolución (128 x 128).

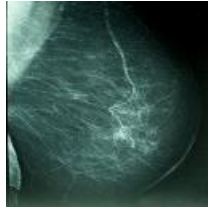


b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación Bicúbica (512 x 512).
PSNR = 30.55 dB
SSIM = 0.8407

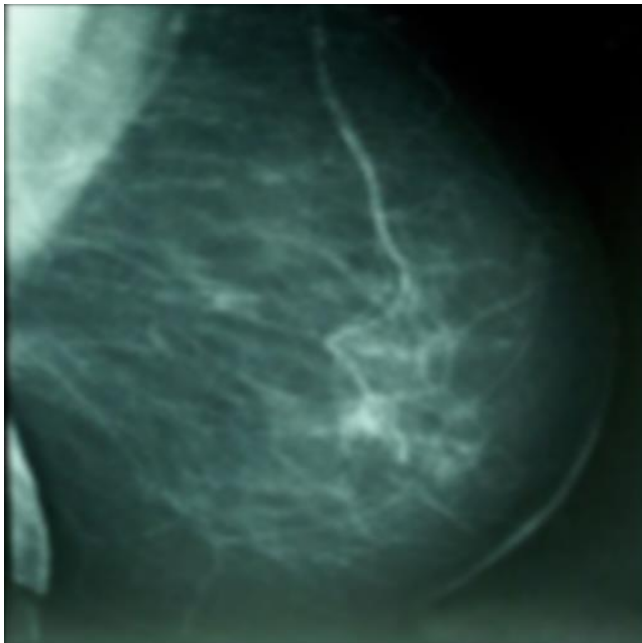
Figura 3.39 Estimación mediante el método de interpolación Bicúbica (1:4).



a) Imagen de baja resolución (128 x 128).

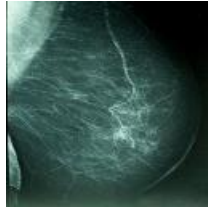


b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación O-MOMS (512 x 512).
PSNR = 29.60 dB
SSIM = 0.8786

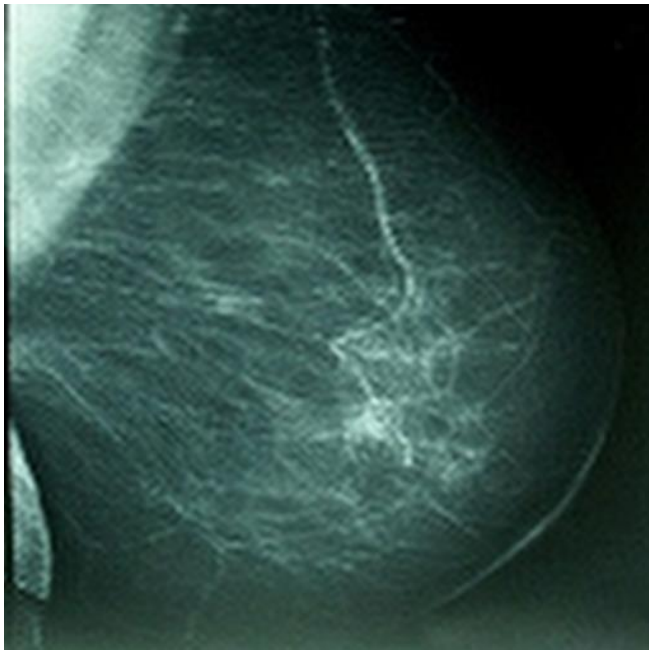
Figura 3.40 Estimación mediante el método de interpolación O-MOMS (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).

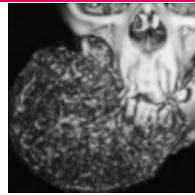


c) Estimación usando el método de interpolación DCT (512 x 512).

PSNR = 29.68 dB

SSIM = 0.8027

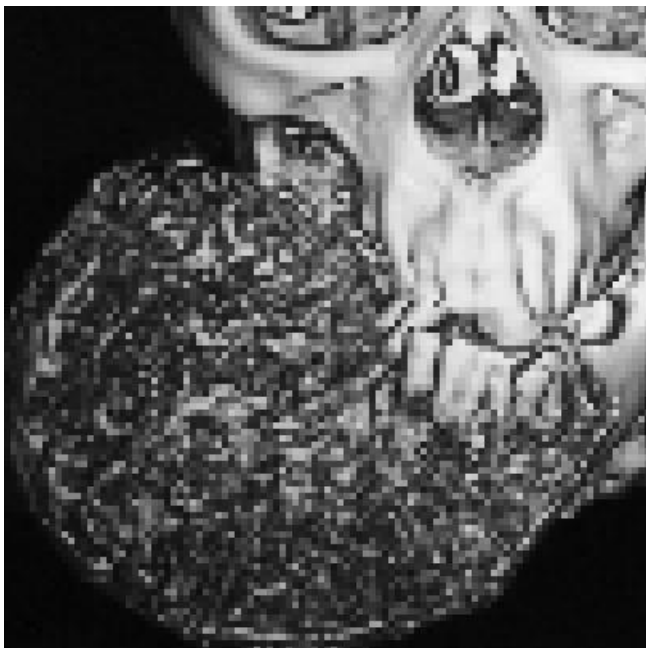
Figura 3.41 Estimación mediante el método de interpolación DCT (1:4).



a) Imagen de baja resolución (128 x 128).

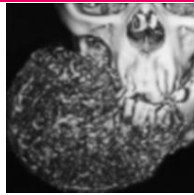


b) Imagen original de alta resolución (512 x 512).

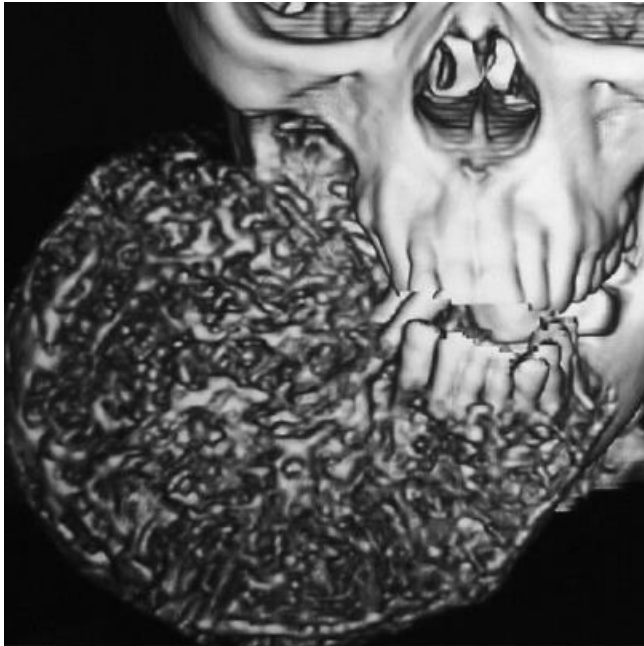


c) Estimación usando el método del vecino más cercano (512 x 512).
PSNR = 18.22 dB
SSIM = 0.5294

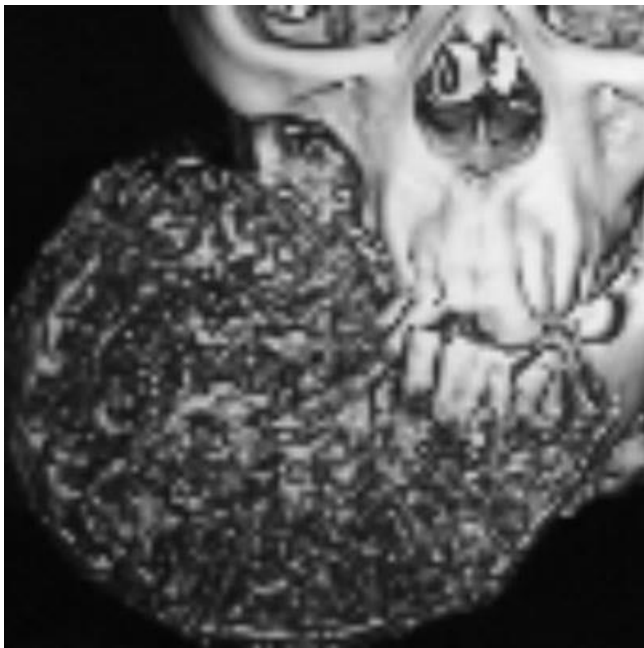
Figura 3.42 Estimación mediante el método del vecino más cercano (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).

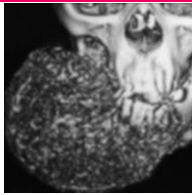


c) Estimación usando el método de interpolación Bilineal (512 x 512).

PSNR = 22.69 dB

SSIM = 0.7786

Figura 3.43 Estimación mediante el método de interpolación Bilineal (1:4).



a) Imagen de baja resolución (128 x 128).

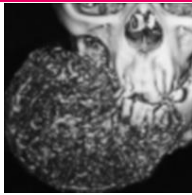


b) Imagen original de alta resolución (512 x 512).

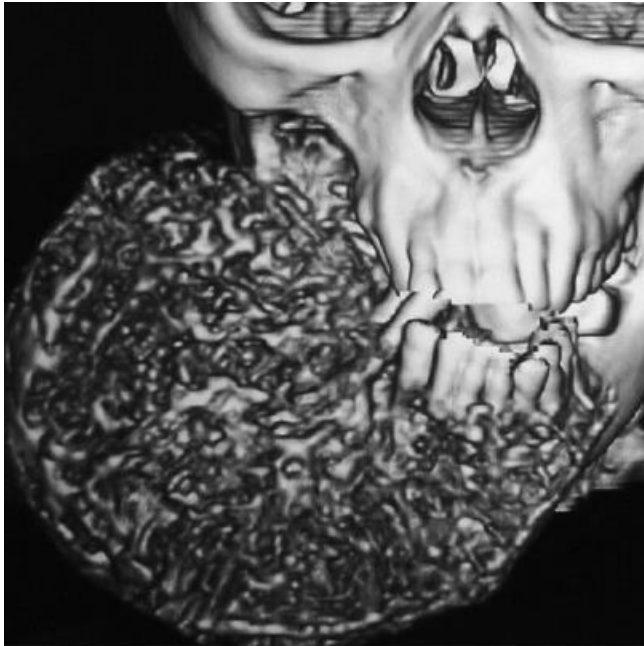


c) Estimación usando el método de interpolación Bicúbica (512 x 512).
PSNR = 21.17 dB
SSIM = 0.6577

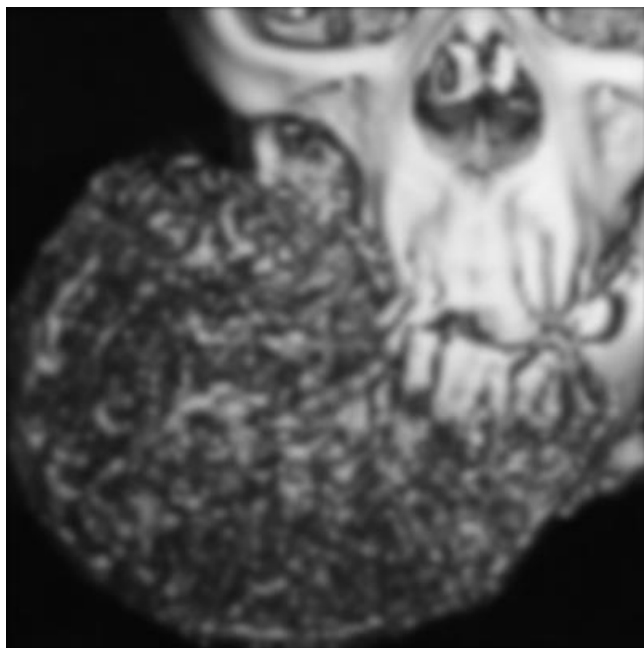
Figura 3.44 Estimación mediante el método de interpolación Bicúbica (1:4).



a) Imagen de baja resolución (128 x 128).

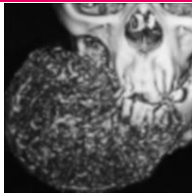


b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación O-MOMS (512 x 512).
PSNR = 21.92 dB
SSIM = 0.7654

Figura 3.45 Estimación mediante el método de interpolación O-MOMS (1:4).



a) Imagen de baja resolución (128 x 128).

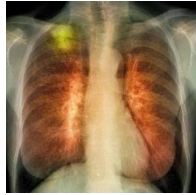


b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación DCT (512 x 512).
PSNR = 19.12 dB
SSIM = 0.5714

Figura 3.46 Estimación mediante el método de interpolación DCT (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).

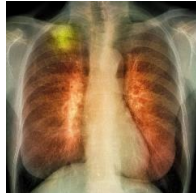


c) Estimación usando el método del vecino más cercano (512 x 512).

PSNR = 33.70 dB

SSIM = 0.8561

Figura 3.47 Estimación mediante el método del vecino más cercano (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación Bilineal (512 x 512).

