

UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS



TESIS

**“EVALUACIÓN DEL DESEMPEÑO DE 2 MÉTODOS DE SÚPER RESOLUCIÓN
APLICADO A IMÁGENES MÉDICAS EN 2D”**

PARA OBTENER EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS

ELABORADO POR
ENZO CAMARGO FERNÁNDEZ - BACA

ASESOR
Mg. SAMUEL OPORTO DÍAZ

LIMA - PERÚ

2016

DEDICATORIA

A Dios, mi patria, familia y a todos los hombres y mujeres de buena voluntad que día a día dan lo mejor de sí, aun sus vidas por lograr una patria y sociedad feliz, de respeto, cooperación, progreso y paz para todos y en la que podamos cumplir sus sueños y anhelos.

AGRADECIMIENTO

Gracias a Dios porque es mi roca y fortaleza, fuente de amor y vida,

Gracias a mis padres Walter y Vilma, cuyo valioso e incondicional amor, apoyo, ejemplo y sabiduría han guiado mis pasos, cada logro se los debo a ellos, no siendo suficientes las palabras para expresar el agradecimiento hacia ellos,

Gracias a mis hermanos y familia por la unión, amistad, soporte y fraternidad, haciendo la vida más bella,

Gracias a la UNI, mi alma mater, esperando poder retribuir lo recibido, con creces a mi patria y sociedad,

Gracias a todas las personas valiosas que han contribuido y apoyado mi desarrollo personal y profesional.

INDICE

DEDICATORIA	i
AGRADECIMIENTO	ii
INDICE	iii
ABSTRACT	xvi
PROLOGO.....	xviii
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	1
1.1 Descripción de la situación problema	1
1.1.1 Imágenes de menor resolución dificultan un diagnóstico temprano. 1	
1.1.2 Limitación de los métodos tradicionales para mejorar la calidad de una imagen médica	3
1.2 Formulación del problema	6
1.3 Objetivos.....	8
1.3.1 Objetivo general:	8
1.3.2 Objetivo específico	8
1.4 Justificación	9
1.5 Aplicaciones de procesamiento de imágenes en el país	10
1.6 Hipótesis.....	11
1.6.1 Hipótesis General	11
1.6.2 Hipótesis específica.....	11
1.6.3 Criterio de Validación de Hipótesis.....	12
1.7 Antecedentes.....	12
CAPÍTULO II: FUNDAMENTO TEORICO	16
2.1 Fundamentos de Imágenes Médicas.....	17
2.1.1 Concepto de Imagen Médica.....	17
2.1.2 Imágenes médicas en el Espectro Electromagnético	18
2.1.3 Clasificación de imágenes medicas.....	19
2.1.4 Mejor Calidad de imagen pero más radiación	31

2.1.5	Como la radiación afecta el ADN.....	32
2.2	Fundamentos de Imágenes digitales	34
2.2.1	Concepto de una Imagen Digital	34
2.2.2	Representación Matemática de una Imagen Digital	43
2.2.3	Que es la Calidad de una imagen digital	44
2.2.4	Fenómenos que afectan la calidad de una imagen	51
2.3	Fundamentos de Súper Resolución.....	53
2.3.1	Clasificación de Súper Resolución	53
2.3.2	Otros conceptos de Súper Resolución:	62
CAPÍTULO III: FUENTE DE DATOS		63
3.1	Recolección y Fuentes de Datos	63
3.2	Imágenes Aeroespaciales	63
3.3	Imágenes Médicas.....	66
CAPÍTULO IV: DESARROLLO DE LA SOLUCION		69
4.1	Cómo funciona la solución planteada	69
4.2	Formulación matemática de la Solución	71
4.3	Esquema general de la solución.....	72
4.3.1	Análisis Efecto – Causa – Solución en el proceso de obtener una imagen en Súper Resolución (SR) a partir de una imagen en baja resolución (LR)	76
4.3.2	Vista general de los 2 métodos de súper resolución: Iterative Kernel Steering (IKS) e Iterative Back Projecting (IBP)	77
4.4	Método de Súper Resolución basado en varias imágenes: Iterative Back Projecting.....	80
4.4.1	Esquema del Método IBP	83
4.4.2	Modelamiento matemático del Método IBP	84
4.4.3	Diagrama de Flujo del método IBP	85
4.4.4	Ejemplos de aplicación del método IBP	86

4.5	Método de Súper Resolución basado en una imagen: Iterative Kernel Steering (IKS).....	90
4.5.1	Código de programa en MATLAB de la regresión Kernel tradicional aplicado a una función aleatoria de X.....	92
4.5.2	Esquema de la regresión Kernel	93
4.5.3	Modelamiento matemático del Método de Regresión Kernel	95
4.5.4	Diagrama de Flujo	101
4.5.5	Ejemplos de aplicación de Método IKS	103
	CAPÍTULO V: EXPERIMENTACIÓN	104
5.1	Pruebas sintéticas	104
5.2	Interpolación de una imagen con varios métodos	108
5.3	Eliminar el ruido de las Imágenes (Image Denoising).....	111
5.4	Eliminar el ruido combinando métodos.....	112
5.5	Interpolación de imágenes.....	116
5.6	Súper Resolución con sub muestreo y ruido	117
5.7	Súper Resolución adicionando el PSF	120
5.7.1	Ejecución de Iterative Back Projecting y resultados de corrida ...	120
5.8	Súper Resolución con PSF, sub muestreo y ruido	125
5.9	Resumen de resultados obtenidos	130
5.10	Contrastación de la Hipótesis	132
	CONCLUSIONES	133
	RECOMENDACIONES.....	137
	REFERENCIAS BIBLIOGRÁFICAS.....	140
	GLOSARIO DE TÉRMINOS	144
	ANEXOS.....	148
A.	Programas desarrollados en Matlab.....	148
A.1	Fuentes de modulo principal del IKS	148
A.2	Minimizar ruido con IKS.....	151
A.3	Función de soporte: Cálculo de PSNR # 1	154

A.4 Funciones de Soporte: Cálculo de PSNR # 2.....	156
A.5 Pasar a Escala de grises	158
B. Programas C++ del método de super resolución IBP [5].....	160
C. Resultados de programas en C++ ejecución en IBP	188
C.1 Resultados de ejecución con IBP #1	188
C.2 Resultados de ejecución con IBP #2.....	190

Lista de Figuras

Figura 1: Una imagen en LR dificulta el diagnóstico médico.	3
Figura 2: Detalles de la reconstrucción utilizando 3 imágenes ortogonales.....	5
Figura 3: Proceso para obtener una imagen digital y los fenómenos que se presentan.....	6
Figura 4: Referencias de MRI en libros y publicaciones(MRI: Magnetic resonancia imaging).	14
Figura 5: Referencias en libros y publicaciones acerca de Súper Resolución (súper resolution en inglés).....	15
Figura 6: El espectro electromagnético es ilustrado con los niveles de Energía en eV para rayos X, mamografía, entre otros.	18
Figura 7: Producción de Bremsstrahlung.	21
Figura 8: Grafica de Rayos X: Energía (KeV) en el Eje X, Salida relativa de Fotones en el Eje Y.	21
Figura 9: La radiografía del pecho muestra un contraste entre el corazón (H) y los pulmones (L), se aprecia un tumor (T).	22
Figura 10: Imagen Médica de una Radiografía Dental.....	24
Figura 11: Vista medio lateral oblicuo, este paciente muestra una cantidad moderada de tejido fibroglandular.....	25
Figura 12: La imagen del epigastrio muestra dos áreas de atenuación (M) confirmando una múltiple metástasis hepática de un tumor estromal gastrointestinal.....	26
Figura 13: Una imagen de ultrasonido transversal de la vesícula muestra un cálculo biliar (la flecha) porque las ondas sonoras no pueden penetrar el cálculo biliar.....	28
Figura 14: Mayor resolución de la imagen medica con el uso de mayor radiación. El promedio de la exposición detectada de los rayos X por imagen es: a) 0.16 uR, b) 1.6 uR, c) 16uR, y 24uR.	31

Figura 15: Daño en la base del nucleótido, causado por los rayos X	33
Figura 16: Rotura de hebra de ADN como consecuencia de los Rayos X	33
Figura 17: Clústeres de daño e incidentes aislados seguidos a una interacción de radiación al ADN	34
Figura 18: Posición de los Pixeles en una imagen en escala de grises...	37
Figura 19: Representación de una Orquídea peruana como imagen digital en una matriz de pixeles.	38
Figura 20: Histograma de la imagen en escala de grises de la Orquídea peruana	39
Figura 21: Orquídea peruana en peligro de extinción que crece en Machupicchu	39
Figura 22: Posición de los Pixeles en una imagen a colores RGB	41
Figura 23: Imagen de la Orquídea peruana a color representada por una matriz de pixeles y cada pixel con valores R, G Y B	42
Figura 24: Sistema de coordenadas X, Y en el procesamiento digital de imágenes, donde (X=0, Y= 0) se ubica en la parte superior izquierda	44
Figura 25: Adición de efectos simulados de submuestreo, PSF y ruido a la imagen de la Orquidea peruana	45
Figura 26 : Fenómeno de desenfoque que afecta a la imagen. Izquierda: desenfoque de movimiento. Derecha: desenfoque óptico	53
Figura 27: Esquema General de la Tesis que plantea el problema y su solución utilizando Súper Resolución con los métodos Iterative Kernel Steering(IKS) e Iterative Back Projecting (IBP)	71
Figura 28: Diagrama de Bloques basado en la ecuación 4.1	72
Figura 29: Cuadro Resumen sobre la interrelación entre los efectos, causas y soluciones para obtener una imagen SR a partir de una imagen LR.	76

Figura 30: Diagrama de Flujo General de ambos métodos	79
Figura 31: Diagrama esquemático de algoritmo IBP de Súper Resolución	83
Figura 32: Diagrama de Flujo de la estimación del Desplazamiento entre dos Imágenes	84
Figura 33: Diagrama de flujo propuesto para el algoritmo IBP.	85
Figura 34: Creación de prueba sintética basada en una imagen original en colores(a), luego se convierte a Escala de Grises (b), y finalmente se crea la imagen de menor resolución (c) llamada imagen LR.	87
Figura 35: Ejemplo de método de Súper Resolución IBP aplicado a una imagen fotográfica	88
Figura 36: Ejemplo de método de Súper Resolución IBP aplicado a una mamografía	88
Figura 37: Acercamiento a Figura 36, se puede observar en la imagen de Súper Resolución (b) que los bordes están mejor definidos. 89	
Figura 38: Ejemplo de Regresión Kernel aplicado a una función aleatoria	93
Figura 39: Esquema de Súper Resolución de una imagen de 3x3.	95
Figura 40: Distribución de los pixeles y Grafico de Kernel gaussiano en 2D.	97
Figura 41: Variación del Kernel en función de la matriz de suavizado.	98
Figura 42: (a) Kernel clásico. (b)Kernel adaptable.	99
Figura 43: Ejemplos de Kernels adaptables, donde se aprecia diferentes matrices de suavizado según la posición del pixeles	100
Figura 44: Digrama de Flujo de Metodo IKS para obtener Súper Resolución a partir de 1 sola imagen	101
Figura 45: Ejemplos de aplicación de Método de Regresión de Kernel Adaptativo.....	103

Figura 46: Esquema de las pruebas sintéticas y su relación con una aplicación real.....	106
Figura 47: Pruebas sintéticas aplicadas a imagen: Cara oculta de la luna y la creación de la imagen en baja resolución LR	107
Figura 48: Imagen capturada por Vehículo de exploración de la NASA, exposición a contacto geológico y la creación de la imagen en baja resolución LR.....	107
Figura 49: Basados en la figura 47, se han aplicado diversos métodos para la mejora de las imágenes (a),(b), (c) y (e) obtenidas mediante interpolaciones respectivas; (d) y (f) son el resultado de métodos de súper resolución.....	108
Figura 50: Mejora de la imagen de la Figura 48 aplicando 6 métodos interpolación y Súper Resolución.	110
Figura 51: Performance de los métodos para eliminar el ruido en imágenes médicas de mamografía. RMSE de imágenes (b) – (f) es respectivamente y el PSNR.....	113
Figura 52: Acercamiento a Figuras 51 (a), (b), (c) y (d) respectivamente	114
Figura 53: LR se obtiene submuestreando HR. Combinación de métodos, se aprecia una mejora en lo obtenido en (d) respecto a lo obtenido en (c).....	115
Figura 54: Interpolación a la mamografía, en este caso, solo se le hace sub muestreo en 3 veces, luego se aplica IKS según los parámetros indicados en (a) y (b).....	116
Figura 55: Súper Resolución con sub muestreo y ruido, a la cara oculta de la luna, con los 2 métodos IBP e IKS.	118
Figura 56: Acercamiento a Imágenes de la Luna en LR y en SR aplicando Iterative Back Projecting e Iterative kernel Steering	119
Figura 57: Se toma la imagen de Tabla 7-5 y se obtiene LR, de las figuras (b)-(d) se obtiene PSNR = 17.1520, 18.3677, 17.6071	

respectivamente; RMSE = 35.3950, 30.7718, 33.5882 respectivamente	124
Figura 58: Aplicación con ruido y sin ruido, adicionando sub muestreo y PSF.	128
Figura 59: Aplicación con ruido y sin ruido, adicionando sub muestreo y PSF.	129

Lista de Tablas

Tabla 1: Características de adquisición de cada sistema de imágenes médicas	30
Tabla 2: Resumen del detector de exposición y número de quanta per mm ² usado para crear imágenes mostradas en la Figura 14....	32
Tabla 3: Valores de atributos para imagen en escala de grises de la Orquídea peruana	38
Tabla 4: Valores de atributos para imagen en colores RGB de la Orquídea peruana	42
Tabla 5: Clasificación de los Métodos de Súper Resolución	54
Tabla 6: Lista de imágenes aeroespaciales utilizadas para pruebas sintéticas.....	65
Tabla 7: Lista de imágenes médicas utilizadas para el proyecto	68
Tabla 8: Valores PSNR para la aplicación de métodos de interpolación y súper resolución a la imagen LR de la Figura 47 (b) de la cara oculta de la luna.	109
Tabla 9: Tabulación de resultados de los diferentes métodos de SR	111
Tabla 10: Resumen de resultados obtenidos basados en experimentaciones realizadas con ambos métodos de Súper Resolución IBP e IKS	131

DESCRIPTORES TEMATICOS

1. Súper Resolución de imágenes
2. Procesamiento de Imágenes
3. Calidad de una imagen digital: PSNR
4. Algoritmo de Súper Resolución : Iterative Back Projection
5. Algoritmo de Iterative Kernel Steering
6. Obtención de una imagen digital
7. Fenómenos que afectan la calidad de la imagen
8. Pruebas sintéticas

RESUMEN

La presente tesis tiene como finalidad evaluar el desempeño de 2 métodos de Súper Resolución aplicados a imágenes médicas digitales en 2D: El **método Iterative Back Projecting (IBP)** y el **método Iterative Kernel Steering (IKS)**, utilizando el indicador objetivo denominado PSNR (Peak Signal to Noise Ratio) para evaluar el desempeño de cada método en las pruebas sintéticas, el cual se calcula basado en las diferencias que existen entre la imagen original (HR) y la imagen en Súper Resolución (SR) obtenida.

La Súper Resolución es una tecnología diseñada en el campo del procesamiento de imágenes, diseñada con la finalidad de obtener una imagen de mayor resolución hasta 8 veces mayor a la imagen o imágenes de entrada, mediante el modelamiento de los fenómenos que afectan su obtención y minimizar sus efectos, mediante la adición de información que permita reconstruir la imagen real (HR) de forma más aproximada posible, logrando de esta manera la imagen en Súper Resolución (SR), esto es graficado en la **Figura 27, donde se presenta el Esquema general de la Tesis, presentando el problema y su solución mediante Súper Resolución.**

En la presente tesis se logra mejorar la calidad de la imagen hasta 3 veces (3x) en relación a la imagen inicial obtenida, es decir, si la imagen de entrada tiene una resolución de 240 x 240 pixeles, se incrementa hasta en 720 x 720 pixeles, 3 veces en cada dirección ancho y alto.

En relación al desempeño obtenido de métodos evaluados se demuestra que el método IBP (varias imágenes de entrada) es mejor el IKS (1 sola imagen de entrada), en los que no hay presencia de ruido, es decir, PSNR (IBP) >

PSNR (IKS), asimismo hay casos que los que predomina el ruido, se observa que el IKS tiene mejores resultados obteniéndose $PSNR (IBP) < PSNR (IKS)$.