PSNR = 37.31 dB

SSIM = 0.9095

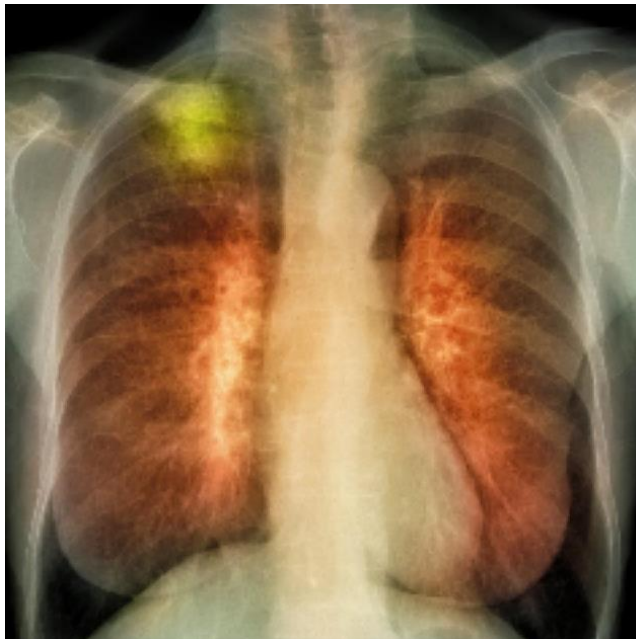
Figura 3.48 Estimación mediante el método de interpolación Bilineal (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).

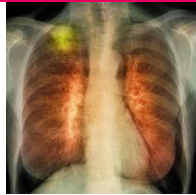


c) Estimación usando el método de interpolación Bicúbica (512 x 512).

PSNR = 33.50 dB

SSIM = 0.8431

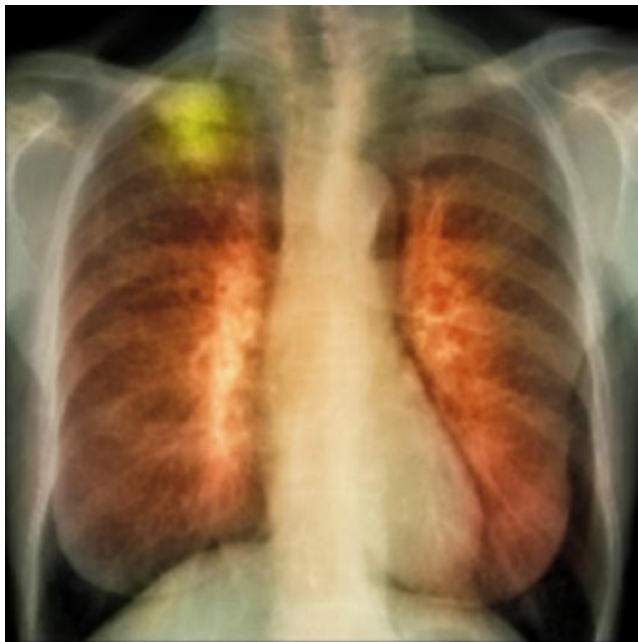
Figura 3.49 Estimación mediante el método de interpolación Bicúbica (1:4)



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación O-MOMS (512 x 512).
PSNR = 28.81 dB
SSIM = 0.9050

Figura 3.50 Estimación mediante el método de interpolación O-MOMS (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación DCT (512 x 512).
PSNR = 32.90 dB
SSIM = 0.8350

Figura 3.51 Estimación mediante el método de interpolación DCT (1:4).



a) Imagen de baja resolución (128 x 128).

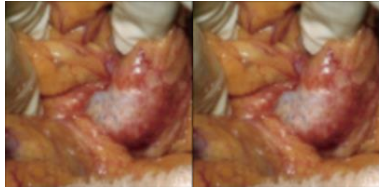


b) Imagen original de alta resolución (512 x 512).

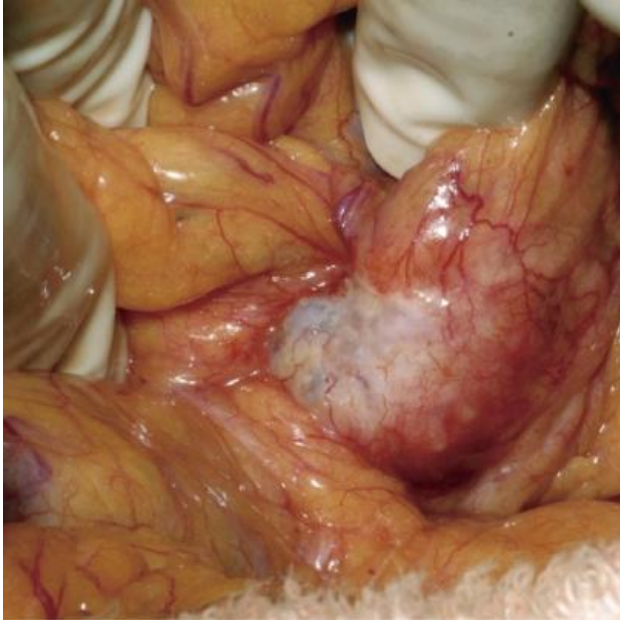


c) Estimación usando el método del vecino más cercano (512 x 512).
PSNR = 25.66 dB
SSIM = 0.6192

Figura 3.52 Estimación mediante el método del vecino más cercano (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación Bilineal (512 x 512).

PSNR = 26.52 dB

SSIM = 0.6808

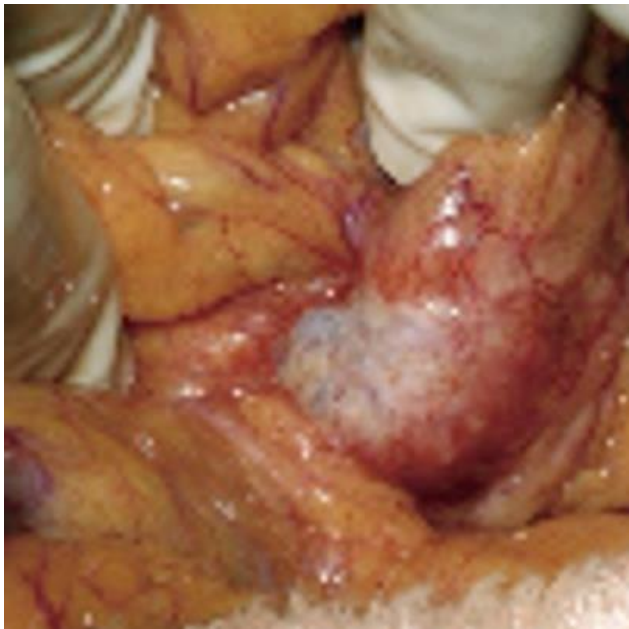
Figura 3.53 Estimación mediante el método de interpolación Bilineal (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).

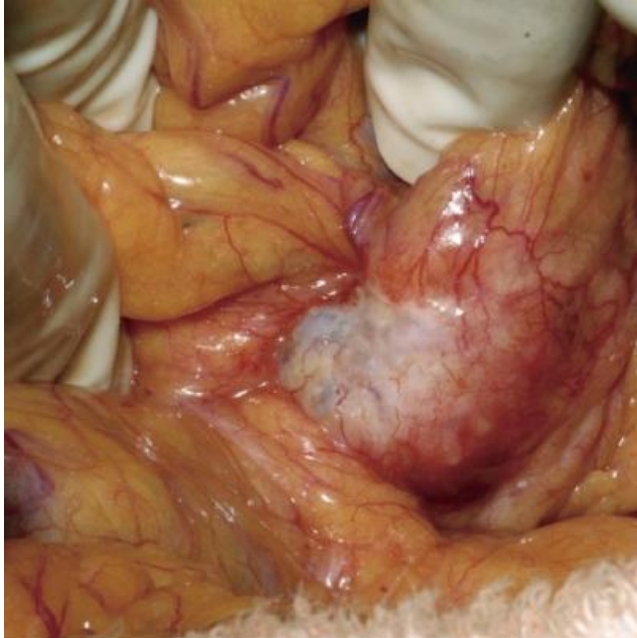


c) Estimación usando el método de interpolación Bicúbica (512 x 512).
PSNR = 27.89 dB
SSIM = 0.7682

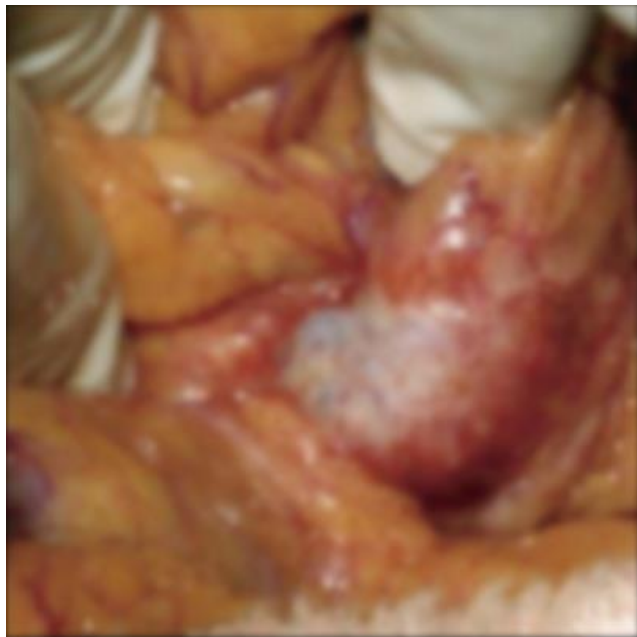
Figura 3.54 Estimación mediante el método de interpolación Bicúbica (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).

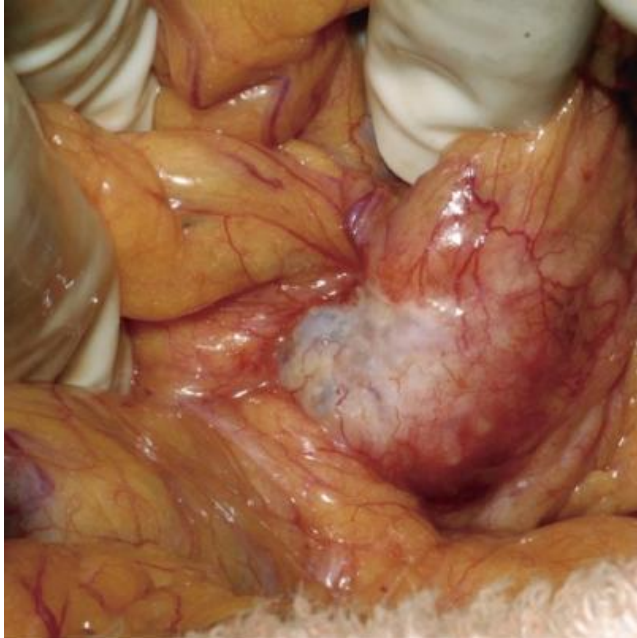


c) Estimación usando el método de interpolación O-MOMS (512 x 512).
PSNR = 25.11 dB
SSIM = 0.8195

Figura 3.55 Estimación mediante el método de interpolación O-MOMS (1:4).



a) Imagen de baja resolución (128 x 128).



b) Imagen original de alta resolución (512 x 512).



c) Estimación usando el método de interpolación DCT (512 x 512).
PSNR = 27.28 dB
SSIM = 0.7273

Figura 3.56 Estimación mediante el método de interpolación DCT (1:4).

Capítulo

4

Evaluación de algoritmos en secuencias de video.

En el capítulo anterior se evaluaron los algoritmos usando solamente imágenes de prueba encontradas comúnmente en los experimentos de procesamiento de imágenes. En este capítulo se evalúan los mismos algoritmos utilizando imágenes tomadas a través de videocámaras utilizadas en la medicina con el fin de evaluar su capacidad de conservar el nivel de detalle de una escena real. En este caso, se consideran solamente dos imágenes, de las cuales se selecciona una región específica de interés sobre la cual se aplican los algoritmos para mejorar su dimensión espacial [5,10].

4.1 Procedimiento de evaluación.

El primer paso fue capturar una serie de imágenes a partir de las secuencias de video [44-48] de la cámara médica. Para evaluar los algoritmos se seleccionaron las imágenes que se muestran en la figura 4.1.