El aporte del presente trabajo y su desarrollo es trascendental por las potenciales aplicaciones futuras en mamografía, ecografía, rayos X, digitalización de placas de imágenes médicas, así como aplicaciones en la industria, seguridad civil y militar, exploración minera, forestal, hídrica y recursos en general, esto debido a que las imágenes al ser digitalizadas se pueden procesar y explotar de muchas maneras y extraer información relevante según su área de aplicación para la toma de decisiones.

Palabras Clave: Super Resolución, Método Iterative Back Projecting (IBP), Método Iterative Kernel Steering (IKS), mejora de la resolución de una imagen digital, fenómenos que afectan la resolución de la imagen, ruido, sub muestreo, desenfoque, interpolación de imágenes, Pico de Señal Ratio–Ruido o Peak Signal to Noise Ratio en inglés (PSNR), Punto de Función de Propagación o Point Spread Function (PSF), Pruebas sintéticas, imágenes médicas digitales, aplicaciones para el desarrollo nacional, aplicaciones en imágenes médicas, aplicaciones en imágenes satelitales.

ABSTRACT

Digital images have been widely applied in medicine like X rays, ecography, MRI among others. However, the image analisys can be limited by motion blur, movement induced by patient behaviour such little kids, low performance of equipment and the noise inherent within the image sensors, that every image digital equipment has. Therefore, the improvement of image quality becomes an important task for image analisys. I present super resolution methods as a important solution for improvement of image quality and finally evaluate their performance in medical images, also with photos.

Super resolution is the process in which one or more images are used for the construction of just one superior resolution image more resolution than the input images. The input images are known as a low resolution image (LR: Low resolution image) and the result image are known as super resolution image (SR: Super resolution image). The applications of this technology are many besides medical are military, security, robotics, UAVs among others. NASA also uses super resolution for the image obtained from telescopes, satelites and in the aerospace research. Peru has realized important advances in the aerospace field, UNI has launched its own satellite and it was the first satellite launched by our Country history. Therefore this field of image processing and super resolution is so important for national development.

Here, we develop a super resolution framework, one method is based on frecueny domain, uses iterative back projecting to achieve the super resolution image using many input images; and the another is based on methods that need just one image. A quantity comparison in computation time, visual performance and PSNR for measure objectively the image quality are

performed. The results shows a great improvement of the resolution in 3x, for example one image of 171x171 pixels is improved to 512x512 pixels and the low resolution images of 30.19 dB for synthetic images, and a considerable visual improvement in sharpness and visual details for real medical images and real pictures images. The method based on many input images are more efficient and get better results than the method based on just one input image.

Keywords: Super resolution, Iterative back projecting method (IBP), Iterative Kernel Steering, improve of quality digital image, phenomena that affects image resolution, noise, aliasing, downsampling, upsampling, interpolation, peak signal to noise ratio (PSNR), PSF (Point Spread Function), synthetic test, digital imaging, applications for national development, medical imaging, satelital imaging.

PROLOGO

En el ámbito del procesamiento digital de imágenes **un CCD** es un concepto que representa a un equipo de adquisición de imágenes en general como una cámara fotográfica, equipo de rayos X, MRI, o un telescopio, entre otros; **el objeto o escena existente del mundo real**, del cual se requiere obtener una o varias imágenes, **las pruebas sintéticas** son pruebas realizadas en un escenario controlado y está conformado por una **Imagen Original Real (HR)**, el cual es un concepto teórico creado para pruebas sintéticas, que representa la imagen obtenida del objeto o escena existente (high-resolution imagen, por sus siglas en inglés, ver [6]), a partir de la imagen HR se crea la imagen o las **imágenes de baja resolución (Low resolution image: LR)** en el caso de aplicaciones reales la imagen obtenida es la imagen en LR; **la imagen Súper Resolución (SR)**, es la mejor aproximación posible a la HR, lograda por los métodos de Súper Resolución a partir de una o varias imágenes de baja resolución (LR), calculándose esta aproximación con el indicador objetivo llamado **PSNR (Peak Signal to Noise Ratio)**, el cual es una comparación entre la imagen HR y la imagen SR, o la imagen a la que se desea evaluar su calidad. En la medida que la imagen SR se aproxima más a HR, el PSNR es de mayor valor, por lo que un PSNR más alto significa mejores resultados. . **En el caso de aplicaciones reales no es posible calcular el PSNR** porque solo se tiene la imagen en baja resolución (LR) y no se tiene como dato la Imagen Original Real (HR)

En la presente tesis, se logra un aumento de la resolución en 3x al aplicar las técnicas de Súper Resolución, por ejemplo con una imagen de 171 x 171 pixeles se obtiene 512 x 512 pixeles, con ambos métodos, el método de IBP y el método IKS; adicionalmente se observa en las pruebas realizadas que la

calidad de la imagen utilizando IBP es mejor que el método de Kernel en los casos indicados en Capítulo 0, basados en el indicador objetivo PSNR.

Los algoritmos de Súper Resolución se clasifican en 3 (Sección 2.3); por el número de imágenes: basado en 1 sola imagen o basados en varias imágenes; por el campo de aplicación: aplicación a imágenes médicas, imágenes satelitales, entre otros; por el dominio de las imágenes: dominio de la frecuencia, dominio del espacio, entre otros. Destacan los trabajos de Hiroyuki Takeda, Peyman Milanfar [19] quienes proponen el Método Iterative Kernel Steering, así como el Michael Irani y Samuel Peleg [12] los cuales desarrollan el Método de Iterative Back Projecting basado en la forma como operan los tomógrafos para obtener la imagen. Ambos métodos se evalúan, prueban y comparan en la presente tesis. El autor participo en el desarrollo e implementación del método IBP, y otros métodos de SR, los cuales no se presentan en el trabajo actual, indicados en las referencias [5] y [7].

En el desarrollo de la presente TESIS tenemos en el Capítulo I, el Planteamiento del Problema; en el Capítulo II, el marco teórico; en el Capítulo III, se especifica la fuente de datos; en el Capítulo IV: se desarrolla la solución; en el Capítulo V: se desarrolla la experimentación y se presenta las Conclusiones y Recomendaciones, luego se presentan las Referencias Bibliográficas y los Anexos correspondientes.

En el CAPÍTULO I, se plantea el problema debido a la limitación de los métodos tradicionales para mejorar la resolución de una imagen médica, la cual dificulta el diagnóstico temprano de enfermedades en algunos casos, debido a que la misma imagen es una herramienta de diagnóstico básica, resolviéndose el problema usualmente repitiendo la prueba con mayor radiación, lo cual puede alterar el ADN, entre otras alternativas realizadas; por

ello la alternativa adicional de la mejora en la resolución de la imagen utilizando los métodos de Súper Resolución son de gran importancia. Asimismo se plantea la hipótesis que un método de Súper Resolución con varias imágenes de entrada obtiene mejores resultados que un método que se basa en una sola imagen de entrada.

En el CAPÍTULO II, En el marco teórico, se revisa los fundamentos teóricos de imágenes médicas, las imágenes digitales, y los métodos de Súper Resolución de imágenes y video.

En el CAPÍTULO III, se indica las fuentes de datos de las imágenes médicas e imágenes aeroespaciales, las cuales luego se utilizaran en los ejemplos y experimentaciones, de los Capítulos IV y V. Se especifican sus tamaños, formatos, entre otros.

En el CAPÍTULO IV, se desarrollan cada uno de los métodos, empezando con el método Iterative Back Projecting (IBP) y luego el Método Iterative Kernel Steering (IKS), se presenta el modelo matemático, el diagrama de flujo, así como el resultado de las pruebas unitarias en cada uno de los métodos

En el CAPÍTULO V: se presenta la experimentación para evaluar los 2 métodos en un entorno controlado llamado pruebas sintéticas a imágenes médicas e imágenes aeroespaciales, evaluando el desempeño mediante el indicador objetivo calculado PSNR y RMSE. Se realizan pruebas con imágenes médicas de mamografías principalmente y de forma complementaria con imágenes aeroespaciales. Las pruebas sintéticas son creadas con Matlab, y la implementación del método IBP está en C++ y el método IKS esta implementado en Matlab. Luego se presentan los resultados y análisis de la aplicación de los métodos de Súper Resolución en imágenes

médicas. Finalmente se presentan las Conclusiones y recomendaciones, así como los potenciales trabajos futuros.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción de la situación problema

Una imagen medica es aquella que se obtiene de diversas partes y/o estructuras parciales o totales de un ser humano o animal, mediante un equipo de adquisición de imágenes médicas como los Rayos X, mamografía entre otros, y constituye la información básica y fundamental para diagnóstico de enfermedades. Se presentan las siguientes limitaciones o problemas:

1.1.1 Imágenes de menor resolución dificultan un diagnóstico temprano.

Una imagen médica es una unidad fundamental de diagnóstico médico, esta imagen dificulta un diagnóstico temprano cuando presenta carencias en las ampliaciones al visualizar detalles específicos.

La Figura 1 y Figura 2, se puede observar ejemplos en los cuales las imágenes en LR conllevan a dificultar el diagnóstico. En cada una de estas figuras, que son 2 tipos de exámenes médicos diferentes obtenidos de autores distintos, se observa el mismo problema o limitación.

En la Figura 1, se aprecia Cortes coronales de LR (e), HR (g) y SR FLAIRs con su correspondiente segmentación y LesionTOADS son mostrados. Es evidente que usando una imagen en baja resolución, afecta la segmentación e inclusive la corteza.

En la **Figura 2**, se aprecian imágenes en corte ortogonal de un cerebro, en los cuales se observan 2 métodos aplicados a las imágenes de la izquierda, según el autor no se especifica que implique un diagnóstico mejor, a pesar de ello podemos ver la mejora sustancial de la imagen de la derecha respecto de la primera.

Un ejemplo de la importancia de la calidad de imagen en el diagnóstico temprano es el cáncer de mama, el cual es uno de los más lesivos para las mujeres por su alta ocurrencia según [20]; si es diagnosticado tempranamente es curable y este diagnóstico se realiza inicialmente mediante imágenes médicas.

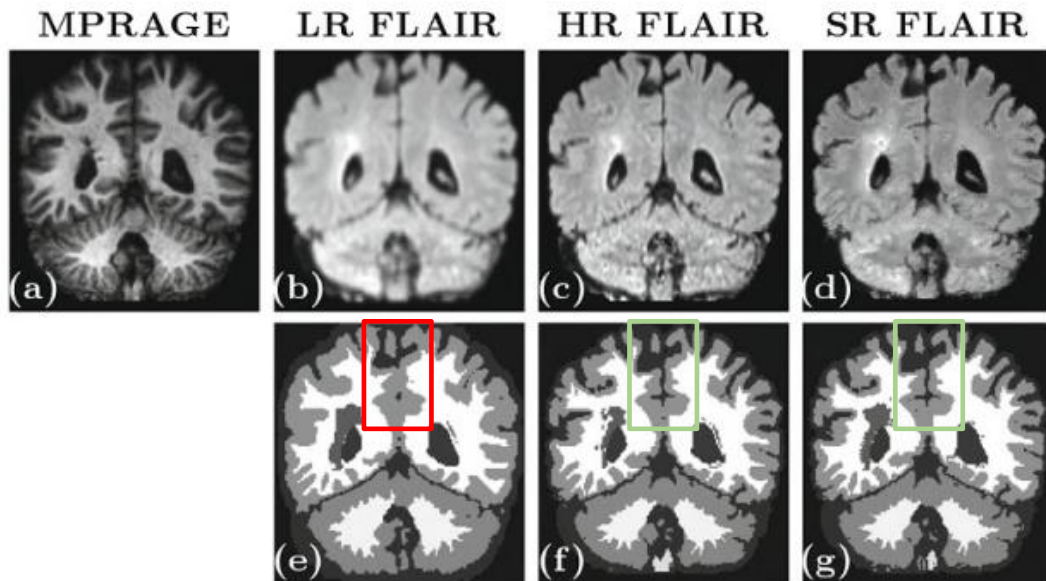


Figura 1: Una imagen en LR dificulta el diagnóstico médico.

Fuente: Basado en [18], página 733

Elaboración: Sabhal Mor Ostaig, Isle of Skye, cuadrículas añadidas por Enzo Camargo Fernandez Baca.

1.1.2 Limitación de los métodos tradicionales para mejorar la calidad de una imagen médica

Luego de obtener una imagen en LR desde un CCD, se realiza usualmente lo siguiente:

1. Ninguna acción adicional, es decir, continuar el procedimiento con la imagen obtenida para el diagnóstico

En el caso que se requiera mejorar la resolución de la imagen:

2. Se usa un mejor hardware para la toma de las imágenes; lo cual significa repetir la prueba en otro equipo de adquisición de imágenes médicas en el mismo lugar o desplazarse a otro Centro Medico, en caso sea factible.
3. Según [2], se incrementa la radiación para lograr mejores imágenes. La radiación y campos magnéticos puede alterar el ADN de un ser humano.
4. Repetir la prueba

El presente trabajo, presenta una alternativa adicional la cual es mejorar la calidad de la imagen mediante el procesamiento computacional de la imagen, específicamente con el uso de la tecnología de Súper Resolución (Ver 2.3)

Sobre los puntos 2 al 4, se puede decir que se ha llegado a un límite técnico-humano en la resolución de imágenes y video médicos, relacionado a la radiación y campos magnéticos que el ser humano puede recibir, siendo estos límites determinados en [2], en el caso pasarlo se podría alterar el ADN, y ello a su vez origina problemas críticos en el ser humano como el cáncer (Ver 2.1.5.1) ; estos límites dependen de la edad, condiciones de enfermedad o estados físicos (mujeres embarazadas) en las que la dosis de radiación debe ser mínima o nula.

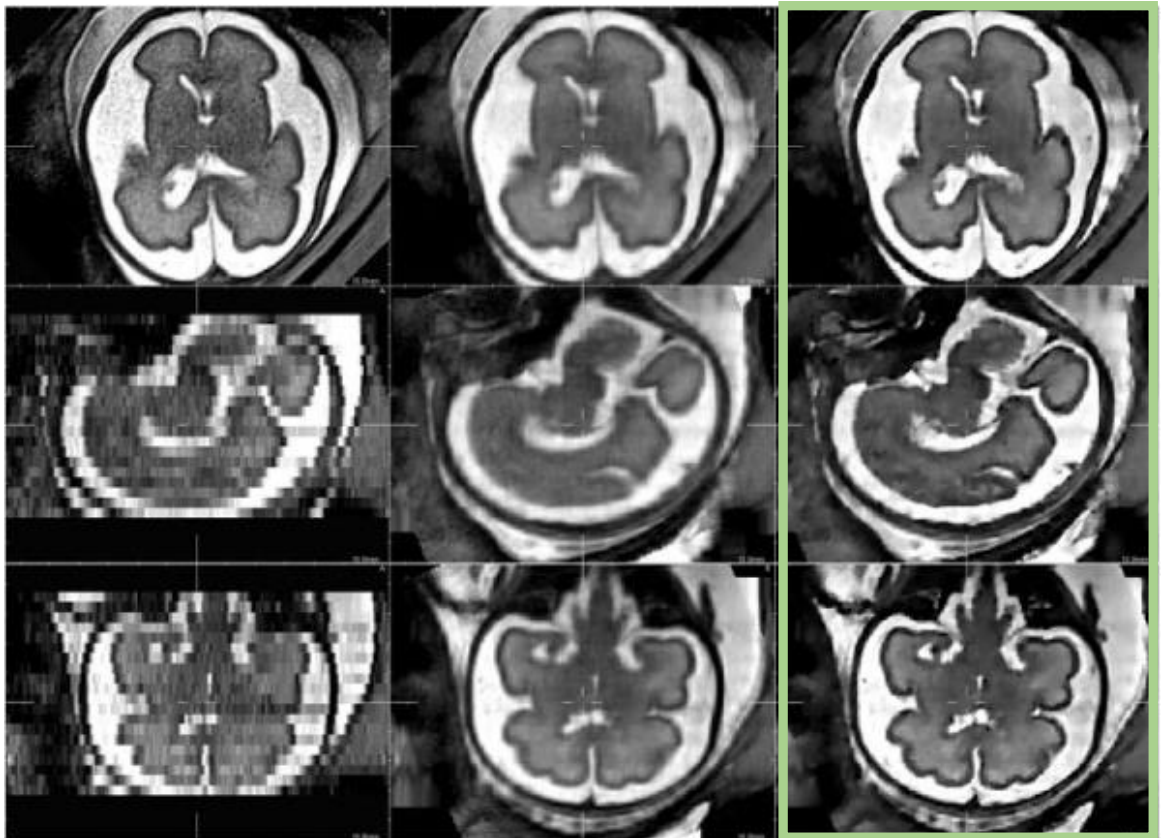


Figura 2: Detalles de la reconstrucción utilizando 3 imágenes ortogonales.