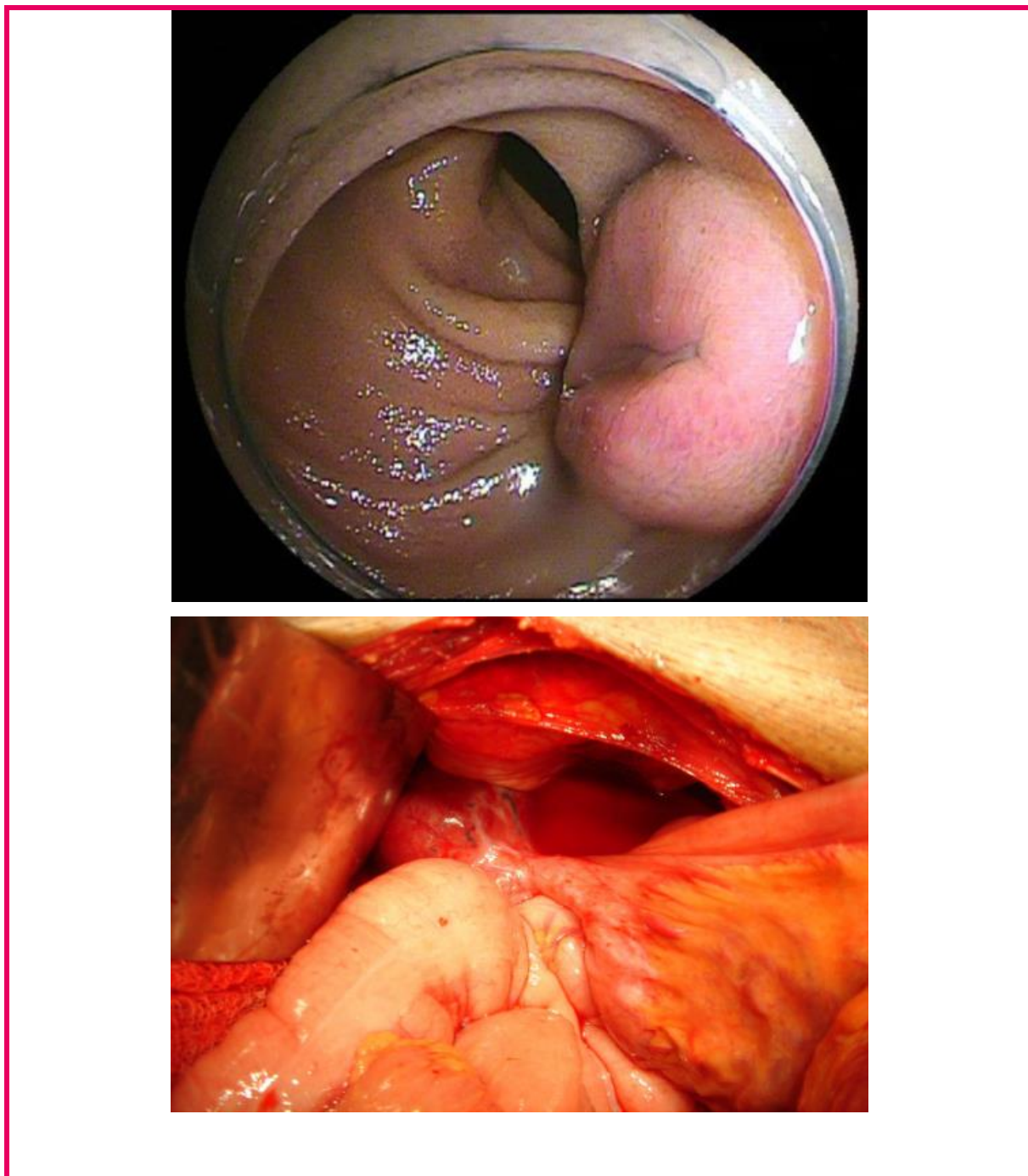


Figura 4.1 Tramas del video para las pruebas

Se podría evaluar cada uno de los algoritmos considerando toda la dimensión espacial de cada una de las imágenes de prueba, como se muestra en la figura 4.1, sin embargo, en la práctica resulta más útil analizar ciertas zonas de la imagen más que otras, principalmente aquellas donde aparecen objetos o personas que requieren ser identificados con mayor precisión.

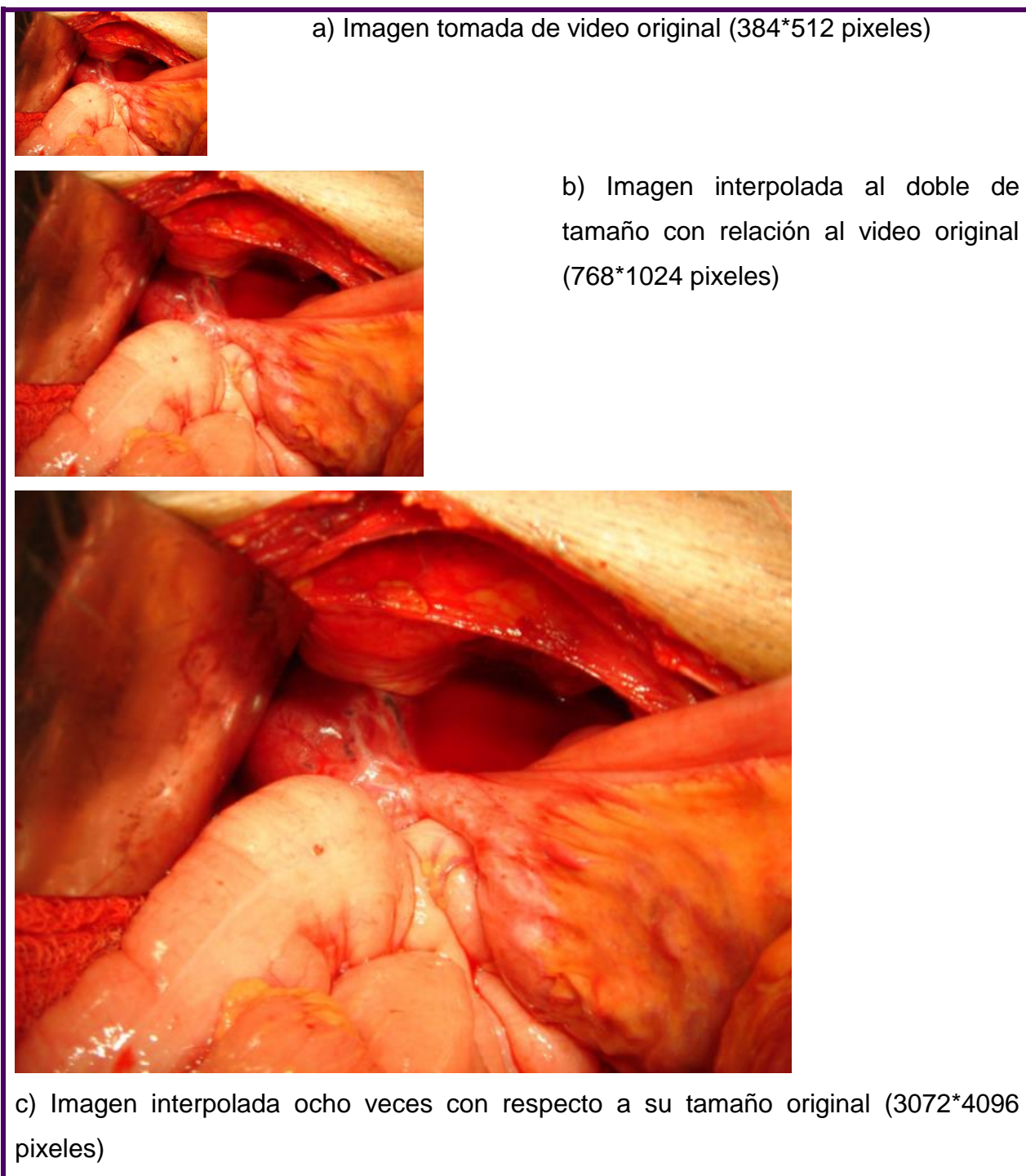


Figura 4.2 Incremento de la dimensión espacial de la escena completa.

Por lo tanto, el siguiente paso del procedimiento fue seleccionar y extraer una región de interés sobre una de las imágenes, como se muestra en la figura 4.3. Estas regiones de interés son entonces asumidas como las imágenes de baja resolución sobre las cuales se aplican los algoritmos para mejorar su dimensión espacial, como puede observarse en la misma figura.

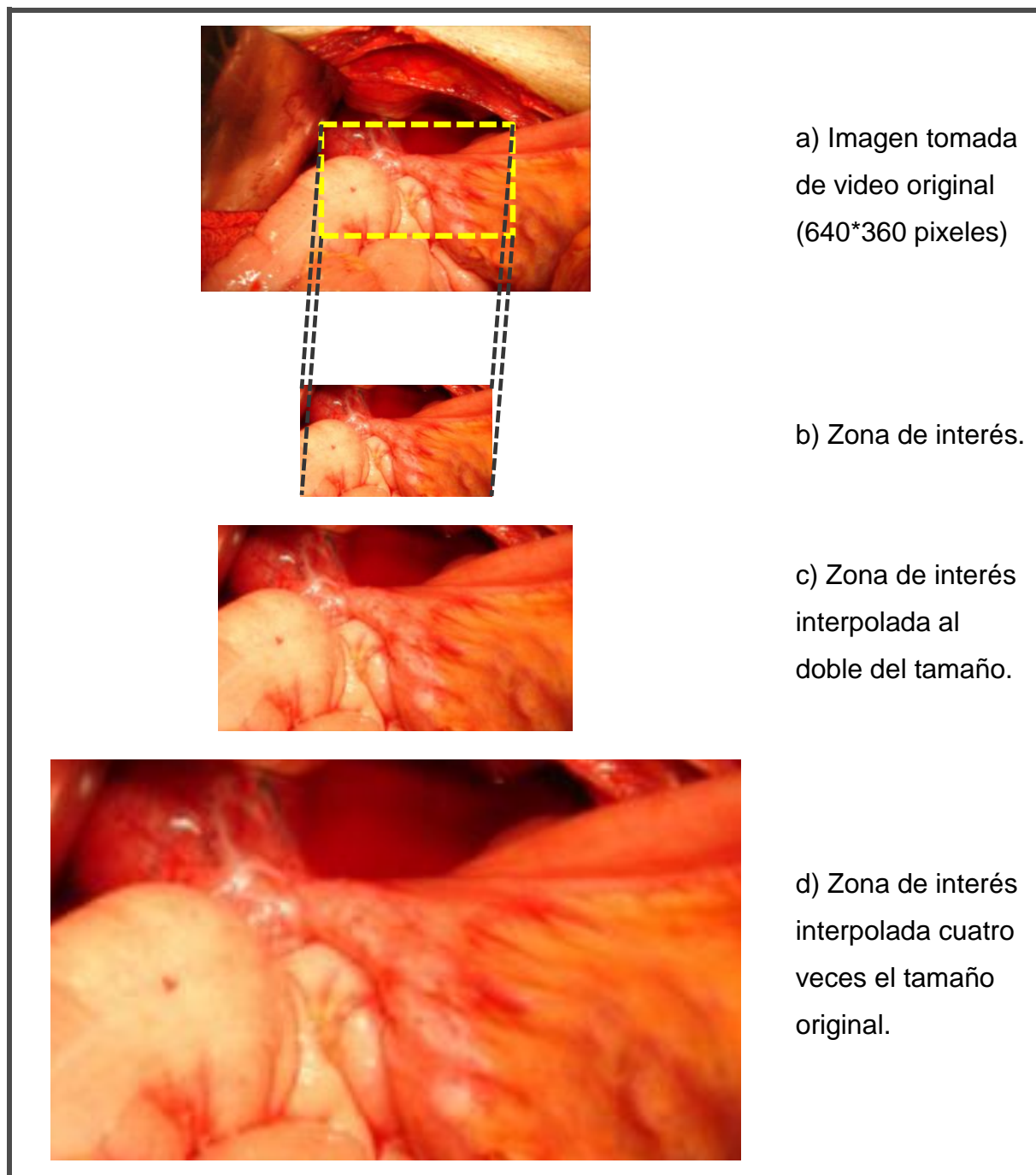


Figura 4.3 Proceso de extracción de la región de interés para su interpolación.

4.2 Evaluación de los algoritmos sobre las regiones de interés.

En las figuras siguientes se presentan los resultados obtenidos al aplicar los algoritmos desarrollados sobre las regiones de interés de cada una de las dos imágenes de prueba. Sin embargo, la evaluación se realiza en forma cualitativa desde su percepción visual, como el PSNR o el SSIM como fue el caso del capítulo anterior.

En primer lugar, la figura 4.4 muestra los resultados obtenidos con el método del vecino más cercano al aumentar la dimensión espacial de la región de interés dos y cuatro veces su tamaño original. Como era previsto, las imágenes resultantes presentan un efecto importante de aliasing y distorsión, lo cual es característico de este método.

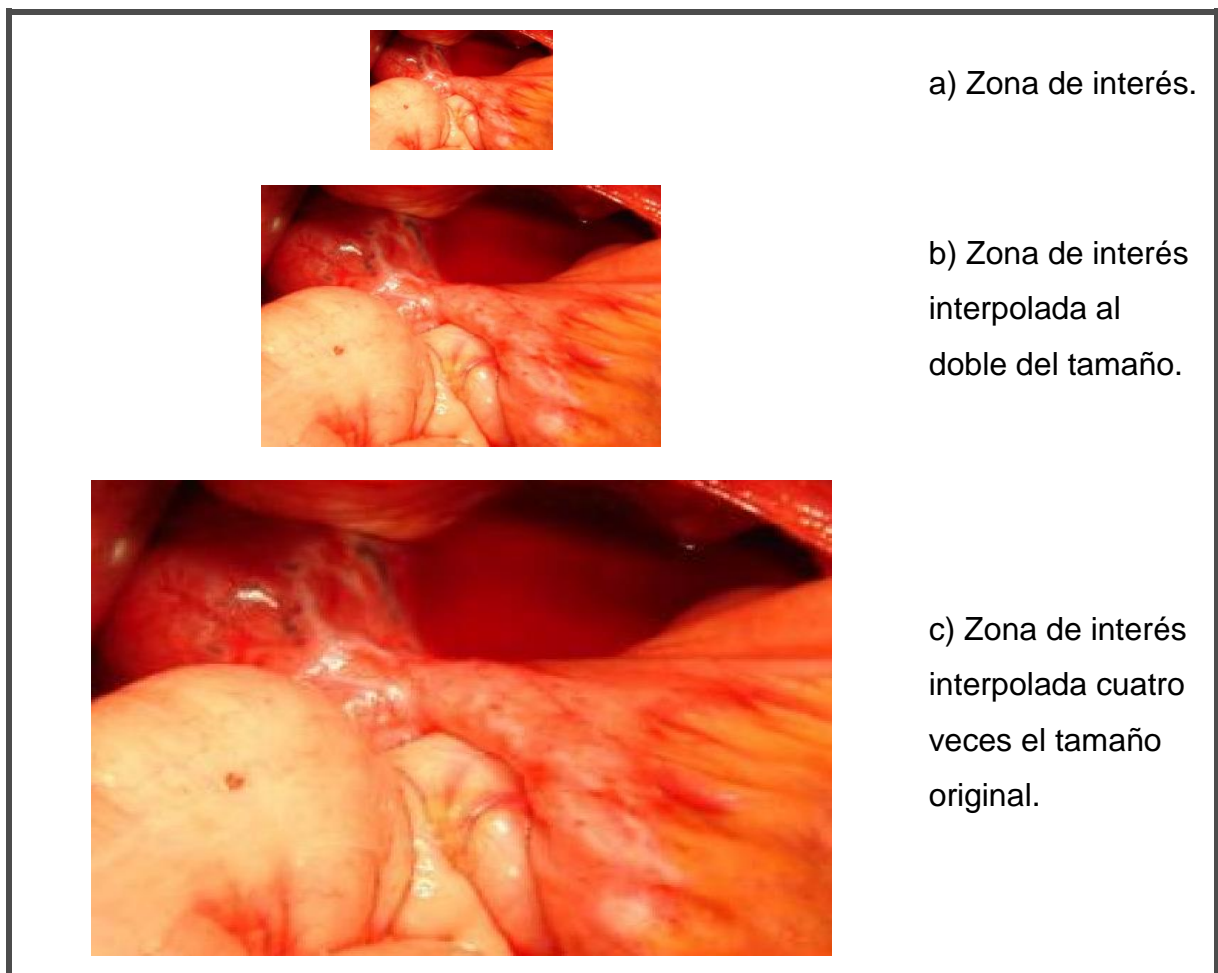


Figura 4.4 Método del vecino más cercano sobre la región de interés de la primera imagen de prueba.

Capítulo

5

Conclusiones y trabajo futuro.

Para finalizar la documentación del trabajo, en este capítulo se presentan las conclusiones y se describe lo que en el futuro podría realizarse como continuación de lo realizado y reportado a lo largo del documento.

5.1 Conclusiones.

Por lo que se puede observar en este documento que describe el análisis e implementación de los métodos y algoritmos de súper-resolución, evaluando su eficiencia en el mejoramiento en la resolución espacial a imágenes y tramas de video. Para ello se abordó la técnica de interpolación como parte esencial del proceso general de súper-resolución y se presentaron sus fundamentos matemáticos y sus principales métodos. Posteriormente se describió con detalle las técnicas clásicas de interpolación, polinomial y basada en una transformación de dominio. La primera de ellas permitió implementar cuatro métodos: el del vecino más cercano, la interpolación bilineal, la interpolación bicúbica y el de interpolación O-MOMS. La segunda técnica sirvió para implementar el método de interpolación basado en la Transformada Coseno Discreta. Y para ello se realizó el diseño de los algoritmos de cada método y codificándose en lenguaje de MatLab, para poder calcular el PSNR y SSIM, a cada una de las imágenes de prueba, es por ello que se dio el caso que algunos resultados obtenidos durante la simulación de los algoritmos, se observó que dichos resultados comparados en las tablas 3.1 y 3.2, que resaltaban estas métricas de calidad balanceándose más hacia los métodos bilineal y bicúbico.

El método de interpolación del vecino más cercano es un algoritmo que requiere de poca información por lo que el tiempo de procesamiento es muy rápido. Su desventaja es que los resultados presentan deformaciones en forma de dientes en los bordes y detalles finos, así como ruido y distorsión en las imágenes finales.

La interpolación bilineal requiere de más información debido a que requiere un proceso más complejo que el vecino más cercano, en cuanto a los resultados, estos representan una mejoría que el método anteriormente mencionado, debido a que el fenómeno de dentado se suprime haciendo la preservación de bordes y detalles finos mejor que el algoritmo del vecino más cercano.

La interpolación bicúbica, es el método más conocido entre la familia de los métodos de interpolación polinomial, por lo que se pudo observar esta interpolación es similar a la bilineal, solo que en lugar de tomar los 4 píxeles más cercanos, tomo los valores de los 16 píxeles más cercanos para promediar y encontró el valor que se debe

asignar al pixel desconocido, es así que este método por tomar mayor número de valores a diferentes distancias del pixel desconocido, logra mayor nitidez sobre la imagen de salida.

La técnica O-MOMS se definió como función base o kernels, donde se utilizó la combinación lineal de versiones desplazadas de una función base, el cual tiene un soporte mínimo, para una precisión dada (también conocida como orden de aproximación). Esta clase de funciones a la que denominan funciones de orden mínimo de soporte (MOMS). Se formó por combinaciones lineales de la B-Spline del mismo orden y sus derivadas.

La técnica basada en la Transformada Coseno Discreto (DCT), representa una imagen como la suma sinusoides de distintas amplitudes y frecuencias, donde obtuvo una imagen de dimensiones $M \times N$, que por lo general, está técnica se utilizó también para llevar la compresión de imágenes, al mismo tiempo fue útil para estimar la información perdida que tiene una imagen de baja resolución y así poder reconstruir la imagen de alta resolución.

5.2 Trabajo a futuro

Aunque cada método posee ventajas y desventajas, para la interpolación de imágenes, nunca está demás seguir la investigación sobre las técnicas utilizadas en dicho proyecto, así como se mencionó seguir con el estudio de los anteriores métodos, sería posible explotar otros métodos de interpolación más avanzados y complejos, con el fin de poder realizar pruebas en diferentes tipos de imágenes como YCbCr o YPbPr entre otras, también en diversos formatos de video y lograr la codificación en distintos tipos de lenguajes.

Anexo 1. Códigos de programación ejecutados sobre MatLab.

% Programa para evaluar el método de interpolación del Vecino más Cercano

```
%-----RESET-----  
  
clear all;  
  
clc  
  
%-----IMAGEN ORIGINAL-----  
  
[NombreImagen RutaImagen]=uigetfile({  
    '*.jpeg;*.jpg;*.tif', 'Imagen JPEG (*.jpg, *.jpeg, *.tif);  
    '*.bmp', 'Imagen BMP (*.bmp)'},'MultiSelect','off');  
  
if ~isequal(RutaImagen,0) && ~isequal(NombreImagen,0)  
    PathCompletoImagen=strcat(RutaImagen,NombreImagen);  
    Imagen=imread(PathCompletoImagen);  
  
end  
  
%-----INTERPOLACION AL DOBLE DE RESOLUCIÓN---  
  
Zoom = 2;  
  
%-----OBTENCION DE IMAGEN DE LR (DOWNSAMPLING)  
  
ImagenLR=DimensionamientoLR(Imagen);  
  
%-----OBTENCION DE LA REJILLA DE HR PARA INTERPOLACION  
  
% c(x)  
  
if isrgb(Imagen)==1  
    G=DimensionamientoHR(ImagenLR(:, :, 2), Zoom);  
    B=DimensionamientoHR(ImagenLR(:, :, 3), Zoom);  
end  
  
R=DimensionamientoHR(ImagenLR(:, :, 1), Zoom);  
  
%-----KERNEL DE INTERPOLACIÓN-----
```

```

Kx = [1 1] %(Este es Bo)

Tam=size(ImagenLR);

m=Tam(1);

n=Tam(2);

g_x=zeros(m*Zoom,n*Zoom);

if isrgb(Imagen)==1

    g_x=zeros(m*Zoom,n*Zoom);

    for i=1:m*Zoom

        convolucion_fila=conv(G(i,:),Kx,'same');

        g_x(i,:)=convolucion_fila;

    end

    for i=1:n*Zoom

        convolucion_col=conv(g_x(:,i)',Kx,'same');

        g_x(:,i)=convolucion_col';

    end

    G=g_x;

    G(m*2,:)=G(m*2-1,:);

    G(:,n*2)=G(:,n*2-1);

    g_x=zeros(m*Zoom,n*Zoom)

    for i=1:m*Zoom

        convolucion_fila=conv(B(i,:),Kx,'same');

        g_x(i,:)=convolucion_fila;

    end

    for i=1:n*Zoom

        convolucion_col=conv(g_x(:,i)',Kx,'same');

        g_x(:,i)=convolucion_col';

```

```

end

B=g_x;

B(m*2,:)=B(m*2-1,:);

B(:,n*2)=B(:,n*2-1);

end

g_x=zeros(m*Zoom,n*Zoom)

for i=1:m*Zoom

    convolucion_fila=conv(R(i,:),Kx,'same');

    g_x(i,:)=convolucion_fila;

end

for i=1:n*Zoom

    convolucion_col=conv(g_x(:,i)',Kx,'same');

    g_x(:,i)=convolucion_col';

end

R=g_x;

R(m*2,:)=R(m*2-1,:);

R(:,n*2)=R(:,n*2-1);

Interpol(:,:,1)=uint8(R);

[ssimR ssim_map] = ssim_index(uint8(Imagen(:,:,1)),uint8(R));

ssim_1 = ssimR;

if isrgb(Imagen)==1

    Interpol(:,:,2)=uint8(G);

    Interpol(:,:,3)=uint8(B);

    [ssimG ssim_map] = ssim_index(uint8(Imagen(:,:,2)),uint8(G));

    [ssimB ssim_map] = ssim_index(uint8(Imagen(:,:,3)),uint8(B));

    ssim_1=(ssimR+ssimG+ssimB)/3

```

```

end

valor_psnr = Calcula_psnr(Imagen,Interpol)

imtool(Imagen);

imtool(uint8(Interpol));

str = {'Vecino más cercano';'Bilineal';'Bicúbica'};

[seleccion,v] = listdlg('PromptString','Select a file:',...
    'SelectionMode','single',...
    'ListString',str)

switch seleccion

case 1

    if isrgb(ImagenLR)==1

        G=uint8(MasCercano(double(G),double(ImagenLR(:,2)),Zoom));

        B=uint8(MasCercano(double(B),double(ImagenLR(:,3)),Zoom));

    end

    R=uint8(MasCercano(double(R),double(ImagenLR(:,1)),Zoom));

case 2

    if isrgb(Imagen)==1

        G=uint8(Bilineal(double(G),double(ImagenLR(:,2)),Zoom));

        B=uint8(Bilineal(double(B),double(ImagenLR(:,3)),Zoom));

    end

    R=uint8(Bilineal(double(R),double(ImagenLR(:,1)),Zoom));

case 3

    if isrgb(Imagen)==1

        G=uint8(Bicubica(double(G),double(ImagenLR(:,2)),Zoom));

        B=uint8(Bicubica(double(B),double(ImagenLR(:,3)),Zoom));

    end

```

```

        R=uint8(Bicubica(double(R),double(ImagenLR(:,1)),Zoom));
    end

    Interpolacion(:,1)=R;

    if isrgb(Imagen)==1

        Interpolacion(:,2)=G;

        Interpolacion(:,3)=B;

    end

    valor_psnr = Calcula_psnr(Imagen,Interpolacion)

```

% Programa para evaluar el método de interpolación Bilinial

```

%----RESET-----

clear all;

clc

%----IMAGEN ORIGINAL-----

[Nombrelmagen Rutalmagen]=uigetfile({
    '*.jpeg;*.jpg;*.tif', 'Imagen JPEG (*.jpg, *.jpeg, *.tif)';
    '*.bmp', 'Imagen BMP (*.bmp)'},'MultiSelect','off');

if ~isequal(Rutalmagen,0) && ~isequal(Nombrelmagen,0)

    PathCompletoImagen=strcat(Rutalmagen,Nombrelmagen);

    Imagen=imread(PathCompletoImagen);

end

%----INTERPOLACION AL DOBLE DE RESOLUCIÓN---

Zoom = 2;

%----OBTENCION DE IMAGEN DE LR (DOWNSAMPLING)

ImagenLR=DimensionamientoLR(Imagen);

%----OBTENCION DE LA REJILLA DE HR PARA INTERPOLACION