De izquierda a derecha: A) Lazo izquierdo, original imagen en LR B) Imagen en medio, reconstruida usando interpolación esparcida local C) Lado derecho, SR reconstrucción usando JCH regularización.

Fuente: [15] Medical Image Computing and Computer-Assisted Intervention - - MICCAI 2010

Elaboración: F. Rousseau, cuadrícula adicionada por Enzo Camargo Fernández - Baca

1.2 Formulación del problema

El proceso de obtención de las imágenes sufre fenómenos que afectan su calidad, estos fenómenos se presentan en todos los equipos de adquisición de imágenes (CCD); como un tomógrafo, un equipo de rayos X, un telescopio, una cámara fotográfica, entre otros.

Según la **Figura 3**, el proceso de obtener una imagen digital de baja resolución (LR) sigue diversas etapas y cada etapa es afectada por uno o varios fenómenos, como son el ruido, muestreo de continuo a discreto con aliasing, el sub muestreo, el desenfoque, la rotación y traslación.

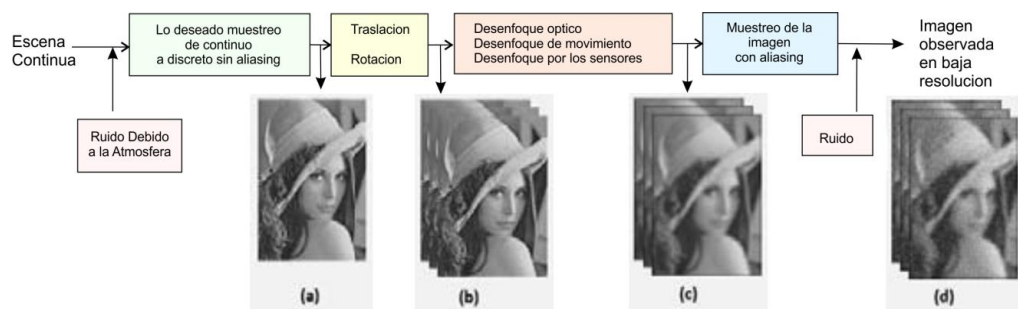


Figura 3: Proceso para obtener una imagen digital y los fenómenos que se presentan

Entrada: Objeto o Escena (Escena continua)

(a): Imagen digital con muestreo (continuo a discreto)

(b): Imagen digital con muestreo y desenfoque

(c): Imagen digital con muestreo, desenfoque y aliasing

Salida final:

(d): Imagen digital muestreada con desenfoque, aliasing y ruido = Imagen LR

Para obtener la imagen de Súper Resolución (SR), estos fenómenos son modelados matemáticamente, para luego ser optimizados, siendo posibles múltiples soluciones y por ello existen múltiples métodos de Súper Resolución.

La secuencia para obtener la imagen SR, es ir a la inversa del proceso indicado en Figura 3, optimizando y/o minimizando los fenómenos que afectan la calidad de la imagen, mediante la formulación matemática del problema y el despeje de la variable de interés.

1.3 Objetivos

1.3.1 Objetivo general:

Evaluar el desempeño de 2 métodos de Súper Resolución aplicados a imágenes médicas en 2D, utilizando el indicador objetivo PSNR.

Desarrollar alternativas adicionales a las tradicionales existentes para la mejora de la resolución de imágenes médicas en la etapa de post procesamiento, mediante métodos de Súper Resolución las cuales se encuentran en el estado del arte.

1.3.2 Objetivo específico

1. Registrar las imágenes, con la finalidad de colocarlas en el mismo eje de coordenadas, considerando que tienen una traslación y rotación, este paso es aplicable cuando se tienen varias imágenes de entrada.
2. Minimizar el efecto del ruido, ya sea atmosférico o ruido interno propio del equipo.
3. Minimizar el efecto del desenfoque de movimiento, sensores, y PSF.
4. Aumentar el muestreo, es decir, adicionar información a la imagen LR, basado en interpolación y/o otras técnicas que permita finalmente obtener una imagen con mayor resolución, llamada imagen en Súper Resolución (SR)
5. Realizar pruebas sintéticas
6. Evaluar el desempeño de cada método en base a criterios objetivos (PSNR) y subjetivos (percepción visual).
7. Demostrar la hipótesis planteada

El método de Súper Resolución se aplica en los pasos 2, 3, 4 y según el método puede aplicarse de forma secuencial, conjunta, en paralelo y en un orden determinado, lo común de los métodos es que todos buscan realizar estos pasos.

El paso 1, es un paso previo que prepara las imágenes de entrada para ser procesadas, colocándolas en un mismo sistema de coordenadas, y es realizada solo cuando se tiene más de una imagen de entrada en LR. En el paso 5, es la validación mediante las pruebas sintéticas. El paso 6 y 7 nos permiten evaluar el desempeño y demostrar la hipótesis planteada.

1.4 Justificación

Las imágenes son una fuente de información para el diagnóstico médico, y la mejora de la calidad de la imagen tiene un impacto directo en un mejor diagnóstico; lo cual influye de forma directa ó indirecta a lograr mayor salud e inclusive salvar vidas; aunque esta proposición no se demuestra por no formar parte del alcance.

Una mejor calidad de imagen médica (Ver 2.2.3) es de importancia para lo siguiente:

1. Mejorar el diagnóstico médico.
2. Reducción de tiempo de diagnóstico de una enfermedad, porque en algunos casos, la detección temprana hace factible su cura.

3. Detección de tumores, malformaciones de forma más rápida y confiable, mediante la amplificación de la zonas de la imagen y mejorando su resolución.
4. Reducción de costos en diagnóstico de enfermedades.
5. Equipos de baja performance pueden ser repotenciados, por la posibilidad de lograr mejores resultados en sus imágenes.
6. Digitalización de bases de datos de imágenes analógicas con mayor fidelidad y resolución.
7. Es un medio no invasivo de diagnóstico utilizado ampliamente por la tecnología médica actual.

1.5 Aplicaciones de procesamiento de imágenes en el país

El Perú requiere soluciones de procesamiento de imágenes y también específicamente soluciones en el campo de la Súper Resolución, en las siguientes actividades económicas y estratégicas para el desarrollo nacional:

- Imágenes médicas, diagnóstico; telemedicina, compresión, entre otros.
- Desarrollo aeroespacial, el cual es transversal a todas las actividades económicas del país.
- Imágenes satelitales de proyectos emblemáticos como el nano satélite UNI, posteriormente el satélite de observación terrestre denominado Perú SAT-1.
- Seguridad ciudadana y militar

- Agroindustria (clasificación de tamaños de frutas mediante imágenes, etc.)
- En general, todas las actividades económicas y de desarrollo del país.

Por ello es importante desarrollar esta tecnología en nuestro propio país siendo considerablemente mejor a la dependencia de tecnología externa, la cual implica sobrecostos e ineficiencia en nuestra industria.

1.6 Hipótesis

1.6.1 Hipótesis General

Es posible mejorar la resolución de las imágenes médicas con los métodos de Súper Resolución propuestos. Asimismo, se pueden obtener mejores resultados con el método Iterative Back Projecting (IBP), el cual se basa en varias imágenes de entrada, que el resultado obtenido con el método Iterative Kernel Steering (IKS) que se basa en una sola imagen de entrada.

1.6.2 Hipótesis específica

El método de Súper Resolución IBP basada en varias imágenes, mejora la resolución de imagen mejor que el método IKS.

1.6.3 Criterio de Validación de Hipótesis

$$\text{PSNR}_{(i)} > \text{PSNR}_{(j)}$$

Donde:

i: Es el método SR basado en varias imágenes

j: Es el método SR basado en una sola imagen

1.7 Antecedentes

Según [2], desde inicios del siglo XX y luego de 80 años de predominancia de los rayos X en filmina/pantalla, y luego en las siguientes 2 décadas, las imágenes médicas han experimentado el desarrollo y comercialización de abundantes nuevas tecnologías como Tomografía Computacional, imágenes MRI, angiografía digital, imagen por ultrasonido Doppler, entre otros proporcionando al radiólogo, mayores y mejores herramientas de Diagnóstico con mayor fiabilidad de detección y diagnóstico de enfermedades. Esto también se puede comprobar en la Figura 4, donde el termino MRI (Magnetic resonance imaging) recién se inicia su difusión y desarrollo en la bibliografía mundial desde la década del 1970 y un gran crecimiento en su difusión en la década de 1980 hasta la fecha.

La evolución significativa de la tecnología de imágenes médicas, es atribuible al avance de la ciencia en el conocimiento fundamental de los fenómenos físicos en el cual esta tecnología está basada primariamente y esto ha sido posible con el incremento del poder computacional y comunicación de datos de alta velocidad.

Ahora con la separación entre la captura de imágenes y su visualización en pantalla, hace posible que sean optimizados independientemente como 2 subsistemas separados, lo cual no era posible antes.

Esta separación genera oportunidades desde el campo del procesamiento de imágenes se trabajen y mejoren las imágenes y entre ellos destacan los métodos de Súper Resolución aplicados en el presente trabajo.

El artículo de Tsai y Huan [11] publicado en 1984, es generalmente considerado como el inicio de la Súper Resolución, su observación clave es seguida del hecho que si el sensor de una imagen tiene un muestreo bajo, el contenido de la frecuencia en 2D de una imagen será mostrada con Aliasing. Esto también se visualiza en el buscador de estadísticas de referencias bibliográficas de Google en la Figura 5, donde el termino súper resolución se empieza a utilizar ampliamente desde la década de 1980.

Posteriormente se mejoraron y desarrollaron más técnicas, métodos de súper resolución como Michel Irani y Samuel Peleg [12], quienes plantean un método que toma como fundamento inicial, la técnica del Back Projecting que se utiliza en la tomografía, a la cual llamaron Iterative Back Projecting por además considerar en su método el aporte del proceso iterativo, este método requiere varias imágenes de entrada. Asimismo autores como Hiroyuki Takeda plantean un “*framework*” para la aplicación de un método Iterative Kernel Steering, el cual tomando como base el método de Kernel tradicional lo mejora considerando adicionalmente aportes sustanciales como el análisis del entorno de cada pixel y utilizar información relevante cercana a este pixel para poder mejorar la imagen en su totalidad [2].

Google Books Ngram Viewer

Graph these comma-separated phrases: case-insensitive
between and from the corpus with smoothing of

Google books Ngram Viewer

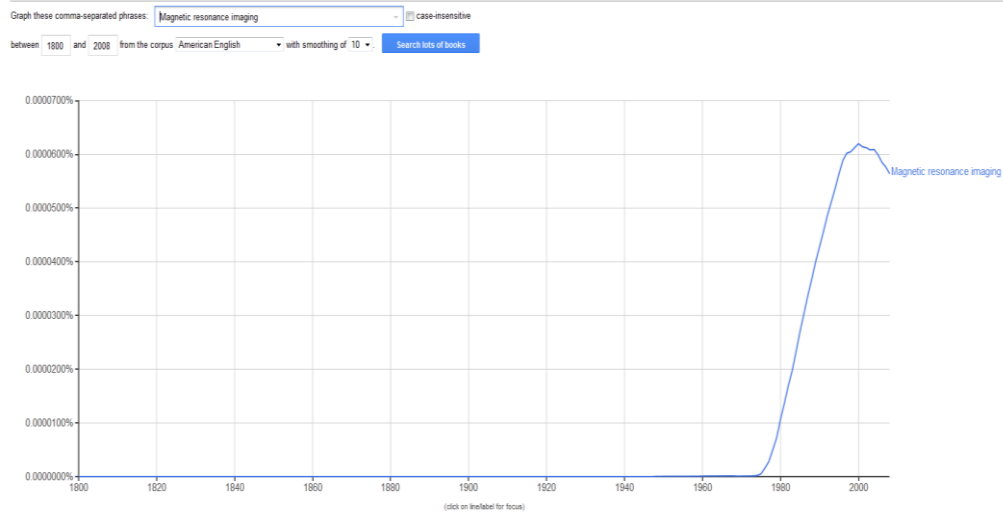


Figura 4: Referencias de MRI en libros y publicaciones(MRI: Magnetic resonancia imaging).

Fuente: Google (<https://books.google.com/ngrams>) Buscador de estadísticas de referencias bibliográficas

Elaboración: Enzo Camargo Fernández - Baca

Actualmente las técnicas de Súper Resolución se aplican en áreas de seguridad y satelital ampliamente, en el caso de imágenes médicas se encuentran aplicaciones a nivel académico y de investigación, obteniéndose resultados como el indicado en la Figura 1 y Figura 2, en la cuales se observa que una imagen en LR dificulta el diagnostico, y la imagen en Súper Resolución (SR) facilita el diagnostico. Todavía no se utiliza la súper resolución a nivel clínico de forma masiva, se espera que en los próximos años se inicie dicha aplicación.

En el presente trabajo, exploramos la aplicación de 2 métodos de súper resolución aplicado a imágenes médicas digitales en 2D capturadas, y evaluamos su desempeño a fin de compararlos.

Google Books Ngram Viewer

Graph these comma-separated phrases: case-insensitive
between and from the corpus with smoothing of

Google books Ngram Viewer

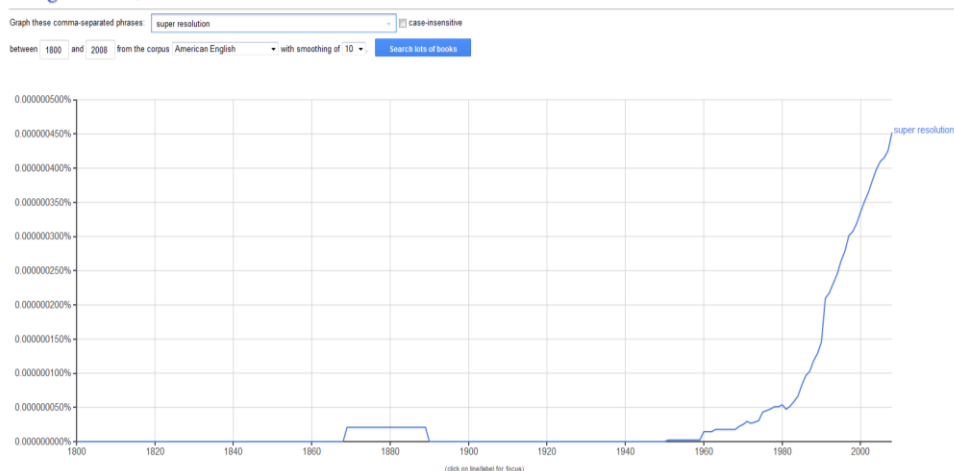


Figura 5: Referencias en libros y publicaciones acerca de Súper Resolución (súper resolution en inglés).

Fuente: Google (<https://books.google.com/ngrams>) Buscador de estadísticas de referencias bibliográficas

Elaboración: Enzo Camargo Fernández - Baca

CAPÍTULO II: FUNDAMENTO TEORICO

En este capítulo se presenta los fundamentos de imágenes médicas, comprendido por los tipos de tecnologías utilizadas para su obtención, los formatos, acciones que se realizan actualmente para obtener mejores imágenes médicas para mejorar el diagnóstico. Asimismo se presentan los fundamentos de Súper Resolución y sus métodos como una alternativa adicional a los métodos tradicionales en los cuales se logra una mejor resolución aumentando la radiación, asimismo se presenta los efectos de radiación de los rayos X en el ADN, con la finalidad de explicar por qué y cómo se afecta el ADN y qué consecuencias tiene cuando la exposición se realiza a mayores niveles del permitido según [2] y [13].

Luego presentamos los fundamentos de las imágenes digitales, como se obtiene una imagen, como se representa una imagen formada por pixeles y cada pixel a su vez representada por canales, y estos a su vez están representados por bits de 1s y 0s, por ejemplo cuando es formato de escala de grises se tiene 1 solo canal, y esto varía según los diversos formatos. Asimismo se explica como medir la calidad de una imagen digital cuando se realizan las pruebas sintéticas, es decir, entornos controlados de pruebas para desarrollar y observar los resultados de métodos o técnicas en análisis y/o estudio, y finalmente se analizan los fenómenos que afectan la calidad de una imagen desde el objeto o escena hasta la imagen obtenida (LR).

Posteriormente, se explica los fundamentos de la súper resolución, los cuales nacen a partir del entendimiento de los fenómenos que afectan la calidad de una imagen, su solución mediante una ecuación matemática en 4.2, la revisión bibliográfica que presentan los avances desarrollados en este campo, resueltos utilizando la estadística, hasta redes neurales y en general técnicas y filtros que permiten obtener los valores intermedios (interpolación) que añaden información a la imagen LR para obtener la imagen en SR. Finalmente se muestran las aplicaciones en súper resolución la cual es transversal a campos como imágenes médicas para diagnósticos, hasta seguridad, imágenes satelitales, entre otros.