```

```

if isrgb(Imagen)==1

    R=DimensionamientoHR(ImagenLR(:,1),Zoom);

    G=DimensionamientoHR(ImagenLR(:,2),Zoom);

    B=DimensionamientoHR(ImagenLR(:,3),Zoom);

else

    R=DimensionamientoHR(ImagenLR,Zoom);

end

%G1=G;

%B1=B;

%R1=R;

%G2=G;

%B2=B;

%R2=R;

%-----KERNEL DE INTERPOLACIÓN-----

Kx = [0.5 1 0.5]

Tam=size(ImagenLR);

m=Tam(1);

n=Tam(2);

if isrgb(Imagen)==1

    g_x=zeros(m*Zoom,n*Zoom);

    for i=1:m*Zoom

        convolucion_fila=conv(G(i,:),Kx,'same');

        g_x(i,:)=convolucion_fila;

    end

    for i=1:n*Zoom

        convolucion_col=conv(g_x(:,i),Kx,'same');

```

```

g_x(:,i)=convolucion_col';
end
G=g_x;
g_x=zeros(m*Zoom,n*Zoom);
for i=1:m*Zoom
    convolucion_fila=conv(B(i,:),Kx,'same');
    g_x(i,:)=convolucion_fila;
end
for i=1:n*Zoom
    convolucion_col=conv(g_x(:,i)',Kx,'same');
    g_x(:,i)=convolucion_col';
end
B=g_x;
G(:,512)=G(:,512)*2;
G(512,:)=G(512,)*2;
B(:,512)=B(:,512)*2;
B(512,:)=B(512,)*2;
end
g_x=zeros(m*Zoom,n*Zoom);
for i=1:m*Zoom
    convolucion_fila=conv(R(i,:),Kx,'same');
    g_x(i,:)=convolucion_fila;
end
for i=1:n*Zoom
    convolucion_col = conv(g_x(:,i)',Kx,'same');
    g_x(:,i)=convolucion_col';

```

```

end

R=g_x;

R(:,512)=R(:,512)*2;

R(512,:)=R(512,:)*2;

Interpol(:,,1)=uint8(R);

[ssimR ssim_map] = ssim_index(uint8(Imagen(:,,1)),uint8(R));

ssim_1 = ssimR

if isrgb(Imagen)==1

    Interpol(:,,2)=uint8(G);

    Interpol(:,,3)=uint8(B);

    [ssimG ssim_map] = ssim_index(uint8(Imagen(:,,2)),uint8(G));

    [ssimB ssim_map] = ssim_index(uint8(Imagen(:,,3)),uint8(B));

    ssim_1=(ssimR+ssimG+ssimB)/3

end

valor_psnr1 = Calcula_psnr(Imagen,Interpol)

%[mssim1 ssim_map1] = ssim_index(uint8(Imagen),uint8(Interpol))

% SEGUNDO METODO

if isrgb(Imagen)==1

    G=DimensionamientoHR(ImagenLR(:,,2),Zoom);

    B=DimensionamientoHR(ImagenLR(:,,3),Zoom);

end

R=DimensionamientoHR(ImagenLR(:,,1),Zoom);

s=0.5;

if isrgb(Imagen)==1

    for i=1:m*Zoom

        for j=2:2:n*Zoom-2

```

```

    G(i,j) = G(i,j-1)*(1-s)+G(i,j+1)*s;

    B(i,j) = B(i,j-1)*(1-s)+B(i,j+1)*s;

    end

end

for i=2:2:m*Zoom-2

    for j=1:n*Zoom

        G(i,j) = G(i-1,j)*(1-s)+G(i+1,j)*s;

        B(i,j) = B(i-1,j)*(1-s)+B(i+1,j)*s;

    end

end

G(m,:)=G(m-1,:);

B(m,:)=B(m-1,:);

G(:,n)=G(:,n-1);

B(:,n)=B(:,n-1);

end

for i=1:m*Zoom

    for j=2:2:n*Zoom-2

        R(i,j) = R(i,j-1)*(1-s)+R(i,j+1)*s;

    end

end

for i=2:2:m*Zoom-2

    for j=1:n*Zoom

        R(i,j) = R(i-1,j)*(1-s)+R(i+1,j)*s;

    end

end

R(m,:)=R(m-1,:);

```

```

R(:,n)=R(:,n-1);

Interpol2(:,,1)=uint8(R);

if isrgb(Imagen)==1

    Interpol2(:,,2)=uint8(G);

    Interpol2(:,,3)=uint8(B);

end

Interpol2(m*Zoom,:)=Interpol2(m*Zoom-1,:)/2;

Interpol2(:,n*Zoom)=Interpol2(:,n*Zoom-1)/2;

valor_psnr2 = Calcula_psnr(Imagen,Interpol2)

%G1=G;

%B1=B;

%R1=R;

%G2=G;

%B2=B;

%R2=R;

%METODO DIFUSO

if isrgb(Imagen)==1

    for i=1:m-1

        for j=1:n-1

            G2(2*i-1,2*j-1) = ImagenLR(i,j,2);

            G2(2*i,2*j-1) = (ImagenLR(i,j,2)+ImagenLR(i+1,j,2))/2;

            G2(2*i-1,2*j) = (ImagenLR(i,j,2)+ImagenLR(i,j+1,2))/2;

            G2(2*i,2*j) = (ImagenLR(i,j,2)+ImagenLR(i+1,j,2)+ImagenLR(i,j+1,2)+ImagenLR(i+1,j+1,2))/4;

            B2(2*i-1,2*j-1) = ImagenLR(i,j,3);

            B2(2*i,2*j-1) = (ImagenLR(i,j,3)+ImagenLR(i+1,j,3))/2;

            B2(2*i-1,2*j) = (ImagenLR(i,j,3)+ImagenLR(i,j+1,3))/2;

```

```

        B2(2*i,2*j) =(ImagenLR(i,j,3)+ImagenLR(i+1,j,3)+ImagenLR(i,j+1,3)+ImagenLR(i+1,j+1,3))/4;
    end
    end
for i=1:m-1
    for j=n:n
        G2(2*i-1,2*j-1) = ImagenLR(i,j,2);
        G2(2*i,2*j-1) = (ImagenLR(i,j,2)+ImagenLR(i+1,j,2))/2;
        G2(2*i-1,2*j) = (ImagenLR(i,j,2)+ImagenLR(i,j-1,2))/2;
        G2(2*i,2*j) =(ImagenLR(i,j,2)+ImagenLR(i+1,j,2)+ImagenLR(i,j-1,2)+ImagenLR(i+1,j-1,2))/4;
        B2(2*i-1,2*j-1) = ImagenLR(i,j,3);
        B2(2*i,2*j-1) = (ImagenLR(i,j,3)+ImagenLR(i+1,j,3))/2;
        B2(2*i-1,2*j) = (ImagenLR(i,j,3)+ImagenLR(i,j-1,3))/2;
        B2(2*i,2*j) =(ImagenLR(i,j,3)+ImagenLR(i+1,j,3)+ImagenLR(i,j-1,3)+ImagenLR(i+1,j-1,3))/4;
    end
end
for i=n:n
    for j=1:m-1
        G2(2*i-1,2*j-1) = ImagenLR(i,j,2);
        G2(2*i,2*j-1) = (ImagenLR(i,j,2)+ImagenLR(i-1,j,2))/2;
        G2(2*i-1,2*j) = (ImagenLR(i,j,2)+ImagenLR(i,j+1,2))/2;
        G2(2*i,2*j) =(ImagenLR(i,j,2)+ImagenLR(i-1,j,2)+ImagenLR(i,j+1,2)+ImagenLR(i-1,j+1,2))/4;
        B2(2*i-1,2*j-1) = ImagenLR(i,j,3);
        B2(2*i,2*j-1) = (ImagenLR(i,j,3)+ImagenLR(i-1,j,3))/2;
        B2(2*i-1,2*j) = (ImagenLR(i,j,3)+ImagenLR(i,j+1,3))/2;
        B2(2*i,2*j) =(ImagenLR(i,j,3)+ImagenLR(i-1,j,3)+ImagenLR(i,j+1,3)+ImagenLR(i-1,j+1,3))/4;
    end
end

```

```

end

G2(2*m-1,2*n-1) = ImagenLR(m,n,2);

G2(2*m,2*n-1) = (ImagenLR(m,n,2)+ImagenLR(m-1,n,2))/2;

G2(2*m-1,2*n) = (ImagenLR(m,n,2)+ImagenLR(m,n-1,2))/2;

G2(2*m,2*n) =(ImagenLR(m,n,2)+ImagenLR(m-1,n,2)+ImagenLR(m,n-1,2)+ImagenLR(m-1,n-1,2))/4;

B2(2*m-1,2*n-1) = ImagenLR(m,n,3);

B2(2*m,2*n-1) = (ImagenLR(m,n,3)+ImagenLR(m-1,n,3))/2;

B2(2*m-1,2*n) = (ImagenLR(m,n,3)+ImagenLR(m,n-1,3))/2;

B2(2*m,2*n) =(ImagenLR(m,n,3)+ImagenLR(m-1,n,3)+ImagenLR(m,n-1,3)+ImagenLR(m-1,n-1,3))/4;

end

for i=1:m-1

for j=1:n-1

R2(2*i-1,2*j-1) = ImagenLR(i,j,1);

R2(2*i,2*j-1) = (ImagenLR(i,j,1)+ImagenLR(i+1,j,1))/2;

R2(2*i-1,2*j) = (ImagenLR(i,j,1)+ImagenLR(i,j+1,1))/2;

R2(2*i,2*j) =(ImagenLR(i,j,1)+ImagenLR(i+1,j,1)+ImagenLR(i,j+1,1)+ImagenLR(i+1,j+1,1))/4;

end

end

for i=1:m-1

for j=n:n

R2(2*i-1,2*j-1) = ImagenLR(i,j,1);

R2(2*i,2*j-1) = (ImagenLR(i,j,1)+ImagenLR(i+1,j,1))/2;

R2(2*i-1,2*j) = (ImagenLR(i,j,1)+ImagenLR(i,j-1,1))/2;

R2(2*i,2*j) =(ImagenLR(i,j,1)+ImagenLR(i+1,j,1)+ImagenLR(i,j-1,1)+ImagenLR(i+1,j-1,1))/4;

end

```

```

end
for i=n:n
for j=1:m-1
    R2(2*i-1,2*j-1) = ImagenLR(i,j,1);
    R2(2*i,2*j-1) = (ImagenLR(i,j,1)+ImagenLR(i-1,j,1))/2;
    R2(2*i-1,2*j) = (ImagenLR(i,j,1)+ImagenLR(i,j+1,1))/2;
    R2(2*i,2*j) =(ImagenLR(i,j,1)+ImagenLR(i-1,j,1)+ImagenLR(i,j+1,1)+ImagenLR(i-1,j+1,1))/4;
end
end
R2(2*m-1,2*n-1) = ImagenLR(m,n,1);
R2(2*m,2*n-1) = (ImagenLR(m,n,1)+ImagenLR(m-1,n,1))/2;
R2(2*m-1,2*n) = (ImagenLR(m,n,1)+ImagenLR(m,n-1,1))/2;
R2(2*m,2*n) =(ImagenLR(m,n,1)+ImagenLR(m-1,n,1)+ImagenLR(m,n-1,1)+ImagenLR(m-1,n-1,1))/4;
%if isrgb(Imagen)==1
% for i=2:2:m*Zoom-1
% for j=2:2:n*Zoom-1
%     G2(i,j) = (G(i-1,j-1)+G(i-1,j+1)+G(i+1,j-1)+G(i+1,j+1))/4;
%     B2(i,j) = (B(i-1,j-1)+B(i-1,j+1)+B(i+1,j-1)+B(i+1,j+1))/4;
% end
% end
%end
% for i=2:2:m*Zoom-1
% for j=2:2:n*Zoom-1
%     R2(i,j) = (R(i-1,j-1)+R(i-1,j+1)+R(i+1,j-1)+R(i+1,j+1))/4;
% end
%end

```

```

Interpol3(:,:,1)=uint8(R2);

[ssimR3 ssim_map] = ssim_index(uint8(Imagen(:,:,1)),uint8(R));

ssim_3 = ssimR3

if isrgb(Imagen)==1

    Interpol3(:,:,2)=uint8(G2);

    Interpol3(:,:,3)=uint8(B2);

    Interpol3(:,:,1)=uint8(R2);

    [ssimR3 ssim_map] = ssim_index(uint8(Imagen(:,:,1)),uint8(R2));

    [ssimG3 ssim_map] = ssim_index(uint8(Imagen(:,:,2)),uint8(G2));

    [ssimB3 ssim_map] = ssim_index(uint8(Imagen(:,:,3)),uint8(B2));

    ssim_3=(ssimR3+ssimG3+ssimB3)/3

end

valor_psnr3 = Calcula_psnr(Imagen,Interpol3)

    if isrgb(Imagen)==1

        EstaMal=1

        G=uint8(Bilinear(double(G),double(ImagenLR(:,:,2)),Zoom));

        B=uint8(Bilinear(double(B),double(ImagenLR(:,:,3)),Zoom));

    end

    R=uint8(Bilinear(double(R),double(ImagenLR(:,:,1)),Zoom));

Interpolacion(:,:,1)=R;

if isrgb(Imagen)==1

    Interpolacion(:,:,2)=G;

    Interpolacion(:,:,3)=B;

end

valor_psnr4 = Calcula_psnr(Imagen,Interpolacion)

```

```

imtool(Imagen);

imtool(Interpol);

imtool(Interpol2);

imtool(Interpol3);

imtool(Interpolacion);

```

% Programa para evaluar el método de interpolación Bicubica

```

%----RESET-----

clear all;

clc

%----IMAGEN ORIGINAL-----

[NombrelImagen RutalImagen]=uigetfile({
    '*.jpeg;*.jpg;*.tif', 'Imagen JPEG (*.jpg, *.jpeg, *.tif);
    '*.bmp', 'Imagen BMP (*.bmp)'},'MultiSelect','off');

if ~isequal(RutalImagen,0) && ~isequal(NombrelImagen,0)

    PathCompletoImagen=strcat(RutalImagen,NombrelImagen);

    Imagen=imread(PathCompletoImagen);

end

%----INTERPOLACION AL DOBLE DE RESOLUCIÓN---

Zoom = 2;

%----OBTENCION DE IMAGEN DE LR (DOWNSAMPLING)

ImagenLR=DimensionamientoLR(Imagen);

%----OBTENCION DE LA REJILLA DE HR PARA INTERPOLACION

if isrgb(Imagen)==1

    G=DimensionamientoHR(ImagenLR(:, :, 2),Zoom);

    B=DimensionamientoHR(ImagenLR(:, :, 3),Zoom);

```

```

end

R=DimensionamientoHR(ImagenLR(:,1),Zoom);

%-----KERNEL DE INTERPOLACIÓN-----

Kx = [0 0.0208 0.1667 0.4792 0.6667 0.4792 0.1667 0.0208 0]

Tam=size(ImagenLR);

m=Tam(1);

n=Tam(2);

if isrgb(Imagen)==1

    g_x=zeros(m*Zoom,n*Zoom);

    for i=1:m*Zoom

        convolucion_fila=conv(G(i,:),Kx,'same');

        g_x(i,:)=convolucion_fila;

    end

    for i=1:m*Zoom

        convolucion_col=conv(g_x(:,i)',Kx,'same');

        g_x(:,i)=convolucion_col';

    end

    G=g_x;

    g_x=zeros(m*Zoom,n*Zoom);

    for i=1:m*Zoom

        convolucion_fila=conv(B(i,:),Kx,'same');

        g_x(i,:)=convolucion_fila;

    end

    for i=1:m*Zoom

        convolucion_col=conv(g_x(:,i)',Kx,'same');

        g_x(:,i)=convolucion_col';

    end

```

```

end

B=g_x;

% G(511:512,:)=G(509:510,:);
% G(:,511:512)=G(:,509:510);
% B(511:512,:)=B(509:510,:);
% B(:,511:512)=B(:,509:510);

end

g_x=zeros(m*Zoom,n*Zoom);

for i=1:m*Zoom

    convolucion_fila=conv(R(i,:),Kx,'same');

    g_x(i,:)=convolucion_fila;

end

for i=1:m*Zoom

    convolucion_col=conv(g_x(:,i)',Kx,'same');

    g_x(:,i)=convolucion_col';

end

R=g_x;

% R(511:512,:)=R(509:510,:);
% R(:,511:512)=R(:,509:510);

Interpol(:,,1)=uint8(R);

[ssimR2 ssim_map] = ssim_index(uint8(Imagen(:,,1)),uint8(R));

ssim_3 = ssimR2

if isrgb(Imagen)==1

    Interpol(:,,2)=uint8(G);

    Interpol(:,,3)=uint8(B);

```

```

[ssimG2 ssim_map] = ssim_index(uint8(Imagen(:,:,2)),uint8(G));

[ssimB2 ssim_map] = ssim_index(uint8(Imagen(:,:,3)),uint8(B));

ssim_3=(ssimR2+ssimG2+ssimB2)/3

end

valor_psnr1 = Calcula_psnr(Imagen,Interpol)

valor_psnr11 = Calcula_psnr(Imagen,Interpol)

% SEGUNDO METODO

if isrgb(Imagen)==1

    G=DimensionamientoHR(ImagenLR(:,:,2),Zoom);

    B=DimensionamientoHR(ImagenLR(:,:,3),Zoom);

end

R=DimensionamientoHR(ImagenLR(:,:,1),Zoom);

s=0.5;

if isrgb(Imagen)==1

    for i=1:m*Zoom

        j=2;

        G(i,j) = G(i,j-1)*(((2+s)^3-4*(1+s)^3+6*s^3)/6) +G(i,j+1)*(((1+s)^3-4*s^3)/6)+G(i,j+3)*(s^3)/6;

        B(i,j) = B(i,j-1)*(((2+s)^3-4*(1+s)^3+6*s^3)/6) +B(i,j+1)*(((1+s)^3-4*s^3)/6)+B(i,j+3)*(s^3)/6;

        for j=4:2:n*Zoom-4

            G(i,j) = G(i,j-3)*(((3+s)^3)-4*((2+s)^3)+6*((1+s)^3)-4*(s^3))/6)+G(i,j-1)*(((2+s)^3-4*(1+s)^3+6*s^3)/6) +G(i,j+1)*(((1+s)^3-4*s^3)/6)+G(i,j+3)*(s^3)/6;

            B(i,j) = B(i,j-3)*(((3+s)^3)-4*((2+s)^3)+6*((1+s)^3)-4*(s^3))/6)+B(i,j-1)*(((2+s)^3-4*(1+s)^3+6*s^3)/6) +B(i,j+1)*(((1+s)^3-4*s^3)/6)+B(i,j+3)*(s^3)/6;

        end

        j=m*Zoom-2;

        G(i,j) = G(i,j-3)*(((3+s)^3)-4*((2+s)^3)+6*((1+s)^3)-4*(s^3))/6)+G(i,j-1)*(((2+s)^3-4*(1+s)^3+6*s^3)/6) +G(i,j+1)*(((1+s)^3-4*s^3)/6);

```

```

    B(i,j) = B(i,j-3)*(((3+s)^3)-4*((2+s)^3)+6*((1+s)^3)-4*(s^3))/6+B(i,j-1)*(((2+s)^3-
4*(1+s)^3+6*s^3)/6) +B(i,j+1)*(((1+s)^3-4*s^3)/6);

    end

    for i=1:m*Zoom

        j=2;

        G(j,i) = G(j-1,i)*(((2+s)^3-4*(1+s)^3+6*s^3)/6) +G(j+1,i)*(((1+s)^3-4*s^3)/6)+G(j+3,i)*(s^3)/6;

        B(j,i) = B(j-1,i)*(((2+s)^3-4*(1+s)^3+6*s^3)/6) +B(j+1,i)*(((1+s)^3-4*s^3)/6)+B(j+3,i)*(s^3)/6;

    for j= 4:2:n*Zoom-4

        G(j,i) = G(j-3,i)*(((3+s)^3)-4*((2+s)^3)+6*((1+s)^3)-4*(s^3))/6)+G(j-1,i)*(((2+s)^3-
4*(1+s)^3+6*s^3)/6) +G(j+1,i)*(((1+s)^3-4*s^3)/6)+G(j+3,i)*(s^3)/6;

        B(j,i) = B(j-3,i)*(((3+s)^3)-4*((2+s)^3)+6*((1+s)^3)-4*(s^3))/6)+B(j-1,i)*(((2+s)^3-
4*(1+s)^3+6*s^3)/6) +B(j+1,i)*(((1+s)^3-4*s^3)/6)+B(j+3,i)*(s^3)/6;

    end

    j=n*Zoom-2;

    G(j,i) = G(j-3,i)*(((3+s)^3)-4*((2+s)^3)+6*((1+s)^3)-4*(s^3))/6)+G(j-1,i)*(((2+s)^3-
4*(1+s)^3+6*s^3)/6) +G(j+1,i)*(((1+s)^3-4*s^3)/6);

    B(j,i) = B(j-3,i)*(((3+s)^3)-4*((2+s)^3)+6*((1+s)^3)-4*(s^3))/6)+B(j-1,i)*(((2+s)^3-
4*(1+s)^3+6*s^3)/6) +B(j+1,i)*(((1+s)^3-4*s^3)/6);

    end

    G(512,:)=G(511,:);

    B(512,:)=B(511,:);

    G(:,512)=G(:,511);

    B(:,512)=B(:,511);

    end

    for i=1:m*Zoom

        j=2;

        R(i,j) = R(i,j-1)*(((2+s)^3-4*(1+s)^3+6*s^3)/6) +R(i,j+1)*(((1+s)^3-4*s^3)/6)+R(i,j+3)*(s^3)/6;

    for j=4:2:n*Zoom-4

```

```

    R(i,j) = R(i,j-3)*(((3+s)^3)-4*((2+s)^3)+6*((1+s)^3)-4*(s^3))/6+R(i,j-1)*(((2+s)^3-
4*(1+s)^3+6*s^3)/6) +R(i,j+1)*(((1+s)^3-4*s^3)/6)+R(i,j+3)*(s^3)/6;

    end

    j=m*Zoom-2;

R(i,j) = R(i,j-3)*(((3+s)^3)-4*(2+s)^3+6*(2+s)^3-4*s^3)/6+R(i,j-1)*(((2+s)^3-4*(1+s)^3+6*s^3)/6)
+R(i,j+1)*(((1+s)^3-4*s^3)/6);

end

for i=1:n*Zoom

    j=2;

    R(j,i) = R(j-1,i)*(((2+s)^3-4*(1+s)^3+6*s^3)/6) +R(j+1,i)*(((1+s)^3-4*s^3)/6)+R(j+3,i)*(s^3)/6;

    for j=4:2:m*Zoom-4

        R(j,i) = R(j-3,i)*(((3+s)^3)-4*((2+s)^3)+6*((1+s)^3)-4*(s^3))/6+R(j-1,i)*(((2+s)^3-
4*(1+s)^3+6*s^3)/6) +R(j+1,i)*(((1+s)^3-4*s^3)/6)+R(j+3,i)*(s^3)/6;

        end

        j=n*Zoom-2;

        R(j,i) = R(j-3,i)*(((3+s)^3)-4*((2+s)^3)+6*((1+s)^3)-4*(s^3))/6+R(j-1,i)*(((2+s)^3-
4*(1+s)^3+6*s^3)/6)+R(j+1,i)*(((1+s)^3-4*s^3)/6);

    end

    R(512,:)=R(511,:);

    R(:,512)=R(:,511);

Interpol2(:,,1)=uint8(R);

[ssimR2 ssim_map] = ssim_index(uint8(Imagen(:,,1)),uint8(R));

ssim_3 = ssimR2;

if isrgb(Imagen)==1

    Interpol2(:,,2)=uint8(G);

    Interpol2(:,,3)=uint8(B);

    [ssimG2 ssim_map] = ssim_index(uint8(Imagen(:,,2)),uint8(G));

    [ssimB2 ssim_map] = ssim_index(uint8(Imagen(:,,3)),uint8(B));

```

```

    ssim_3=(ssimR2+ssimG2+ssimB2)/3
end
valor_psnr2 = Calcula_psnr(Imagen,Interpol2)
str = {'Vecino más cercano';'Bilineal';'Bicúbica'};
[seleccion,v] = listdlg('PromptString','Select a file:',...
    'SelectionMode','single',...
    'ListString',str)
switch seleccion
case 1
    if isrgb(ImagenLR)==1
        G=uint8(MasCercano(double(G),double(ImagenLR(:,:,2)),Zoom));
        B=uint8(MasCercano(double(B),double(ImagenLR(:,:,3)),Zoom));
    end
    R=uint8(MasCercano(double(R),double(ImagenLR(:,:,1)),Zoom));
case 2
    if isrgb(Imagen)==1
        G=uint8(Bilineal(double(G),double(ImagenLR(:,:,2)),Zoom));
        B=uint8(Bilineal(double(B),double(ImagenLR(:,:,3)),Zoom));
    end
    R=uint8(Bilineal(double(R),double(ImagenLR(:,:,1)),Zoom));
case 3
    if isrgb(Imagen)==1
        G=uint8(Bicubica(double(G),double(ImagenLR(:,:,2)),Zoom));
        B=uint8(Bicubica(double(B),double(ImagenLR(:,:,3)),Zoom));
    end
    R=uint8(Bicubica(double(R),double(ImagenLR(:,:,1)),Zoom));