2.1 Fundamentos de Imágenes Médicas

2.1.1 Concepto de Imagen Médica

Una imagen médica, es aquella que proviene del conjunto de técnicas y procesos usados para crear imágenes del cuerpo humano o parte de él, con propósitos clínicos, es decir, procedimientos médicos que buscan revelar, diagnosticar, examinar enfermedades o bien con propósitos científicos médicos, tales como el estudio de la anatomía física y metabólica. [2]

Las imágenes médicas, están muy relacionadas con muchas disciplinas médicas principalmente en la radiología, seguidamente de la endoscopia, la termografía médica, la fotografía médica y la microscopia y también con la física

Las imágenes médicas son obtenidas en formato DICOM, lo cual es explicado en el Glosario de términos.

2.1.2 Imágenes médicas en el Espectro Electromagnético

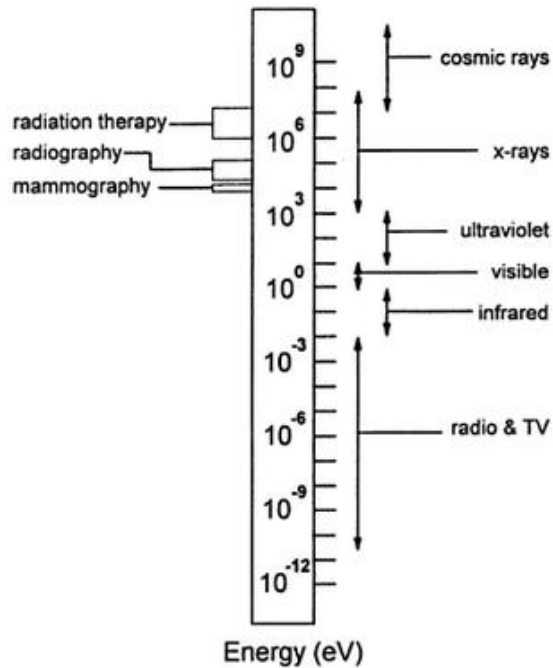


Figura 6: El espectro electromagnético es ilustrado con los niveles de Energía en eV para rayos X, mamografía, entre otros.

Fuente: [2]

Elaboración: Jacob Beutel, Harold L. Kundel, Richard L. Van Meter

En esta Figura 6, se observa algo muy importante, los rayos X son los de más alta energía y frecuencia, que los otros tipos indicados en el espectro electromagnético.

Los rayos X son de alta energía, longitud de onda corta. Los rayos cósmicos de 100GeV son los más altos conocidos.

2.1.3 Clasificación de imágenes medicas

Las imágenes médicas que se capturan actualmente se clasifican en:

Imágenes médicas analógicas, son aquellas que son almacenadas en placas y dispositivos de almacenamiento. Constituyen la primera tecnología para la captura de imágenes médicas y actualmente son todavía utilizadas.

Imágenes médicas digitales, son aquellas cuyo resultado se almacena en formato digital usando el estándar DICOM. Este tipo de mayores ventajas que los equipos de imágenes analógicos principalmente por ser digitales, es decir, por ser más fácil su transmisión, almacenamiento, posibilidad de interconsultas, telemedicina, procesamiento, entre otros.

A su vez las imágenes médicas digitales se clasifican en:

- Imágenes médicas digitales en 2D
- Imágenes médicas digitales en 3D
- Imágenes médicas digitales en 4D

Según su aplicación y tecnología de adquisición:

- Rayos X
- Radiografía
- Radiografía dental
- Mamografía
- Tomografía computarizada
- Ecografía
- Otros

2.1.3.1 Rayos X

Los rayos X y los rayos Gamma, son formas de radiación electromagnética la cual es la energía que surge en la interacción de los átomos, que tienen el potencial de liberar electrones desde los átomos. **Los rayos X y los rayos Gamma entonces son formas de radiación ionizante y esta es la característica fundamental que distingue estos rayos del resto del espectro.**

La onda electromagnética de frecuencia ν , tiene una energía proporcional a ν , con la constante de proporcionalidad dada por la constante de Planck, h :

$$E = h \times \nu \quad (2.1)$$

Dónde:

$$h=4.135 \times 10^{-15} \text{ eV-s}$$

2.1.3.1.1 Producción de Rayos X, Interacción y detección

En la siguiente Figura 7, se muestra como se producen los rayos X, como consecuencia de la liberación de electrones en el átomo.

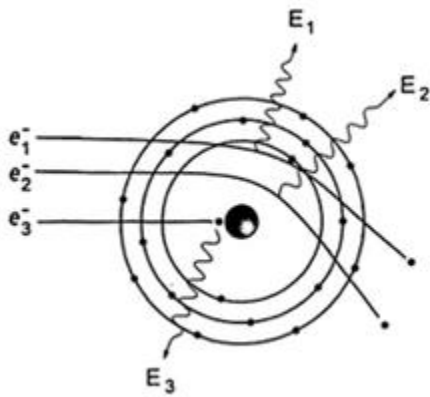


Figura 7: Producción de Bremsstrahlung.
Fuente [2].

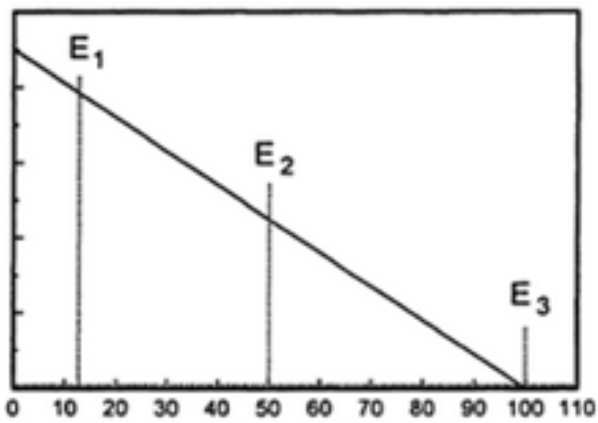


Figura 8: Grafica de Rayos X: Energía (KeV) en el Eje X, Salida relativa de Fotones en el Eje Y.
Fuente [2].

2.1.3.2 Radiografía

La radiografía es la imagen formada en un sustrato fotográfico, obtenida mediante la exposición de una parte del organismo a una fuente de emisión de rayos X. La creación de imágenes se forma a partir de la radiación reflejada que puede ser convertida en luz, haciendo uso de una pantalla fluorescente,

para lo cual debe estar soportada en una película fotográfica. En la Figura 9 se muestra una imagen radiográfica.

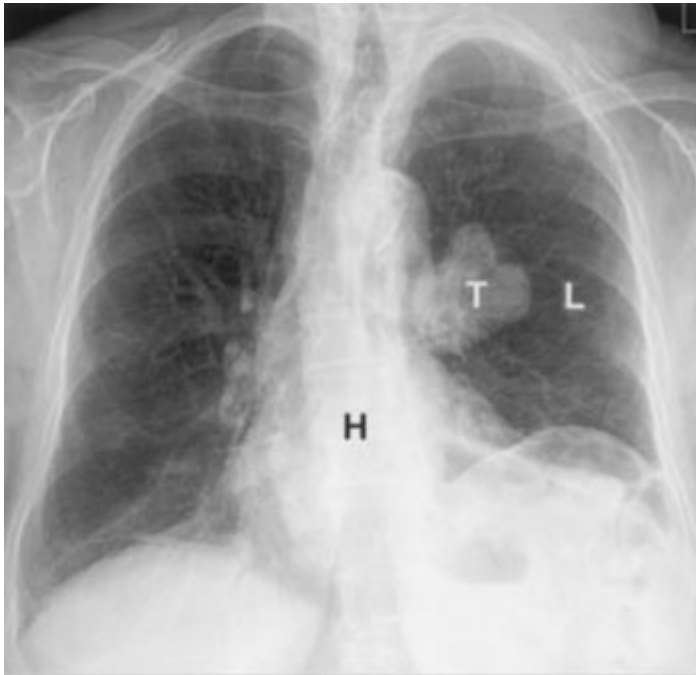


Figura 9: La radiografía del pecho muestra un contraste entre el corazón (H) y los pulmones (L), se aprecia un tumor (T).

Las radiografías utilizan rayos X en distintas proporciones e intensidades, su uso está en función de la parte del cuerpo a examinar, y determinado por otros factores indicados por el radiólogo.

Los tejidos duros, tales como el tejido óseo, requieren una fuente de energía relativamente elevada. Usualmente para generar la radiación, se utiliza un ánodo de tungsteno alimentado con una alta diferencia de potencial (50 - 150 KVp) en una maquina trifásica de alta frecuencia. Los tejidos más densos absorben los rayos X, de manera que la película fotográfica queda expuesta en la zona correspondiente a los mismos.

En cambio, en las zonas correspondientes a tejidos blancos, que no son capaces de absorber tal cantidad de rayos X, la película fotográfica queda totalmente expuesta a la radiación y por tanto adquiere el color negro. Esta configuración suele utilizarse para diagnosticar fracturas óseas, objetos ingeridos y para patologías como la osteoartritis, osteomielitis, cáncer de huesos, escoliosis, acondroplasia y problemas de crecimiento.

Los tejidos utilizan la misma configuración descrita en el apartado anterior, pero con una configuración más suave en cuanto a la energía del emisor. En esta denominación están incluidos las imágenes correspondientes a los pulmones, el corazón, los intestinos, los globos oculares, los tejidos blancos del cuello, así como los tejidos blandos del interior de los huesos para diagnosticar diferentes traumas ocultos.

2.1.3.3 Radiografía Dental

La radiografía dental usa radiación electromagnética con alta penetración para visualizar los dientes, que son relativamente densos. Suelen utilizarse máquinas monofásicas de radiación DC.

Existen dos métodos esencialmente para obtener una imagen digital radiográfica: la imagen radiográfica digitalizada, que se obtiene mediante el escaneo o captura fotográfica de la imagen de una placa radiográfica y la imagen radiográfica digital se obtiene mediante la captura digital directa de la imagen para convertir los rayos X directamente a señales electrónicas. La Figura 10, muestra un ejemplo de radiografía dental.



Figura 10: Imagen Médica de una Radiografía Dental.

2.1.3.4 Mamografía

La mamografía es la radiografía proyectiva correspondiente a las mamas. Su uso fundamental es el diagnóstico del cáncer de mama, aunque puede tener otros usos, tales como localización de tejidos sospechosos antes de realizar una biopsia. En la Figura 11, se muestra como ejemplo una mamografía.

Se utiliza normalmente un ánodo de molibdeno con una diferencia de potencial de 30KV, lo que proporciona un rango de energías que va desde los 15 hasta los 30KeV.

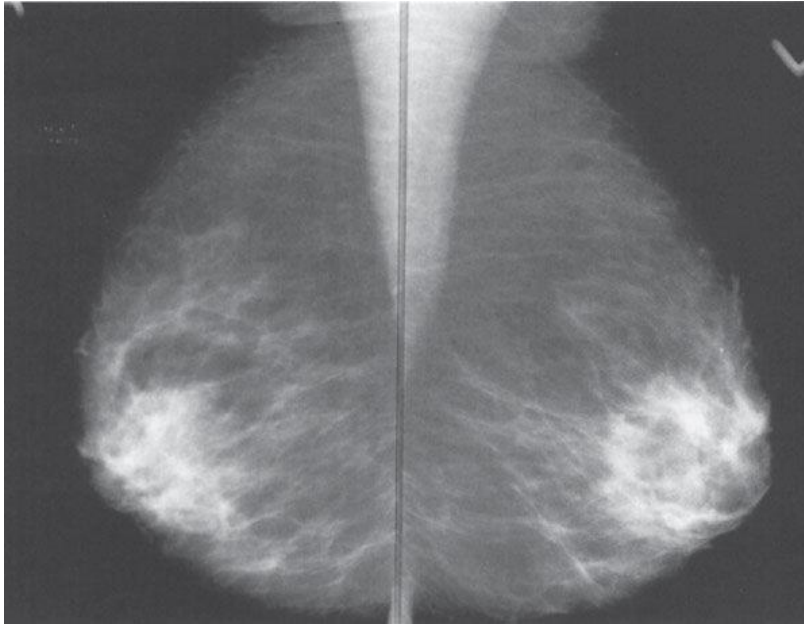


Figura 11: Vista medio lateral oblicua, este paciente muestra una cantidad moderada de tejido fibroglandular.

2.1.3.5 Tomografía Computarizada

Es una técnica topográfica axial, produce imágenes que son perpendiculares al eje longitudinal del cuerpo. Los valores de atenuación en la imagen, generados por la tomografía computarizada, reflejan: la densidad y el número atómico de los diferentes tejidos y se expresa generalmente como los coeficientes de atenuación relativos o Unidades Hounsfield (HU).

Por definición, las HU del agua y del aire son 0 y -1000 respectivamente. El HU de la gama de tejidos varía de 10 a 50, con grasa que muestran HU negativos. El hueso es por lo menos 1000HU.

La resolución de contraste de las estructuras vasculares, de los órganos y patologías, pueden mejorarse con la infusión intravenosa de medios de

contraste solubles en agua, agentes solubles en agua o suspensiones de bario, se pueden administrar para una mejor visualización del intestino.

En esta prueba se producen artefactos que provocan distorsiones en la imagen obtenida, los artefactos, se producen por el movimiento de los pacientes o por una alta densidad de cuerpos extraños, como clips quirúrgicos. La Figura 12, se confirma un tumor mediante la prueba de tomografía.

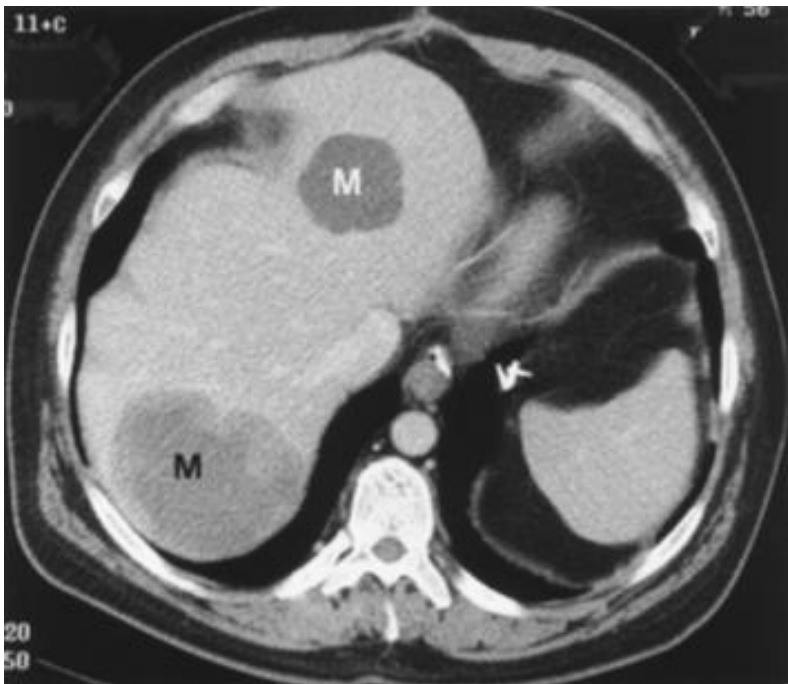


Figura 12: La imagen del epigastrio muestra dos áreas de atenuación (M) confirmando una múltiple metástasis hepática de un tumor estromal gastrointestinal.

En la actualidad se dispone de técnicas mejoradas para aplicaciones más específicas.

La Angiografía CT, para obtener la distribución vascular, con una reducción de los artefactos producidos, incrementando la resolución. La angiografía se ha vuelto importante para el análisis de las arterias: pulmonar e iliaca y sus ramificaciones, la aorta torácica y la circulación carotidea intracraneal y extra craneal.

La Colono grafía CT, para obtener imágenes en 2-D o 3-D del colon. Esta tecnología es usada principalmente en la detección y la caracterización de pólipos del colon, estas imágenes muestran su superficie y la densidad interna de las lesiones descubiertas, también la pared del intestino y estructuras fuera del colon, estructuras abdominales/pélvicas.

2.1.3.6 Ecografía

El diagnostico por ultrasonido es una técnica de obtención de imágenes no invasiva, la frecuencia de las ondas usadas son del orden de los 20 kHz. Un transductor es usado para emitir y recibir las ondas de sonido reflejadas en varios tejidos del cuerpo humano.

El transductor es colocado sobre la piel del paciente con una capa fina de gel de acoplamiento. Este gel desplaza todo el aire que de otra manera refleja prácticamente toda la viga de ultrasonido. Como el sonido viaja en el paciente en forma de frentes de onda, en su trayecto disminuye la intensidad del haz, la atenuación del haz se produce en la absorción del tejido en forma de calor, en las interfaces de cada tejido el rayo es parcialmente reflejado y transmitido. Las ondas reflejadas viajan de vuelta al transductor y son convertidos en señales eléctricas.

La amplitud de la onda que vuelve depende del grado de absorción del haz. Un tono de gris es asignado a cada amplitud, los ecos débiles se asignan a un color negro casi al final del espectro.

La profundidad del tejido que produce la reflexión puede ser calculada, a partir del tiempo de viaje del haz y de la velocidad media del sonido en el tejido humano (1540 m/s)

Las limitaciones son principalmente dependientes del operador, limitaciones adicionales incluyen la visualización de órganos medianos abdominales (páncreas), la vasculatura cuando el intestino es cubierto por un gas, así como la incapacidad de las ondas sonoras para atravesar el gas o el hueso.

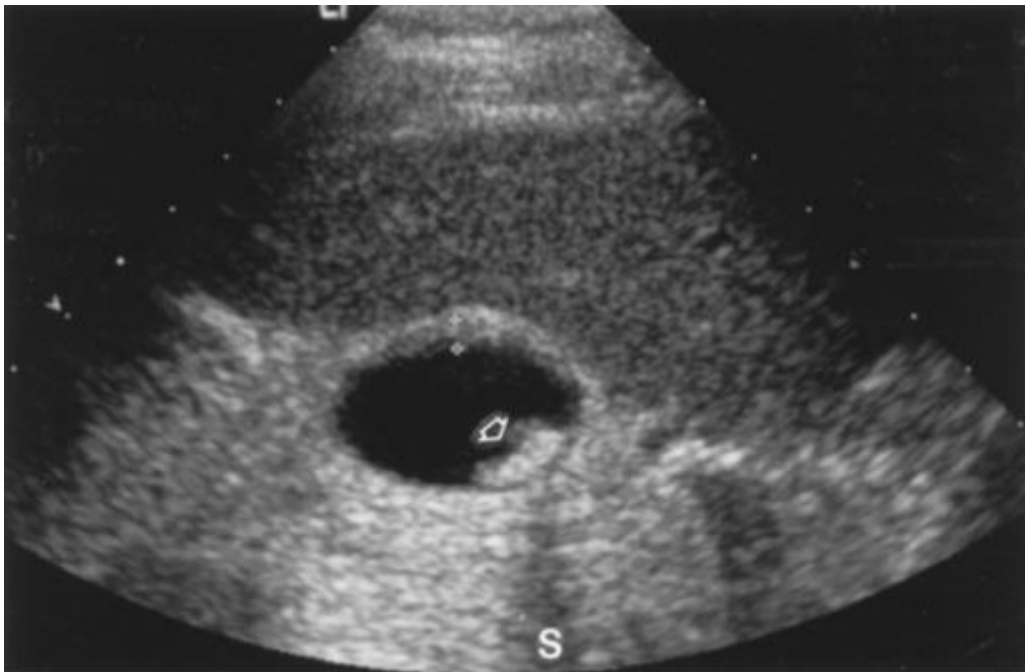


Figura 13: Una imagen de ultrasonido transversal de la vesícula muestra un cálculo biliar (la flecha) porque las ondas sonoras no pueden penetrar el cálculo biliar.

Hay muchas aplicaciones comunes de la ecografía, que incluyen imágenes del abdomen (el hígado, la vesícula, el páncreas, riñones), órganos femeninos reproductivos, el feto (estudios rutinarios fetales para la detección de anomalías), el sistema vascular (aneurismas, comunicaciones arteriales venosas, la trombosis profundamente venosa), testículos, pechos, el cerebro pediátrico (hemorragia, malformaciones congénitas), el pecho (el tamaño y la posición de colecciones pleurales fluidas) e intervenciones dirigidas por ultrasonido

En general las imágenes médicas se obtienen usando diversos métodos, sus dimensiones también varían de un tipo de prueba a otra. La ventaja de las pruebas mencionadas es que no son invasivas, es decir no requieren una perforación de la piel para observar la parte interna del cuerpo humano.

En la siguiente tabla se muestran las modalidades empleadas para obtener una imagen médica, su objeto de estudio y algunas propiedades más.

2.1.3.7 Resumen de los tipos de Imágenes médicas según la energía transmitida

En la Tabla 1, se muestran las modalidades empleadas para obtener una imagen médica, donde podemos apreciar que existen varias tecnologías para obtenerlas desde rayos X, Sonido, RF hasta luz visible. Asimismo en cuanto a las imágenes obtenidas observamos que la composición de los pixeles es diferente para cada modalidad. Todas estos tipos de imágenes son almacenadas en archivos DICOM.

Modalidad	Objeto de estudio	Energía Transmitida	Propiedad Medible	Energía Recibida	Método de Detección	Procesamiento	Visualización	Tamaño N x M (pixeles)	Niveles de gris	Bits por pixel
Radiología Convencional	Tejido duro y blando	Rayos X	Absorción	Rayos X	Placas	Químico	Placas	Mod. Analógica		
Tomografía Computada	Tejido duro y blando	Rayos X	Absorción	Rayos X	Detectores	Transformada de Radón	Monitor y Placas	512 x 512	4096	12
Ultrasonido	Tejido Blando	Sonido	Reflexión	Sonido	Cristal Piezoeléctrico	Barrido en espacio y tiempo.	Video, Monitor o Impresión.	512 x 512	256	8
Doppler	Flujo Sanguíneo	Sonido	Dispersión	Sonido (variación frecuencial)	Cristal Piezoeléctrico	Barrido en espacio, tiempo y frecuencia.	Video, Monitor o Impresión.	512 x 512	256	8
Resonancia Magnética Nuclear	Tejido Blando	RF	Respuesta a la señal de resonancia magnética.	RF	Antenas	Transformada de Fourier	Monitor y Placas	256 x 256	4096	12
Radiología Digital Computada	Tejido duro y blando	Rayos X	Absorción	Rayos X	Detectores	Conversión A/D	Monitor y Placas	4000x2000	4096	12
Endoscopia	Organos internos	Luz visible	Reflexión	Luz Visible	Cámara	Digitalización	Monitor e impresión	512 x 512	2 ²⁴ colores	24
Microscopia	Muestras	Luz visible	Transmisión	Luz Visible	Cámara	Digitalización	Monitor e impresión	512 x 512	2 ²⁴ colores	24
Medicina Nuclear	Funcionalidad Orgánica		Número de desintegraciones	γ , β	Escintilación CCD Placas	Análisis de Eventos.	Monitor Impresión en placas y otros	128 x 128	65536	16

Tabla 1: Características de adquisición de cada sistema de imágenes

2.1.4 Mejor Calidad de imagen pero más radiación

En la siguiente Figura 14 tomada de [2], se observa una prueba hecha que consiste en obtener imágenes y medir la radiación utilizada, como se observa en (a) – (d) pasa de borrosa a mas nitida, y de 0.16 uR, 1.6 uR, 16uR, y 24uR respectivamente, aquí se puede observar claramente que para obtener una imagen medica mas nitida es necesario mayores dosis de radiación.

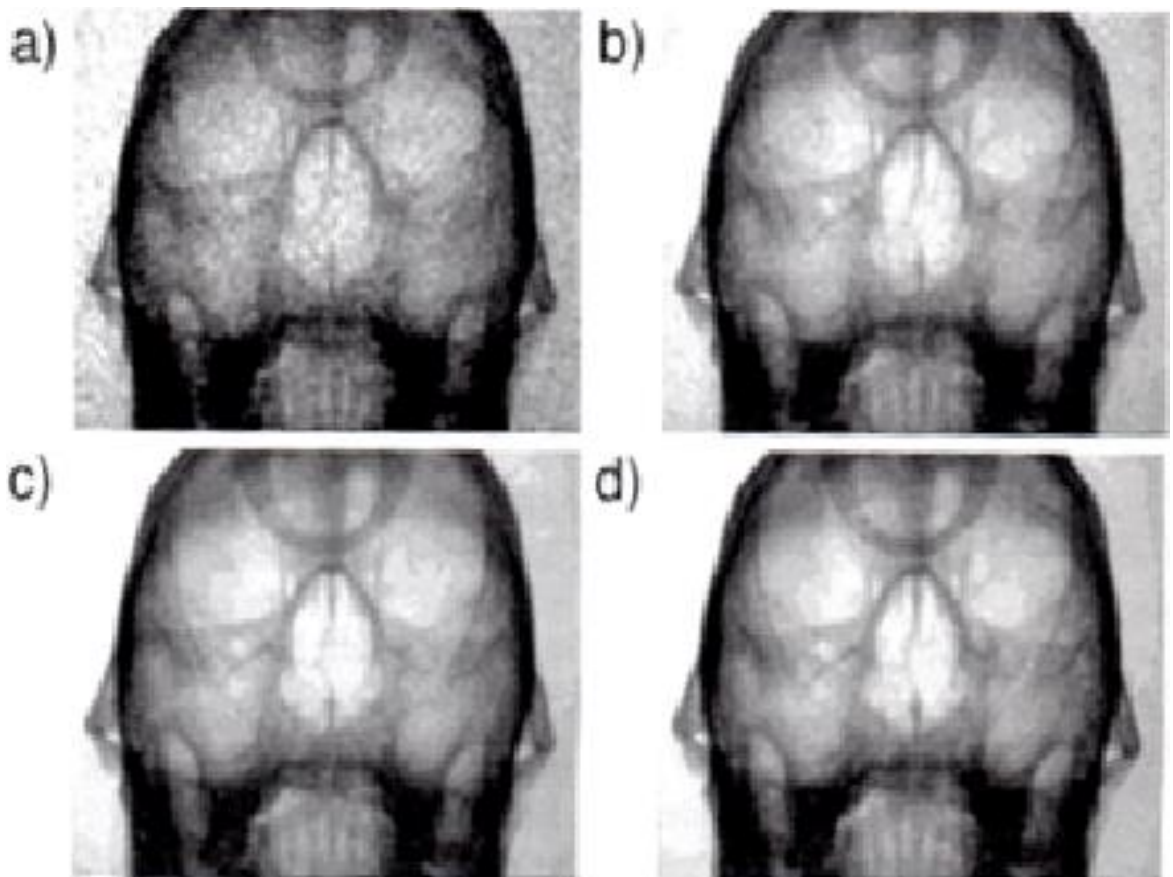


Figura 14: Mayor resolución de la imagen medica con el uso de mayor radiación. El promedio de la exposición detectada de los rayos X por imagen es: a) 0.16 uR, b) 1.6 uR, c) 16uR, y 24uR.

Fuente: [2]

	Detector de Exposición	Radiación (Quanta) por mm ²
a)	0.16 uR	45
b)	1.6 uR	450
c)	16 uR	4500
d)	24 uR	6720

Tabla 2: Resumen del detector de exposición y número de quanta per mm² usado para crear imágenes mostradas en la Figura 14.

Fuente: [2]

2.1.5 Como la radiación afecta el ADN

Según [19], cuando los rayos X interactúan directamente con el ADN, tres tipos claves de daños ocurren:

1. Daño en la base nucleótido (Ver Figura 15)
2. Rotura simple de una hebra de ADN,
3. Rotura doble de una hebra de ADN (Ver Figura 16)

Una dosis de 1mSv de Low-LET radiación en promedio se espera lo siguiente: 2 bases nucleótido dañadas, 1 rotura simple de hebra y 0.04 roturas dobles de hebra.

2.1.5.1 Consecuencias de un ADN dañado

Según [19], el ADN dañado causado por los rayos X puede subsecuentemente conllevar a una variedad de mutaciones y aberraciones en cromosomas, resultando en vías que pueden causar o pueden contribuir a la patogénesis del cáncer.

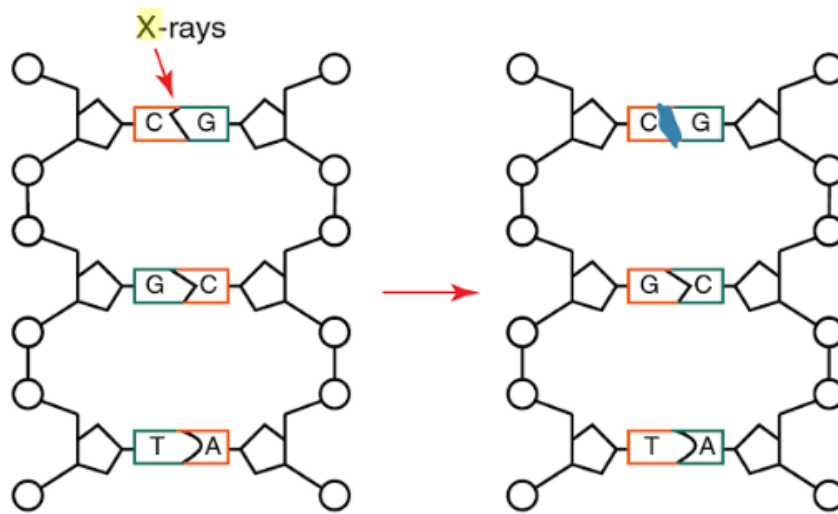
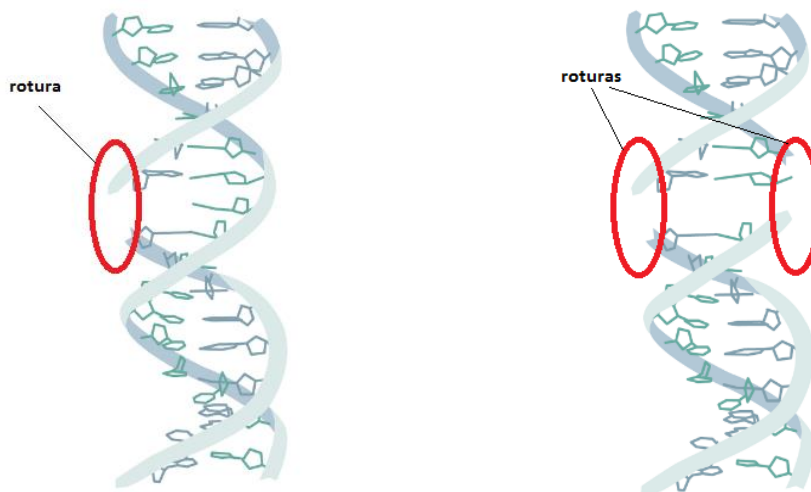


Figura 15: Daño en la base del nucleótido, causado por los rayos X

Fuente: Radiation Protection In Diagnostic X-Ray Imaging [19]

Elaboración: Euclid Seeram, Patrick C Brennan



(a) rotura simple de hebra

(b): rotura doble de hebra

Figura 16: Rotura de hebra de ADN como consecuencia de los Rayos X

Fuente: Radiation Protection In Diagnostic X-Ray Imaging [19]

Elaboración: Euclid Seeram, Patrick C Brennan

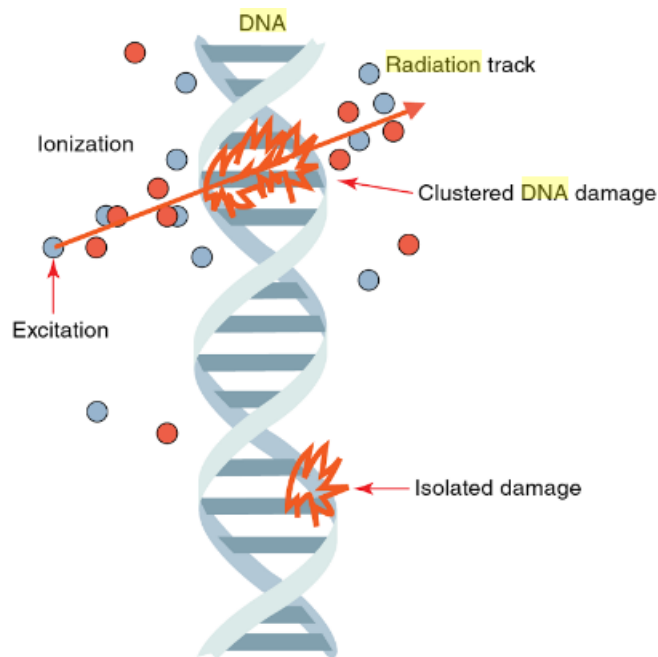


Figura 17: Clústeres de daño e incidentes aislados seguidos a una interacción de radiación al ADN

Fuente: Radiation Protection In Diagnostic X-Ray Imaging [19]

Elaboración: Euclid Seeram, Patrick C Brennan

2.2 Fundamentos de Imágenes digitales

2.2.1 Concepto de una Imagen Digital

Es la imagen capturada de un objeto o escena del mundo real desde un CCD, y es representada por unos y ceros.