```

```

end

Interpolacion(:,:,1)=R;

if isrgb(Imagen)==1

    Interpolacion(:,:,2)=G;

    Interpolacion(:,:,3)=B;

end

valor_psnr3 = Calcula_psnr(Imagen,Interpolacion)

imtool(uint8(ImagenLR))

imtool(Imagen);

imtool(Interpol);

imtool(Interpol2);

imtool(Interpolacion);

```

% Programa para evaluar el método de interpolación O-MOMS

```

%----RESET-----

clear all;

clc

%----IMAGEN ORIGINAL-----

[Nombrelmagen Rutalmagen]=uigetfile({

    '*.jpeg;*.jpg;*.tif', 'Imagen JPEG (*.jpg, *.jpeg, *.tif)';

    '*.bmp', 'Imagen BMP (*.bmp)'},'MultiSelect','off');

if ~isequal(Rutalmagen,0) && ~isequal(Nombrelmagen,0)

    PathCompletoImagen=strcat(Rutalmagen,Nombrelmagen);

    Imagen=imread(PathCompletoImagen);

end

%----INTERPOLACION AL DOBLE DE RESOLUCIÓN---

```

```

Zoom = 2;

%----OBTENCION DE IMAGEN DE LR (DOWNSAMPLING)

ImagenLR=DimensionamientoLR(Imagen);

%----OBTENCION DE LA REJILLA DE HR PARA INTERPOLACION

if isrgb(Imagen)==1

    G=DimensionamientoHR(ImagenLR(:,:,2),Zoom);

    B=DimensionamientoHR(ImagenLR(:,:,3),Zoom);

end

R=DimensionamientoHR(ImagenLR(:,:,1),Zoom);

%-----KERNEL DE INTERPOLACIÓN-----

Kx = [0.0327 0.1905 0.4673 0.6190 0.4673 0.1905 0.0327]

Tam=size(ImagenLR);

m=Tam(1);

n=Tam(2);

if isrgb(Imagen)==1

    g_x=zeros(m*Zoom,n*Zoom);

    for i=1:m*Zoom

        convolucion_fila=conv(G(i,:),Kx,'same');

        g_x(i,:)=convolucion_fila;

    end

    for i=1:m*Zoom

        convolucion_col=conv(g_x(:,i)',Kx,'same');

        g_x(:,i)=convolucion_col';

    end

    G=g_x;

    g_x=zeros(m*Zoom,n*Zoom);

```

```

for i=1:m*Zoom

    convolucion_fila=conv(B(i,:),Kx,'same');

    g_x(i,:)=convolucion_fila;

end

for i=1:m*Zoom

convolucion_col=conv(g_x(:,i)',Kx,'same');

g_x(:,i)=convolucion_col';

end

B=g_x;

G(m*2,:)=G(m*2-1,:);

B(m*2,:)=B(m*2-1,:);

G(:,n*2)=G(:,n*2-1);

B(:,n*2)=B(:,n*2-1);

end

g_x=zeros(m*Zoom,n*Zoom);

for i=1:m*Zoom

    convolucion_fila=conv(R(i,:),Kx,'same');

    g_x(i,:)=convolucion_fila;

end

for i=1:m*Zoom

    convolucion_col=conv(g_x(:,i)',Kx,'same');

    g_x(:,i)=convolucion_col';

end

R=g_x;

R(m,:)=R(m*2-1,:);

R(:,n)=R(:,n*2-1);

```

```

Interpol(:,:,1)=uint8(R);

if isrgb(Imagen)==1
    Interpol(:,:,2)=uint8(G);
    Interpol(:,:,3)=uint8(B);
end

[ssimR3 ssim_map] = ssim_index(uint8(Imagen(:,:,1)),uint8(R));
ssim_3 = ssimR3

if isrgb(Imagen)==1
    Interpol(:,:,2)=uint8(G);
    Interpol(:,:,3)=uint8(B);

    [ssimG3 ssim_map] = ssim_index(uint8(Imagen(:,:,2)),uint8(G));
    [ssimB3 ssim_map] = ssim_index(uint8(Imagen(:,:,3)),uint8(B));

    ssim_3=(ssimR3+ssimG3+ssimB3)/3
end

valor_psnr1 = Calcula_psnr(Imagen,Interpol)

imtool(uint8(ImagenLR))

imtool(Imagen);

imtool(uint8(Interpol));

% SEGUNDO METODO

if isrgb(Imagen)==1
    G=DimensionamientoHR(ImagenLR(:,:,2),Zoom);
    B=DimensionamientoHR(ImagenLR(:,:,3),Zoom);
end

R=DimensionamientoHR(ImagenLR(:,:,1),Zoom);

s=0.5;

if isrgb(Imagen)==1

```

```

for i=1:m*Zoom

    j=2;

    G(i,j) = G(i,j-1)*((1/2)*(s^3)-(s^2)+(s/14)+(13/21)) +G(i,j+1)*((1/2)*((1-s)^3)-((1-
s)^2)+(s/14)+(13/21))+G(i,j+3)*(-(1/6)*((2-s)^3)+((2-s)^2)-(85/42)*(2-s)+(29/21));

    B(i,j) = B(i,j-1)*((1/2)*(s^3)-(s^2)+(s/14)+(13/21)) +B(i,j+1)*((1/2)*((1-s)^3)-((1-
s)^2)+(s/14)+(13/21))+B(i,j+3)*(-(1/6)*((2-s)^3)+((2-s)^2)-(85/42)*(2-s)+(29/21));

    for j=4:2:n*Zoom-4

        G(i,j) = G(i,j-3)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+G(i,j-1)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +G(i,j+1)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21))+G(i,j+3)*(-(1/6)*((2-
s)^3)+((2-s)^2)-(85/42)*(2-s)+(29/21));

        B(i,j) = B(i,j-3)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+B(i,j-1)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +B(i,j+1)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21))+B(i,j+3)*(-(1/6)*((2-
s)^3)+((2-s)^2)-(85/42)*(2-s)+(29/21));

    end

    j=m*Zoom-2;

    G(i,j) = G(i,j-3)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+G(i,j-1)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +G(i,j+1)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21));

    B(i,j) = B(i,j-3)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+B(i,j-1)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +B(i,j+1)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21));

    end

for i=1:n*Zoom

    j=2;

    G(j,i) = G(j-1,i)*((1/2)*(s^3)-(s^2)+(s/14)+(13/21)) +G(j+1,i)*((1/2)*((1-s)^3)-((1-
s)^2)+(s/14)+(13/21))+G(j+3,i)*(-(1/6)*(2-s)^3+(2-s)^2-(85/42)*(2-s)+(29/21));

    B(j,i) = B(j-1,i)*((1/2)*(s^3)-(s^2)+(s/14)+(13/21)) +B(j+1,i)*((1/2)*((1-s)^3)-((1-
s)^2)+(s/14)+(13/21))+B(j+3,i)*(-(1/6)*(2-s)^3+(2-s)^2-(85/42)*(2-s)+(29/21));

    for j=4:2:m*Zoom-4

        G(j,i) = G(j-3,i)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+G(j-1,i)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +G(j+1,i)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21))+G(j+3,i)*(-(1/6)*((2-
s)^3)+((2-s)^2)-(85/42)*(2-s)+(29/21));

```

```

    B(j,i) = B(j-3,i)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+B(j-1,i)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +B(j+1,i)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21))+B(j+3,i)*(-(1/6)*((2-
s)^3)+(2-s)^2-(85/42)*(2-s)+(29/21));

```

```

end

```

```

j=m*Zoom-2;

```

```

    G(j,i) = G(j-3,i)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+G(j-1,i)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +G(j+1,i)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21));

```

```

    B(j,i) = B(j-3,i)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+B(j-1,i)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +B(j+1,i)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21));

```

```

end

```

```

G(m*2,:)=G(m*2-1,:);

```

```

B(m*2,:)=B(m*2-1,:);

```

```

G(:,n*2)=G(:,n*2-1);

```

```

B(:,n*2)=B(:,n*2-1);

```

```

end

```

```

for i=1:n*Zoom

```

```

    j=2;

```

```

    R(i,j) = R(i,j-1)*((1/2)*(s^3)-(s^2)+(s/14)+(13/21)) +R(i,j+1)*((1/2)*(1-s)^3-(1-
s)^2+(s/14)+(13/21))+R(i,j+3)*(-(1/6)*((2-s)^3)+((2-s)^2)-(85/42)*(2-s)+(29/21));

```

```

    for j=4:2:m*Zoom-4

```

```

        R(i,j) = R(i,j-3)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+R(i,j-1)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +R(i,j+1)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21))+R(i,j+3)*(-(1/6)*(2-s)^3+(2-
s)^2-(85/42)*(2-s)+(29/21));

```

```

    end

```

```

    j=m*Zoom-2;

```

```

    R(i,j) = R(i,j-3)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+R(i,j-1)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +R(i,j+1)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21));

```

```

end

```

```

for i=1:n*Zoom

```

```

    j=2;

```

```

R(j,i) = R(j-1,i)*((1/2)*(s^3)-(s^2)+(s/14)+(13/21)) +R(j+1,i)*((1/2)*(1-s)^3-(1-
s)^2+(s/14)+(13/21))+R(j+3,i)*(-(1/6)*(2-s)^3+(2-s)^2-(85/42)*(2-s)+(29/21));

```

```

for j=4:2:m*Zoom-4

```

```

    R(j,i) = R(j-3,i)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+R(j-1,i)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +R(j+1,i)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21))+R(j+3,i)*(-(1/6)*(2-s)^3+(2-
s)^2-(85/42)*(2-s)+(29/21));

```

```

end

```

```

j=m*Zoom-2;

```

```

R(j,i) = R(j-3,i)*(-(1/6)*(1+s)^3+((1+s)^2)-(85/42)*(1+s)+(29/21))+R(j-1,i)*((1/2)*(s^3)-
(s^2)+(s/14)+(13/21)) +R(j+1,i)*((1/2)*((1-s)^3)-((1-s)^2)+(s/14)+(13/21));

```

```

end

```

```

R(m,:)=R(m*2-1,:);

```

```

R(:,n)=R(:,n*2-1);

```

```

Interpol3(:,:,1)=uint8(R);

```

```

if isrgb(Imagen)==1

```

```

    Interpol3(:,:,2)=uint8(G);

```

```

    Interpol3(:,:,3)=uint8(B);

```

```

end

```

```

[ssimR2 ssim_map] = ssim_index(uint8(Imagen(:,:,1)),uint8(R));

```

```

ssim_3 = ssimR2

```

```

if isrgb(Imagen)==1

```

```

    Interpol4(:,:,2)=uint8(G);

```

```

    Interpol4(:,:,3)=uint8(B);

```

```

[ssimG2 ssim_map] = ssim_index(uint8(Imagen(:,:,2)),uint8(G));

```

```

[ssimB2 ssim_map] = ssim_index(uint8(Imagen(:,:,3)),uint8(B));

```

```

ssim_3=(ssimR2+ssimG2+ssimB2)/3

```

```

end

```

```

valor_psnr2 = Calcula_psnr(Imagen,Interpol3)

```

```
imtool(uint8(Interpol3));
```

% Programa para evaluar el método de interpolación DTC

```
clear all;
```

```
clc
```

```
%----IMAGEN ORIGINAL-----
```

```
[NombreImagen RutalImagen]=uigetfile({
```

```
    '*.jpeg;*.jpg;*.tif', 'Imagen JPEG (*.jpg, *.jpeg, *.tif)';
```

```
    '*.bmp', 'Imagen BMP (*.bmp)'},'MultiSelect','off');
```

```
if ~isequal(RutalImagen,0) && ~isequal(NombreImagen,0)
```

```
    PathCompletoImagen=strcat(RutalImagen,NombreImagen);
```

```
    Imagen=imread(PathCompletoImagen);
```

```
end
```

```
%----INTERPOLACION AL DOBLE DE RESOLUCIÓN---
```

```
Zoom = 2;
```

```
%----OBTENCION DE IMAGEN DE LR (DOWNSAMPLING)
```

```
%----OBTENCION DE LA REJILLA DE HR PARA INTERPOLACION
```

```
ImagenLR=DimensionamientoLR(Imagen);
```

```
%----OBTENCION DE LA REJILLA DE HR PARA INTERPOLACION
```

```
if isrgb(Imagen)==1
```

```
    G=DimensionamientoHR(ImagenLR(:, :, 2),Zoom);
```

```
    B=DimensionamientoHR(ImagenLR(:, :, 3),Zoom);
```

```
end
```

```
R=DimensionamientoHR(ImagenLR(:, :, 1),Zoom);
```

```
[m n]=size(ImagenLR(:, :, 1));
```

```
if isrgb(Imagen)==1
```

```

Matriz_Interpol=zeros(m*2,n*2,3);

Matriz_Interpol(1:m,1:n,1)=dct2(ImagenLR(:,1));

Matriz_Interpol(1:m,1:n,2)=dct2(ImagenLR(:,2));

Matriz_Interpol(1:m,1:n,3)=dct2(ImagenLR(:,3));

Interpolacion(:,1)=idct2(Matriz_Interpol(:,1))*2;

Interpolacion(:,2)=idct2(Matriz_Interpol(:,2))*2;

Interpolacion(:,3)=idct2(Matriz_Interpol(:,3))*2;

[ssimR ssim_map] = ssim_index(uint8(Imagen(:,1)),uint8(Interpolacion(:,1)));

[ssimG ssim_map] = ssim_index(uint8(Imagen(:,2)),uint8(Interpolacion(:,2)));

[ssimB ssim_map] = ssim_index(uint8(Imagen(:,3)),uint8(Interpolacion(:,3)));

(ssimR+ssimG+ssimB)/3

else

Matriz_Interpol=zeros(m*2,n*2);

Matriz_Interpol(1:m,1:n)=dct2(ImagenLR);

Interpolacion=idct2(Matriz_Interpol);

[ssimR ssim_map] = ssim_index(uint8(Imagen(:,1)),uint8(Interpolacion));

SSIM = ssimR

end

valor_psnr1 = Calcula_psnr(Imagen,Interpolacion)

imtool(Imagen)

imtool(uint8(Interpolacion))