En la Figura 19, vemos como una imagen digital, si le hacemos un zoom (b), se puede visualizar el valor de cada pixel, en este caso:

8 bits: Tiene valores desde 0 (negro) hasta 2^8-1 (blanco),

En el caso de 16 bits: Tiene valores desde 0 (negro) hasta $2^{16}-1$ (blanco)

Donde cada valor es un color de gris distinto, lo cual proporciona mayor precisión y resolución de la imagen.

El código en Matlab utilizado es:

2.2.1.1 Código en Matlab para explorar el contenido de una imagen por pixel

```
%*****  
% Explorar una imagen como matriz de pixeles, y cada pixel  
con un valor o varios, según sea el tipo de imagen  
% Desarrollado por: Enzo Camargo Fernandez Baca, 2015  
%*****  
  
myFolder = 'D:\sr';  
  
% f1 = 'orquidea.JPG';  
% f2 = 'orquidea_BN.JPG';  
f1 = 'orquidea_machupicchu.JPG';  
f2 = 'orquidea_machupicchu_BN.jpg';  
  
% Cargando imagen real : HR  
f = imread(fullfile(myFolder,f1));  
imtool(f)  
  
% Pasar a BN  
fbn = rgb2gray(f);  
imwrite(fbn,fullfile(myFolder,f2));  
imtool(fbn)
```

2.2.1.2 Imagen en Escala de Grises

Utilizando el script de Matlab, indicado obtenemos la Figura 19, la cual es una orquídea peruana que crece en Machupicchu, la imagen original en colores se ha pasado a escala de grises para presentar un análisis básico inicial, con las siguientes precisiones:

1. Se observa claramente que la imagen está formada por una matriz de pixeles.
2. Cada pixel tiene un valor, en este caso de 0 a 255 (8 bits = 1 byte, $2^8 - 1 = 255$), se tiene como valor 0 = negro, y valor 255 = blanco. En el caso que se tenga 16 bits por ejemplo, tendríamos una mayor variedad de grises
3. El tamaño, está determinado por ancho x largo x tamaño de cada pixel, entonces tenemos:

Tamaño de imagen digital = $285 \times 510 \times 1 \text{ byte} = 145,350 \text{ bytes}$

Lo cual coincide con lo calculado por Matlab, cuando se muestra las propiedades de la imagen:

```
>> whos fbn
```

Name	Size	Bytes	Class	Attributes
fbn	285x510	145350	uint8	

4. Valores de ejemplo de cada pixel, si observamos la Figura 19, observamos en el cuadro de la derecha una "L", los cuales tienen los siguientes valores:

(279, 79) 170

(279, 80) 207

(280, 80) 216

Por lo tanto, el sentido de la posición es el indicado en la Figura 24:

X: izquierda a derecha

Y: de arriba hacia abajo

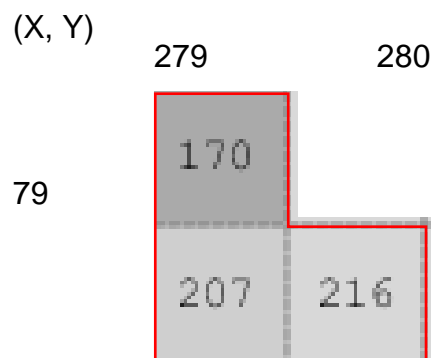


Figura 18: Posición de los Píxeles en una imagen en escala de grises

Fuente: Enzo Camargo Fernández – Baca

Elaboración: Enzo Camargo Fernández - Baca

5. Rango de valores, como se observa en la Tabla 3, los valores mínimo y máximo de los píxeles son 5, 225 respectivamente, asimismo se tienen valores intermedios, los cuales se muestran el Histograma indicado en la Figura 20.

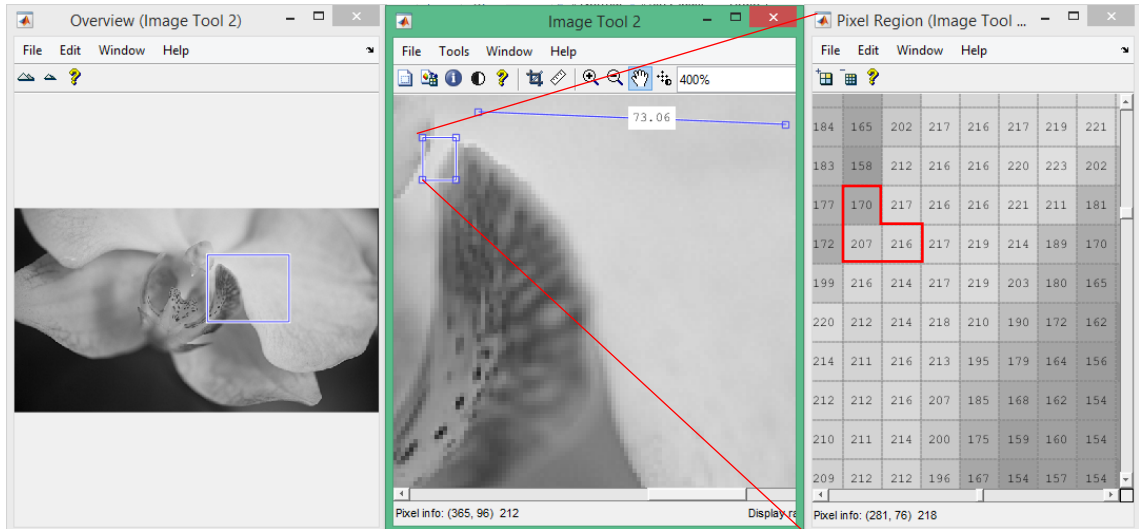


Figura 19: Representación de una Orquídea peruana como imagen digital en una matriz de pixeles.

Fuente: Enzo Camargo Fernandez – Baca

Elaboración: Enzo Camargo Fernandez – Baca

Información de la imagen:

Atributo	Valor
Ancho (columnas)	510
Alto (Filas)	285
Clase	uint8
Tipo imagen	intensity
Intensidad mínima	5
Intensidad máxima	225

Tabla 3: Valores de atributos para imagen en escala de grises de la Orquídea peruana

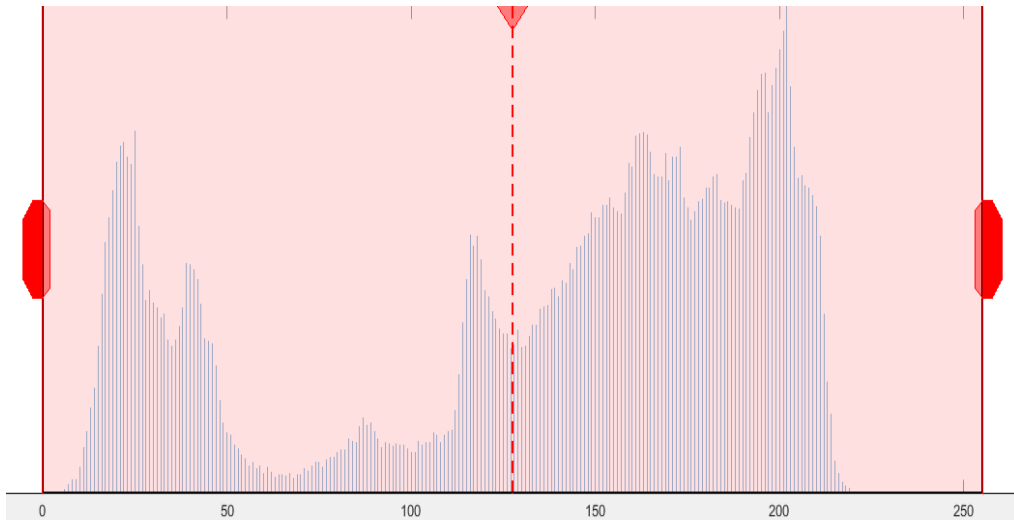


Figura 20: Histograma de la imagen en escala de grises de la Orquídea peruana

Fuente: Enzo Camargo Fernández - Baca

Elaboración: Enzo Camargo Fernández – Baca

2.2.1.3 Imagen en Colores RGB



Figura 21: Orquídea peruana en peligro de extinción que crece en Machupicchu

Fuente: Enzo Camargo Fernández - Baca

Elaboración: Enzo Camargo Fernández - Baca

Utilizando el mismo script 2.2.1.1, indicado previamente a la imagen a color RGB se aprecia lo siguiente:

1. Se observa claramente que la imagen está formada por píxeles, en este caso la diferencia con la imagen anterior es que tiene para cada píxel, 3 valores en lugar de uno como la imagen de la Figura 19 en escala de grises.
2. Cada píxel tiene 3 valores (R, G, B), en este caso 1 byte por cada canal, como son 3 canales R, G y B, entonces tenemos 3 bytes por cada píxel. Cada Canal tiene valores desde 0 a 255 (8 bits = 1 byte).
3. El tamaño, está determinado por ancho x largo x tamaño de cada píxel, entonces tenemos:

Tamaño de imagen digital = $285 \times 510 \times 3 \text{ byte} = 436,050 \text{ bytes}$

Lo cual coincide con lo calculado por Matlab, cuando se muestra las propiedades de la imagen a color RGB:

```
>> whos f
  Name      Size      Bytes  Class  Attributes
  f         285x510x3    436050  uint8
```

4. Valores de ejemplo de cada pixel, si observamos la Figura 23, observamos en el cuadro de la derecha una "L", los cuales tienen los siguientes valores RGB:

(279, 79) [177 166 174]
(279, 80) [211 205 209]
(280, 80) [219 215 214]

El sentido de los ejes X e Y, es igual a una imagen en escala de grises, y es en general para todo tipo de imágenes.

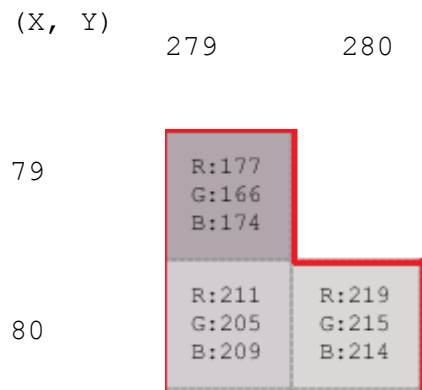


Figura 22: Posición de los Píxeles en una imagen a colores RGB

Fuente: Enzo Camargo Fernández – Baca

Elaboración: Enzo Camargo Fernández - Baca

5. Rango de valores, es para cada canal de 0 a 255, para aplicar el histograma se debería hacer a cada canal R,G y B respectivamente.

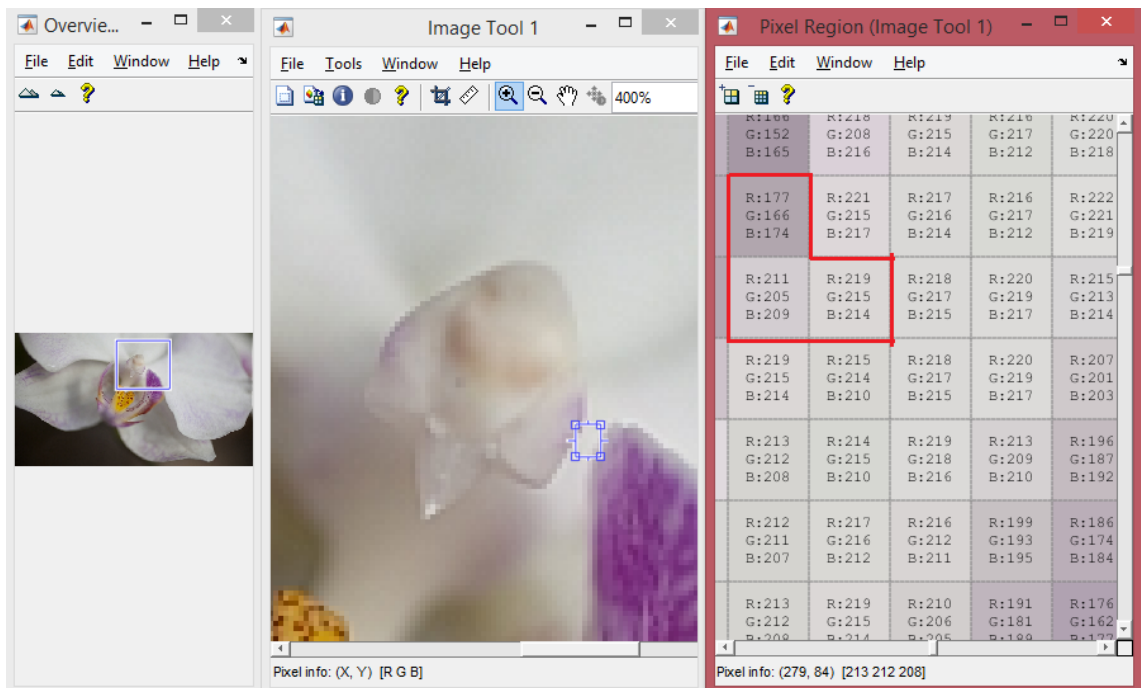


Figura 23: Imagen de la Orquídea peruana a color representada por una matriz de pixeles y cada pixel con valores R, G Y B

Fuente: Enzo Camargo Fernandez – Baca

Los atributos de la imagen mostrada son especificados en la siguiente Tabla:

Atributo	Valor
Ancho (columnas)	510
Alto (Filas)	285
Clase	uint8
Tipo imagen	truecolor

Tabla 4: Valores de atributos para imagen en colores RGB de la Orquídea peruana

2.2.1.4 Tipos de imagen

Los siguientes son los tipos de imágenes, soportadas por MATLAB:

- Imágenes en escala de grises
- Imágenes binarias
- Imágenes indexadas
- Imágenes RGB

2.2.2 Representación Matemática de una Imagen Digital

Entonces, según lo observado en los ejemplos previos, una imagen digital se representa matemáticamente como una matriz ordenada de valores. Formalmente una imagen es una función I que asocia las coordenadas enteras $N \times M$ a un rango de valores (píxeles), esto es:

$$I(u, v) \in P ; \quad u, v \in N \quad (2.2)$$

El valor de cada elemento (u, v) de la matriz indica una posición, tiene asignado un sistema de coordenadas específico, usado por las aplicaciones que editan imágenes. Considerar que $(u, v) = (X, Y)$.

En la definición del sistema de coordenadas tal como se muestra en la figura 3, se ubica al origen en la esquina superior izquierda, la coordenada v se incrementa desde arriba hacia abajo. La coordenada u se incrementa de

izquierda a derecha. La numeración de filas y columnas inician a partir de cero.

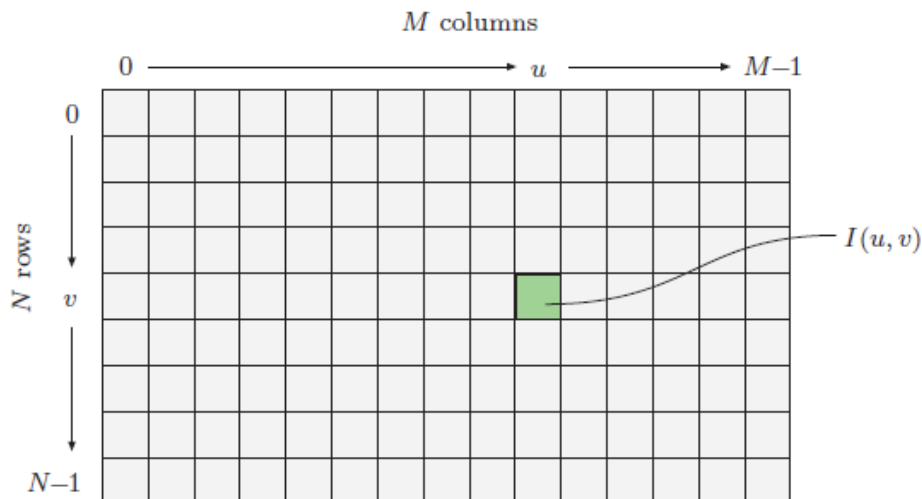


Figura 24: Sistema de coordenadas X, Y en el procesamiento digital de imágenes, donde $(X=0, Y=0)$ se ubica en la parte superior izquierda

La información en un píxel depende del tipo de dato usado para representarlo, estos valores se representan en binario, palabras de longitud k tal que un píxel se puede representar por uno de 2^k diferentes valores. El valor de k está definido como profundidad de bits de la imagen. El nivel de bit exacto de un píxel depende del tipo de la imagen, por ejemplo: binario, escala de grises o color RGB.