```

REFERENCIAS

- [1] Abd el-Samie, F., Hadhoud, M., & El-Khamy, S. (2013). *Image super-resolution and applications*. Boca Raton, FL: CRC Press.
- [2] Aréndiga, F., Donat, R., & Mulet, P. Adaptive interpolation of images. *Signal Processing*, [en línea]; EEUU 2003, [consulta: octubre 2017]. Disponible en: <[http://dx.doi.org/10.1016/s0165-1684\(02\)00445-0](http://dx.doi.org/10.1016/s0165-1684(02)00445-0)>
- [3] Brooks, A., Xiaonan Zhao, & Pappas, T. Structural Similarity Quality Metrics in a Coding Context: Exploring the Space of Realistic Distortions. *IEEE Transactions On Image Processing*, [en línea]; EEUU 2008 [consulta: octubre 2017]. Disponible en: <<http://dx.doi.org/10.1109/tip.2008.926161>> pp 1261-1273.
- [4] El-Khamy, S., Hadhoud, M., Dessouky, M., Salam, B., & Abd El-Samie, F. A new approach for regularized image interpolation. *Journal Of The Brazilian Computer Society*, [en línea]; EEUU 2003, [consulta: Agosto 2017]. Disponible en: <<http://dx.doi.org/10.1007/bf03192383>> pp 65-79.
- [5] Farsiu, S., Elad, M., & Milanfar, P. Video-to-Video Dynamic Super-Resolution for Grayscale and Color Sequences. *EURASIP Journal On Advances In Signal Processing*, [en línea]; EEUU 2006, [consulta: Agosto 2017]. Disponible en: <<http://dx.doi.org/10.1155/asp/2006/61859>>
- [6] Han, D. (2013). Comparison of Commonly Used Image Interpolation Methods. *Proceedings Of The 2Nd International Conference On Computer Science And Electronics Engineering*, pp 1-4.
- [7] Haseyama, M., Izumi, D., & Takizawa, M. Super-Resolution Reconstruction for Spatio-Temporal Resolution Enhancement of Video Sequences. *IEICE Transactions On Information And Systems, E95.D(9)*, [en línea]; EEUU 2012, [consulta: noviembre 2016]. Disponible en: <<http://dx.doi.org/10.1587/transinf.e95.d.2355>> pp 2355-2358.
- [8] Horé, A., & Ziou, D. (2010). Image quality metrics: PSNR vs SSIM. *International Conference On Pattern Recognition*, pp 1-4.
- [9] Kadiravan, D, K. Effective Pixel Interpolation for Image Super Resolution. *IOSR Journal Of Electronics And Communication Engineering*, [en línea]; 2013, [consulta: marzo 2017]. Disponible en: <<http://dx.doi.org/10.9790/2834-0621520>> pp 15-20.

- [10] Katsaggelos, A., Molina, R., & Mateos, J. (2007). *Super resolution of images and video*. San Rafael, Calif.: Morgan & Claypool Publishers.
- [11] Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, [en línea]; 1981, [consulta: noviembre 2017]. Disponible en: <<http://dx.doi.org/10.1109/tassp.1981.1163711>> pp 1153-1160.
- [12] Kumar, R., & Rattan, M. Analysis of Various Quality Metrics for Medical Image Processing. *Internarional Journal Of Advanced Research In Computer Scince And Sorware Engineering*, [en línea]; 2012, [consulta: noviembre 2017]. Disponible en: <<http://dx.doi.org/ISSN:2277128X>> pp 1-8.
- [14] Ma, L., Li, S., & Ngan, K. Reduced-reference image quality assessment in reorganized DCT domain. *Signal Processing: Image Communication*, [en línea]; 2013, [consulta: marzo 2017]. Disponible en: <884-902. <http://dx.doi.org/10.1016/j.image.2012.08.001>>
- [15] Ni, K., & Nguyen, T. Image Superresolution Using Support Vector Regression. *IEEE Transactions On Image Processing*, [en línea]; 2007, [consulta: marzo 2017]. Disponible en: <<http://dx.doi.org/10.1109/tip.2007.896644>> pp 1596-1610.
- [16] Olivier, R., & Hanqiang, C. Nearest Neighbor Value Interpolation. *International Journal Of Advanced Computer Science And Applications*, [en línea]; 2012, [consulta: febrero 2017]. Disponible en: < <http://dx.doi.org/10.14569/ijacsa.2012.030405>>
- [17] Sharma, R. (2014). *An Adaptive Image Zoom Algorithm Using Interpolation Techniques* (Maesters of Science in Mathematics and Computing). Schol of Mathematics and Computer Applications Thapat University.
- [18] *SIFI Image Database*. *Sipi.usc.edu*. Retrieved, [en línea]; 14 March 2016, [consulta: marzo 2016]. Disponible en: < from <http://sipi.usc.edu/database/>>
- [19] Smith, P. Bilinear interpolation of digital images. *Ultramicroscopy*, [en línea]; 1981, [consulta: marzo 2016]. Disponible en: <[http://dx.doi.org/10.1016/s0304-3991\(81\)80199-4](http://dx.doi.org/10.1016/s0304-3991(81)80199-4)> pp 201-204.
- [20] Truong, T., Wang, L., Reed, I., & Hsieh, W. Image data compression using cubic convolution spline interpolation. *IEEE Transactions On Image Processing*, [en línea]; 2000. [consulta: marzo 2016]. Disponible en: <<http://dx.doi.org/10.1109/83.877222>> pp 1988-1995.

- [21] Zhang, H., Tian, X., & Chen, Y. A video structural similarity quality metric based on a joint spatial-temporal visual attention model. *Journal Of Zhejiang University SCIENCE A*, [en línea]; 2009. [consulta: mayo 2017]. Disponible en: <<http://dx.doi.org/10.1631/jzus.a0920035>> pp 1696-1704.
- [22] García, E. and Osuna, R. *Fundamentos de Fotografía Digital*. 1st ed. [en línea]; 2015. [consulta: 23 de abril 2015]. Disponible en: <<http://www2.uned.es/personal/rosuna/resources/photography/ImageQuality/fundamentos.imagen.digital.pdf>>
- [23] Koren, N. (2013). *Understanding resolution and MTF*. [online] Normankoren.com. [en línea]; 2013. [consulta: 23 de abril 2015]. Disponible en: <<http://www.normankoren.com/Tutorials/MTF.html>>
- [24] Zentrum, D. and Raumfahrt, L. (n.d.). Fundamentos del procesamiento digital de imágenes. [en línea]; Argentina: Mendoza. [consulta: 21 de julio 2015]. Disponible en: <http://www.uabcs.mx/geologia/geo_bajamx/pr/Mehl_Fundamentos_PR.pdf>
- [25] Wwwuser.cnb.csic.es. (n.d.). CCD. [en línea]; 2013. [consulta: 23 de junio 2015]. Disponible en: <http://wwwuser.cnb.csic.es/~fotonica/Photonic_en/Review/ccd1.htm>
- [26] Pinilla, C., Ariza, F. and Peláez, J. (n.d.). *Obtención de la Función de Dispersión Puntual (PSF) en Imágenes Spot Convencionales*. 1st ed. [en línea]; España: Santiago Castaño Fernández y Antonio Quintanilla Rodenas. [consulta: 23 de febrero 2015]. Disponible en: <<http://www.aet.org.es/congresos/viii/alb93.pdf>>
- [26] T. Blu, P. Thevenaz and M. Unser. (2001). MOMS: Maximal-Order Interpolation of Minimal Support, *IEEE Trans. Image Proc.*, pp 1069-1080.
- [27] B. Vrcelj and P. P. Vaidyanathan. (2001). Efficient Implementation of All-Digital Interpolation, *IEEE Trans. Image Proc.*, pp 63-1646.
- [28] K. Ichige, T. Blu and M. Unser. (2003). Interpolation of Signals by Generalized Piecewise-Linear Multiple Generators. In *Proceedings of ICASSP*, VI-261-VI-264.
- [29] B. P. Lathi. (1998). *Modern Digital and Analog Communication Systems*. New York: Holt Rinehart and Winston.
- [30] M. Unser, A. Aldroubi and M. Eden. (1993). B-Spline Signal Processing. I: Theory, *IEEE Trans. Signal Proc.*, pp 821-833

- [31] M. Unser, A. Aldroubi and M. Eden. (1993). B-Spline Signal Processing. II: Efficient Design and Applications, *IEEE Trans. Signal Proc.*, pp 834-848.
- [32] E. Meijering and M. Unser. (2003), A Note on Cubic Convolution Interpolation, *IEEE Trans. Image Proc.*, pp 477-479.
- [33] J. H. Shin, J. H. Jung, and J. K. Paik. (1998). Regularized Iterative Image Interpolation and Its Application to Spatially Scalable Coding, *IEEE Trans. Consumer Electronics*, pp 44, 1042-1047.
- [34] W. Y. V. Leung and P. J. Boney. (2001). Static Interpolation of Sampled Images, *Optical Eng.*, pp 40, 547-553.
- [35] E. A. El-Shazly, O. Zahran, S. M. Elaraby [et al]. (2010). Automatic Detection of Buried Landmines Using Cepstral Approach. *In Proceedings of Electrical and Computer Systems Engineering Conference*.
- [36] E. A. El-Shazly, O. Zahran, S. M. Elaraby [et al]. (2010). Cepstral Detection of Buried Landmines from Acoustic Images With a Spiral Scan. *In Proceedings of Electrical and Computer Systems Engineering Conference, Cairo*, pp 97-102.
- [37] F. E. Abd El-Samie, (2009). Detection of Landmines from Acoustic images Based on Cepstral Coefficients, *J. Sensing Imaging*, pp 10, 63-77.
- [38] U.S. Khan, W. Al-Nuaimy and F. E. Abd El-Samie, (2010), Detection of Landmines and Underground Utilities from Acoustic and GRP Images with a Cepstral Approach, *J. Visual Commun. Image Repr.*, pp 21,731-740.
- [39] F. G. Hashad, T.M. Halim. S. M. Dib, B. M. Sallam and F.E. Abd El-Samie, (2010), Fingerprint Recognition Using Mel-Frequency Cepstral Coefficients, *Pattern Recog. Image Anal.*, pp 20,360-369.
- [40] P. Thevenaz, T. Blu and M. Unser, (2000), Interpolation Revisited, *IEEE Trans. Med. Imaging*, pp 9, 739-758.
- [41] M. Unser, (1999), Splines: A Perfect Fit For Signal and Image Processing. *IEEE Signal Proc.*
- [42] J. K. Han and H. M. Kim, (2001). Modified Cubic Convolution Scaler with Minimum Loss of Information, *Optical Eng.*, pp 40, 540-546.

- [43] G. Ramponi. (1999). Warped Distance for Space Variant Linear Image Interpolation, *IEEE Trans. Image Proc.*, pp 8, 629-639.
- [44] A. K. Katsaggelos, R. Molina and J. Mateos, (2007). "Super resolution of images and video", San Rafael, CA, US, Morgan & Claypool Publishers, 1st ed.
- [45] C.A. Segall, R. Molina and J. Mateos, "High-resolution images from low-resolution compressed video", *IEEE Signal Process. Mag.*, Vol 20, pp. 37-48,2003
- [46] D. Mukherjee, B. Macchiavelo, and R. L. de Queiroz, (junio.2007). "A simple reversed-complexity Wyner-Ziv video coding mode base don a espacial reduction frame", Proc. IS&T/SPIE symp. On Electronic Imaging. Visual Communications and Image Processing, San Jose, CA, USA, SPIE vol. 6508, pp.65081 Y1-12,
- [47] B. Girod, A. Aron, S. Rane and D. Rebollo-Monedero, (junio 2005). "Distributed video coding", *Proccedings of the IEEE*, vol.93, no 1, pp.71-83.
- [48] B. Girod, R. L. de Queiroz, and D. Mukherjee, (junio 2005), "Motion based side information generation for a scalable Wyner-Ziv video coder", *Proc. IEEE Intl. Conf. On Image Processing*, vol.1, pp.825-828..