2.2.3 Que es la Calidad de una imagen digital

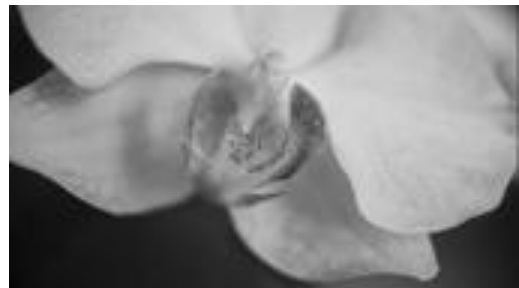
La calidad de una imagen puede ser medida mediante criterios objetivos como el PSNR o mediante criterios subjetivos (ojo humano).

El PSNR es la medida o indicador objetivo en la que una imagen obtenida, como la imagen LR o SR, se aproxima a una de referencia, como la imagen HR, utilizado para pruebas sintéticas, no siendo posible su utilización para aplicaciones reales.

El PSNR y MSE, son indicadores objetivos ampliamente utilizados en el campo de procesamiento de imágenes.



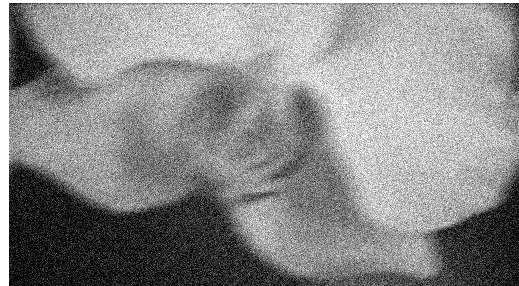
(a) Imagen Original (HR)



(b) Imagen (LR) aplicando sub muestreo a imagen original, disminuido en 3 veces
PSNR = 33.4234



(c) Si a la imagen (b) se le adiciona el efecto del PSF o desenfoque optico, obtenemos
PSNR = 27.2806



(d) Imagen (c) se le adiciona ruido, se obtiene
PSNR = 19.3994

Figura 25: Adicion de efectos simulados de submuestreo, PSF y ruido a la imagen de la Orquidea peruana

Fuente: Toma fotográfica

Elaboración: Enzo Camargo Fernández – Baca

Como se observa en la Figura 25, visualmente la imagen (b) tiene mayor resolución que la imagen (c), y que la imagen (d), comprobado por la medición objetiva del PSNR calculado obteniéndose $PSNR(b) > PSNR(c) > PSNR(d)$.

Para ilustrar este concepto:

Se sub muestrea la imagen Figura 25 (a) y se obtiene (b), y

(c) = Se aplica desenfoque a (b)

(d) = Se aplica ruido a (c)

El MSE es el error cuadrático medio entre la imagen original y la imagen reconstruida. Mientras tenga un valor menor es mejor, significa que la imagen reconstruida se acerca a la imagen original.

Según [13] El PSNR y MSE son los más comunes parámetros usados para medir la calidad de una imagen reconstruida después de compresión, la FDA demanda medir y reportar el MSE para cada factor de compresión utilizada en dispositivos médicos usando métodos de compresión sin pérdida.

PSNR y MSE, son aplicables a los tipos de imágenes indicados en 2.2.1.4, en caso que tenga varios canales se aplica canal por canal.

2.2.3.1 Código Matlab para creación de imágenes de prueba a partir de imágenes HR

```
% *****  
% Elaborado por: Enzo Camargo Fernandez Baca  
% *****  
  
myFolder = 'D:\sr';  
% f1 = 'Biblioteca UNI - HR.JPG';  
f1 = 'orquidea_machupicchu_BN.JPG';  
r=3;  
f2 = 'orquidea_machupicchu_BN_LR.jpg';  
f3 = 'orquidea_machupicchu_BN_blurred.jpg';  
f4 = 'orquidea_machupicchu_BN_blurred_noisy.jpg';  
  
% Leer  
ref = imread(fullfile(myFolder,f1));  
[XR,YR, ZR]=size(ref);  
  
%  
*****  
*  
% Paso 1: Submuestreo de imagen original  
%  
*****  
*  
LR = imresize(ref,1/r);  
% imshow (y)  
  
LR2 = imresize(LR, [XR YR], 'bicubic');  
imwrite(LR2,fullfile(myFolder,f2));
```

```

peaksnr1 = psnr(LR2,ref);
fprintf('\n The Peak-SNR -1 value is %0.4f', peaksnr1);

*****
*
% Paso 2: Adicionar efecto PSF: Simulate a motion blur.
%
*****
*
LEN = 21;
THETA = 11;
PSF = fspecial('motion', LEN, THETA);
blurred = imfilter(LR2, PSF, 'conv', 'circular');
figure, imshow(blurred)
title('Filtrando con PSF conocido')
LR3 = imresize(blurred, [XR YR], 'bicubic');

imwrite(LR3,fullfile(myFolder,f3));
peaksnr2= psnr(LR3,ref);
fprintf('\n The Peak-SNR -2 value is %0.4f', peaksnr2);

%
*****
*
% Paso 3: Simulate additive noise
%
*****
*
noise_mean = 0; % Media del Ruido
noise_var = 0.01;
blurred_noisy = imnoise(blurred, 'gaussian', ...

```

```
                                noise_mean, noise_var);  
    figure, imshow(blurred_noisy)  
%     t=strcat(iCAD,' _ ','Simulate Blur and Noise  
Gaussian');  
%     title(t);  
    LR4 = imresize(blurred_noisy, [XR YR], 'bicubic');  
    imwrite(blurred_noisy,fullfile(myFolder,f4));  
    peaksnr2= psnr(LR4,ref);  
    fprintf('\n The Peak-SNR -3 value is %0.4f', peaksnr2);
```

2.2.3.2 PSNR: Relación Señal a Ruido (PSNR, Peak Signal to Noise Ratio):

Si se hace una verificación visual de las imágenes de la Figura 25, observamos que tenemos desde una imagen original hasta una imagen con ruido, en la que cada vez la vamos degradando usando el código en Matlab indicado en 2.2.3.1, pero en todos los casos no es posible analizar y comparar una imagen con otra visualmente, por ello se requiere de un indicador objetivo como el PSNR, que es utilizado ampliamente en el campo del procesamiento de imágenes.

Según [13] y [15], el PSNR (La relación señal a ruido) es un indicador cuantitativo objetivo para medir la calidad de imágenes y ampliamente aplicado en imágenes médicas y evaluar su performance, definiéndose como:

$$SNR(dB) = 10 \log \left(\frac{P_{señal}}{P_{ruido}} \right) \quad (2.3)$$

Una relación señal a ruido (SNR) elevada implica que la imagen no es ruidosa. Solo en el caso de una MRI, el ruido es distribuido uniformemente a través de la imagen. Una variación del SNR y que sirve como medida cuantitativa de la calidad de la reconstrucción en el ámbito de la compresión de imágenes, es la Relación Señal a Ruido Pico (PSNR), su definición es:

$$PSNR(dB) = 10 \log \left(\frac{MAX_I^2}{MSE} \right) \quad (2.4)$$

2.2.3.3 MSE (Mean Squared Error)

Donde, para dos imágenes I, K de ancho M y alto N de un canal y con una profundidad de bits k, se tiene:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \|I(i,j) - K(i,j)\|^2 \quad (2.5)$$

Para imágenes a color en 3 canales RGB, el MSE es la media aritmética del MSE en cada canal.

En el entorno médico, los tres parámetros analizados (M, N, k) son optimizados durante la adquisición para obtener una imagen médica de alta calidad. Considerar que a mayor resolución espacial y/o de densidad, mayores serán la capacidad de memoria necesaria, tiempo de procesamiento y transmisión de la imagen.

2.2.4 Fenómenos que afectan la calidad de una imagen

Los fenómenos que afectan la calidad de una imagen, están indicados en la Figura 3 son:

- Traslación y/o rotación
- Ruido
 - atmosférico
 - interno(propio del equipo)
- Desenfoco
 - óptico
 - movimiento
 - sensores PSF, etc.
- Muestreo (sampling)

- continuo a discreto (la imagen real tiene una señal continua, al pasar a digital, se pasan a 1s y 0s)
- Decimación (decimación)
 - Disminuir el muestreo (down sampling)

2.2.4.1 PSF

El PSF (Point Spread Function), es el desenfoque optimo que se define como la imagen obtenida cuando por un orificio se hace pasar un punto de luz, esto nos ayuda a identificar como se difumina la imagen o se representa la imagen ante una excitacion puntual conocida. El PSF depende del equipo, por ejemplo si tomamos una fotografia con una camara fotografica profesional y otra con una camara simple del mismo fabricante, tendremos PSFs distintos. El PSF esta ligado directamente al Hardware o al equipamiento que adquiere las imágenes. Veamos los siguientes ejemplos:

Hasta el momento, todos los factores que hemos mencionado en su mayoría son inherentes al equipo que adquiere las imágenes, es por ello, que se puede tener mejores imágenes en equipos de mayor performance, porque estos equipos tienen menor desenfoque (entre otros, mejor PSF), menor distorsion(minimizacion de ruido), mejor tasa de muestreo (expresado por los megapixels) y por lo tanto mayor resolucion de una imagen digital.



Figura 26 : Fenómeno de desenfoque que afecta a la imagen. Izquierda: desenfoque de movimiento. Derecha: desenfoque óptico

2.3 Fundamentos de Súper Resolución

Súper Resolución es el proceso de obtener imágenes de alta resolución a partir de una o varias imágenes de baja resolución, mediante la minimización de los fenómenos que afectan la calidad de la imagen indicados en 2.2.4, se reconstruyen la imagen (LR) de la forma más aproximada posible a HR obteniendo la imagen en Súper Resolución (SR).

2.3.1 Clasificación de Súper Resolución

La siguiente clasificación es basada en la revisión bibliográfica realizada, cuyas referencias son indicadas en cada método o técnica descrita:

1. La cantidad de imágenes requeridas como entrada al proceso de Súper Resolución (1 o varias);
2. Dominio de las imágenes (dominio de las frecuencias o dominio del espacio); y
3. Campo de aplicación.

Tipo de Clasificación	Clasificación	Ejemplo
Por la cantidad de imágenes	Una única Imagen de entrada de baja resolución (LR= {1})	Método de Kernel Método de Kernel adaptativo entre otros
	Múltiples imágenes de entrada de baja resolución (LR = {1,2,..., p}); considerando que cada un video está compuesto por frames, este método puede ser aplicado también a videos.	Método IBP Súper Resolución digital y el teorema del muestreo generalizado
Dominio de las imágenes	Dominio de la frecuencia	IBP (Iterative Back Projection)
	Dominio del espacio	IKS (Iterative Kernel Steering)
Campo de aplicación	Imágenes Médicas Militar Seguridad Robótica Imágenes aeroespaciales y satelitales	Rayos X, Mamografía, CT, MRI, etc. Video vigilancia,

Tabla 5: Clasificación de los Métodos de Súper Resolución

Elaboración: Enzo Camargo Fernández Baca

2.3.1.1 Por la cantidad de imágenes requeridas

Esta clasificación de métodos se basa en el número de imágenes de baja resolución de entrada:

- Una única Imagen de entrada de baja resolución ($LR = \{1\}$)
- Múltiples imágenes de entrada de baja resolución ($LR = \{1, 2, \dots, p\}$); considerando que cada un video está compuesto por frames, este método puede ser aplicado también a videos.

2.3.1.1.1 Súper Resolución con una sola imagen de entrada

Estas técnicas tienen como entrada una sola imagen, la cual es única para realizar todo el proceso de Súper Resolución. La diferencia de esta clasificación con el basado en varias imágenes, es que no precisa de un conjunto para entrenamiento ni para calcular la interpolación, además no realizar la registración.

La registración es el proceso por el cual varias imágenes se alinean a una misma coordenada x , y con la finalidad de luego ser procesadas, y en este caso al ser solo una imagen, no aplica este procedimiento.

A continuación brindamos una recopilación de artículos relacionados:

2.3.1.1.1.1 Súper Resolución desde una simple imagen (Súper Resolución from a single image) según Shai Bagon, Michael Irani [1]

En este artículo propone un marco de desarrollo unificado que combina a dos familias de métodos, la Súper Resolución en sistemas multimagen y Súper Resolución basada en muestras. El objetivo es combinar estas dos familias

de métodos con el fin de brindar un nuevo enfoque que pueda ser utilizado para aplicar Súper Resolución a partir de una sola imagen. El enfoque ha sido basado en la observación de la existencia de secciones en una imagen natural que tienden a repetirse o volverse redundantes en algunas escenas. Este enfoque intenta recuperar (reconstruir) en cada pixel su mejor resolución posible basada en la existencia de secciones redundantes y cercanas al cada pixel.

2.3.1.1.2 Súper Resolución desde señales totalmente perturbadas y sin registración

(Titulo original en inglés: Súper-Resolución from Unregistered and Totally Aliased Signals Using Subspace Methods) [21]. En muchas aplicaciones la frecuencia de muestreo está limitada por las características físicas de los componentes de densidad de pixeles, la muestreo de señal analógica a digital, etc.; un filtro de paso bajo suele aplicarse como un paso previo para evitar el Aliasing.

2.3.1.1.2 Súper Resolución con múltiples imágenes de entrada

Estas técnicas tienden a ser las más usadas, la idea es utilizar un conjunto de imágenes que sirvan como datos básicos en baja resolución necesarios para obtener la imagen en alta resolución, usualmente aplicado a secuencia de imágenes o video. Algunos artículos:

2.3.1.1.2.1 Súper Resolución digital y el teorema del muestreo generalizado

(Titulo original en inglés: Digital súper resolution and the generalized sampling theorem) [18]. En este artículo el problema de Súper Resolución es analizado

desde un punto de vista de teoremas de muestreo generalizado. Aquí es mostrado teóricamente y por simulación de computadora que los cambios de sub píxeles regulares tienden a maximizar la robustez y minimizar el error encontrado en el proceso de reconstrucción para la imagen de alta resolución.

2.3.1.1.2.2 Súper Resolución desde una secuencia de imágenes publicado por N.K. Bose

(Titulo original en inglés: N. K. Bose. Súper resolution from image sequence) [4] Debido al costo de hardware, tamaño y limitaciones en la complejidad de fabricación, sistemas de imágenes CCDs como los detectores de rayos, o cámaras digitales proveen imágenes de baja resolución con degradaciones en su escena capturada.

2.3.1.1.2.3 Reconstrucción de Súper Resolución de secuencia continua de imágenes por Michael Elad and Arie Feuer [8]

(Titulo original en inglés: Súper Resolución Reconstruction of Continuous Image Sequences) Los algoritmos de reconstrucción en base a Súper Resolución utilizan una fusión de algunas imágenes de baja calidad de la misma escena dentro de una simple imagen de calidad mejorada. Este artículo se define una tarea dinámica de Súper Resolución: la restauración de una imagen borrada y una secuencia de imágenes con ruido. Primeramente se modela este problema a través de ecuaciones estado -espaciales, mostrando que este problema puede ser visto como una secuencia de problemas de Súper Resolución estáticas. Dos algoritmos de reconstrucción eficientes son propuestos, ambos siendo aproximaciones adaptativas de los filtros de Kalman.

2.3.1.1.2.4 Técnicas de Súper Resolución para la mejora de secuencias de video comprimido. [10]

Este artículo muestra una novedosa técnica basada en la aplicación de una etapa de pre procesamiento que permite aplicar el algoritmo de Súper Resolución para incrementar la calidad de la imagen solo en aquellas regiones donde la calidad puede efectivamente ser mejorada, disminuyendo de esta forma el coste computacional total.

Asimismo, también se presenta un método que persigue disminuir el tamaño del video comprimido mientras que mantiene su calidad visual.

2.3.1.1.2.5 Súper Resolución aplicada a video (secuencia de imágenes o frames)

Las técnicas de Súper Resolución también son aplicables a video, considerándose el video como un conjunto de imágenes en una secuencia.

Uno de sus principales retos es lograr una buena calidad de video en tiempo real con un bajo costo computacional, a continuación brindamos una serie de artículos relevantes de Súper Resolución aplicado a video.

2.3.1.1.2.6 Súper Resolución y eliminación de ruido en video

(Título original en inglés: Noisy video Súper Resolución) Los videos de baja calidad a menudo no sólo tienen resolución limitada, sino también poseen ruido. El muestreo directo sin considerar ruido podría deteriorar su calidad visual.

Este artículo trata este problema mediante un marco de trabajo unificado que logra limpiar el ruido y aumentar el muestreo de manera simultánea; se formula el problema de la Súper Resolución en videos como un problema de optimización, encaminado a maximizar la calidad visual de los resultados. Los experimentos en una variedad de videos con ruido demuestran la efectividad del algoritmo presentado.

2.3.1.1.2.7 Súper Resolución eficiente y sus aplicaciones en mosaicos

(Titulo original en inglés: Efficient súper-resolution and applications to mosaics) Técnicas de mosaico y Súper Resolución son dos maneras de combinar información de múltiples cuadros en secuencias de video.

Las técnicas de mosaico muestran la información de múltiples cuadros en una simple imagen panorámica. Súper Resolución utiliza regiones que aparecen en múltiples frames para mejorar la resolución y reducir el ruido. El aporte de este trabajo es la construcción de un mosaico de alta resolución a partir de secuencias de video de manera eficiente. La simple combinación de dos métodos es una dificultad, desde que el alineamiento usado en las técnicas de mosaico pueden no ser lo suficientemente exactas para la Súper Resolución. Otro tema es la eficiencia de los algoritmos de Súper Resolución, los cuales requieren grandes costos computacionales, especialmente cuando son aplicados a largas imágenes tales como mosaicos panorámicos.

2.3.1.2 Por el Dominio de las imágenes

Por el Dominio de las imágenes se clasifican en [22] [17]:

- Dominio de la frecuencia
- Dominio del espacio

2.3.1.2.1 Súper Resolución según el dominio de la frecuencia

Estos métodos son ampliamente utilizados, se basan en los siguientes principios fundamentales:

- Las propiedades de intercambio de la Transformada de Fourier;
- La relación de sub muestreo entre la Transformada Continua de Fourier y la transformada discreta de Fourier ;y
- La escena original tiene bandas limitadas.

Estas propiedades permiten la formulación de un sistema de ecuaciones relacionando tales coeficientes, las mismas pueden ser resueltas utilizando los coeficientes originales del dominio de la frecuencia recuperados mediante la transformada inversa de Fourier.

2.3.1.2.2 Súper Resolución según el dominio espacial de las imágenes

En estos métodos, se elabora un modelo de observación para elaborar una reconstrucción en el modelo espacial, los modelos espaciales lineales observados pueden ser adecuados para permitir movimiento global o no, borrado óptico, motion blur y variación espacial,

Según el dominio espacial tenemos algunos métodos relacionados:

- Interpolaciones de muestras espaciadas no uniformes
- Retroproyección iterada.
- Métodos estocásticos de reconstrucción SR
- Métodos de reconstrucción basados en conjuntos teóricos.
- Métodos híbridos, MAP, POCS, ML
- Filtros adaptativos y óptimos.
- Regularización Tikhonov – Arsenin.

2.3.1.3 Según el campo de aplicación en Súper Resolución

Las técnicas de súper resolución pertenecen al campo de procesamiento de imágenes.

Se han desarrollado técnicas de súper resolución específicas según el área de aplicación, en algunos casos se realiza todavía a nivel de investigación como en la medicina, es decir, no hay aplicaciones clínicas

Se presenta ejemplos de campos de aplicación, a nivel de investigación y/o nivel de aplicación comercial:

- Imágenes Médicas (Rayos X, Mamografía, CT, MRI, Ecografía, etc.)
- Militar
- Seguridad
- Robótica
- Imágenes aeroespaciales y satelitales
- Aerofotografía
- Análisis Microscópico, macroscópico

- Entre otros.

2.3.2 Otros conceptos de Súper Resolución:

- Súper Resolución es el proceso de producir una imagen con alta resolución espacial a partir de algunas imágenes con baja resolución.
- Súper Resolución es el proceso de fusión para reconstruir una imagen de alta resolución a partir de algunas de baja resolución que cubren la misma región.
- Súper Resolución involucra un conjunto de conversiones simultánea a la entrada de muestreo para reducir o eliminar el solapamiento y/o distorsión.
- La técnica de súper - resolución se basa en aprovechar la información no redundante presente en una secuencia de imágenes.
- Súper - resolución es el proceso de combinar múltiples imágenes en baja resolución para formar una imagen con alta resolución.
- Súper Resolución son las técnicas para lograr una alta resolución al ampliar o alargar los pixeles de imágenes o videos de baja resolución.

CAPÍTULO III: FUENTE DE DATOS

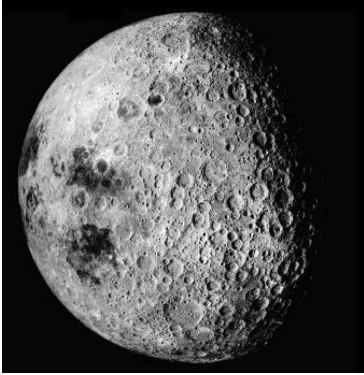


3.1 Recolección y Fuentes de Datos

Se realizó la recolección de imágenes digitales de las siguientes fuentes:

- Las imágenes médicas fueron colectadas del Hospital Cayetano Heredia [5];
- Las imágenes aeroespaciales para las pruebas unitarias y ejemplos, fueron obtenidas del sitio web de la NASA (www.nasa.gov) [3].

3.2 Imágenes Aeroespaciales

Las imágenes aeroespaciales han sido descargadas del sitio web de la NASA [3], las cuales son de diferentes misiones realizadas, entre ellos tenemos imágenes de la Luna, Plutón y hasta la superficie de Marte, en este caso obtenidas desde la misión Curiosity. El objetivo de trabajar con estas imágenes es mostrar también la aplicabilidad de los métodos propuestos en este tipo de imágenes.

Nro	Imagen	Tipo de imagen	Tamaño (Kb)	Formato	Fuente
1		Aeroespacial : Cara oculta de la luna	HR:90 Kb 702 x 720	JPEG uint8, 8 bits por pixel [0 a 255]	NASA
2		Aeroespacial : NASA Rovers Reward for Climbing Exposed Geological Contact	HR:149 Kb 1041 x 642	JPEG uint8, 8 bits por pixel [0 a 255]	NASA
3		Aeroespacial : Cráter lunar	HR:81 Kb 742 x 696	JPEG uint8, 8 bits por pixel [0 a 255]	NASA

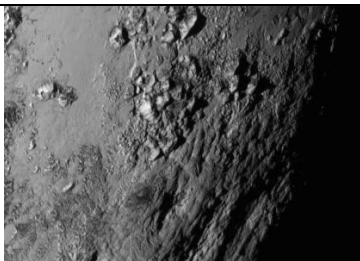
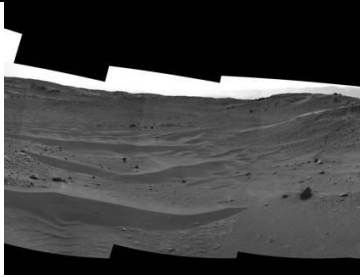
4		Aeroespacial : Montañas de hielo en superficie de Plutón	HR:93 Kb 1041 x 742	JPEG uint8, 8 bits por pixel [0 a 255]	NASA
5		Aeroespacial : Superficie marciana tomado por el Curiosity	HR:14 Kb 430 x 323	JPEG uint8, 8 bits por pixel [0 a 255]	NASA

Tabla 6: Lista de imágenes aeroespaciales utilizadas para pruebas sintéticas

Fuente: Imágenes obtenidas de la NASA ()

Elaboración: Enzo Camargo Fernández - Baca

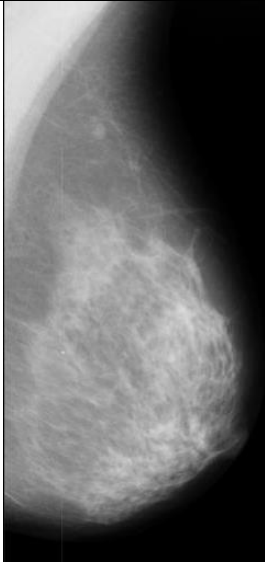
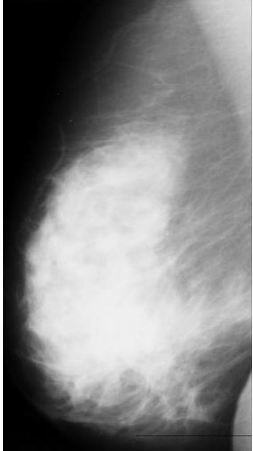
Fuente: www.nasa.gov [3]

3.3 Imágenes Médicas

Las imágenes médicas han sido colectadas del área de radiología del Hospital Cayetano Heredia, como parte del proyecto Súper Rivam [5], siendo el enfoque principal las imágenes de mamografías.

Como se ve en Tabla 7, analizando una imagen se tiene el fondo totalmente negro, grises hasta totalmente blancos, esto se debe a que cuando los rayos X (utilizados en este caso para la mamografía) se disparan y como no encuentran ningún obstáculo en su camino, regresan igual y se marca como totalmente negros, en el caso que encuentran un obstáculo muy grande como en la parte superior izquierda en la Tabla 7, ítem 1, se obtiene el color blanco o cercano, en cambio cuando el órgano a examinar es menos denso se observan los otros niveles de grises.

Las siguientes imágenes, se han obtenido inicialmente en el formato DICOM, para poder ser trabajados en el presente proyecto se han pasado a JPEG.

Nro.	Imagen HR	Tipo de imagen	Tamaño (Kb)	Formato	Fuente
1		Medica : mamografía Nombre Archivo:	HR: 12.8 Kb (512 x 512 x 3)	DICOM, original JPEG 8 bits x canal x 3 canales RGB	HCH
2		Medica : mamografía	HR:12.8 Kb (512 x 512 x 3)	DICOM, original JPEG 8 bits x canal x 3 canales RGB	HCH


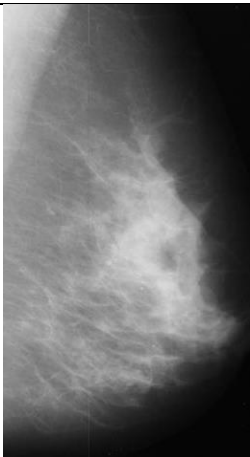
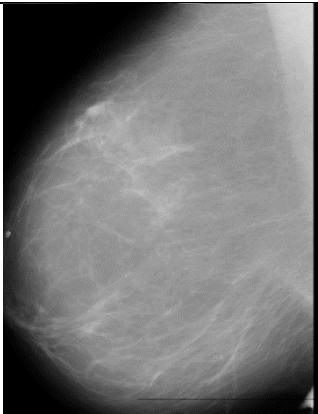
3		Medica : mamografía	HR:13.3 Kb (512 x 512 x 3)	DICOM, original JPEG 8 bits x canal x 3 canales RGB	HCH
4		Medica : mamografía	HR:13.3 Kb (512 x 512 x 3)	DICOM, original JPEG 8 bits x canal x 3 canales RGB	HCH
5		Medica : mamografía	HR:16.1 Kb (512 x 512 x 1)	DICOM, original JPEG 8 bits x canal x 1 canales Grayscale	HCH

Tabla 7: Lista de imágenes médicas utilizadas para el proyecto

Fuente: Hospital Cayetano Heredia. Unidad de Radiología. Lima – Perú.

Elaboración: Enzo Camargo Fernández - Baca

CAPÍTULO IV: DESARROLLO DE LA SOLUCION

En este capítulo, se describe el proceso de obtención de imágenes digitales en general, indicando los fenómenos que afectan su calidad, y su modelamiento conceptual, matemático y desarrollar la solución mediante los métodos elegidos para realizar el proceso inverso minimizando los fenómenos que le afectan durante el proceso de obtención de la imagen. Se eligen 2 métodos de Súper Resolución, a fin de obtener una imagen en Súper Resolución creada a partir de la imagen o imágenes de entrada de baja resolución.

Los métodos de Súper Resolución presentados son el Iterative Back Projecting (IBP) y el método Iterative Kernel Steering (IKS).

Para cada método, se presenta como funciona, esquemas, diagrama de flujo, el modelo matemático, se presentan los ejemplos así como medir la calidad de una imagen en Súper Resolución obtenida, mediante el PSNR, así como la evaluación y comparación entre ambos métodos.

4.1 Cómo funciona la solución planteada

En la siguiente Figura 27, se muestra el esquema general del problema y su solución, precisando lo siguiente:

1. El médico toma las imágenes médicas mediante equipos de captura como ecógrafos, entre otros, la cual es visualizada en su monitor,
2. Se aplica a la imagen obtenida, los métodos de Súper Resolución y los resultados se muestran en una pantalla adicional para esta solución, se visualiza el resultado en SR, es importante indicar que el tiempo de procesamiento máximo es de 1 a 3 minutos referencialmente, para poder observar ambas imágenes en paralelo.
3. En el acercamiento realizado a la imagen que se encuentra en el lado derecho inferior, se tiene la imagen (a) LR (1x) y la imagen (b) el SR (3x), en los cuales se observan con claridad que se amplían en 3 veces los píxeles en cada dirección, asimismo se observa que se colocan nuevos píxeles entre los que habían previamente. La presente Tesis plantea la solución desde el punto de vista computacional y no se presentan pruebas médicas que demuestren sean aplicables en usos médicos de forma inmediata, esto formaría parte de un trabajo futuro, y no forma parte de la presente Tesis.

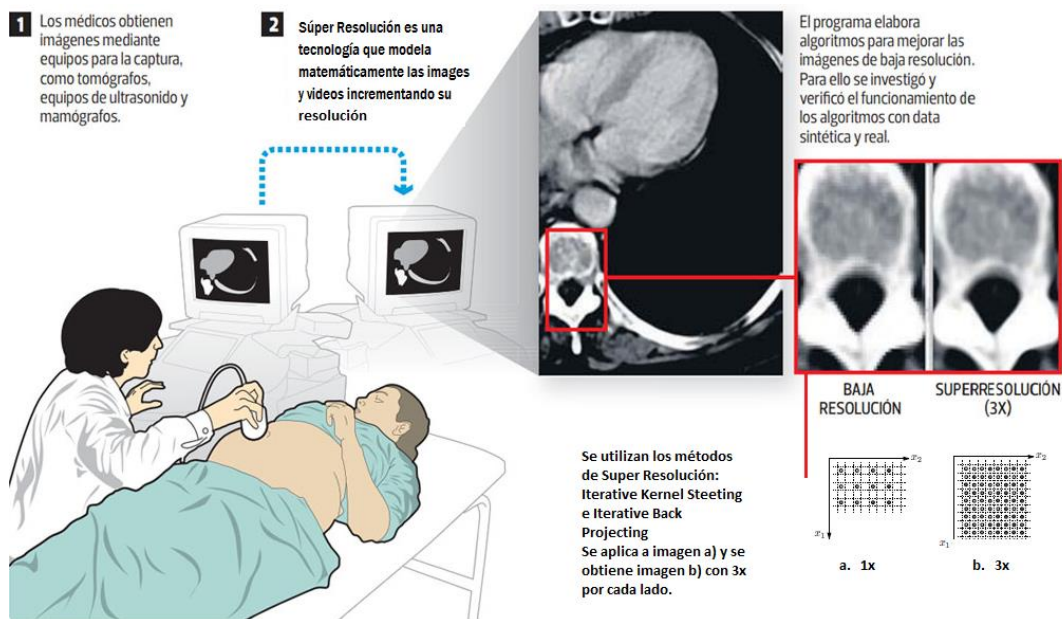


Figura 27: Esquema General de la Tesis que plantea el problema y su solución utilizando Súper Resolución con los métodos Iterative Kernel Steering(IKS) e Iterative Back Projecting (IBP)

4.2 Formulación matemática de la Solución

Según [6] y [7], la Súper Resolución modela la solución desde el punto de vista de la observación:

$$Y_k = D * B_K * M_k * X + N_k, \text{ donde } 1 \leq k \leq p \quad (4.1)$$

Dónde:

Y_k: representa la imagen obtenida baja resolución (LR)

D: es el factor de reducción de tamaño (operador de decimación o decimation operator, en inglés);

B_k: representa el efecto borroso de las lentes de la cámara (blurring operator)

M_k : representa las transformaciones geométricas que se tienen que hacer entre las imágenes de baja resolución con la imagen (frame) de referencia, y

N_k : representa el ruido o error (additive noise).

X : representa la imagen en HR deseada

$k = 1, 2, \dots, p$, donde k , es la k -ésima imagen en LR,

$p =$ representa en número de imágenes en LR disponibles, usualmente se crean en las pruebas sintéticas como una potencia de 2 (2, 4, 16, ...)

Los métodos de Súper Resolución se enfocan en resolver la ecuación 2.9.

4.3 Esquema general de la solución

Según siguiente Figura 28, la solución es realizar el proceso a la inversa de cómo se genera el problema.

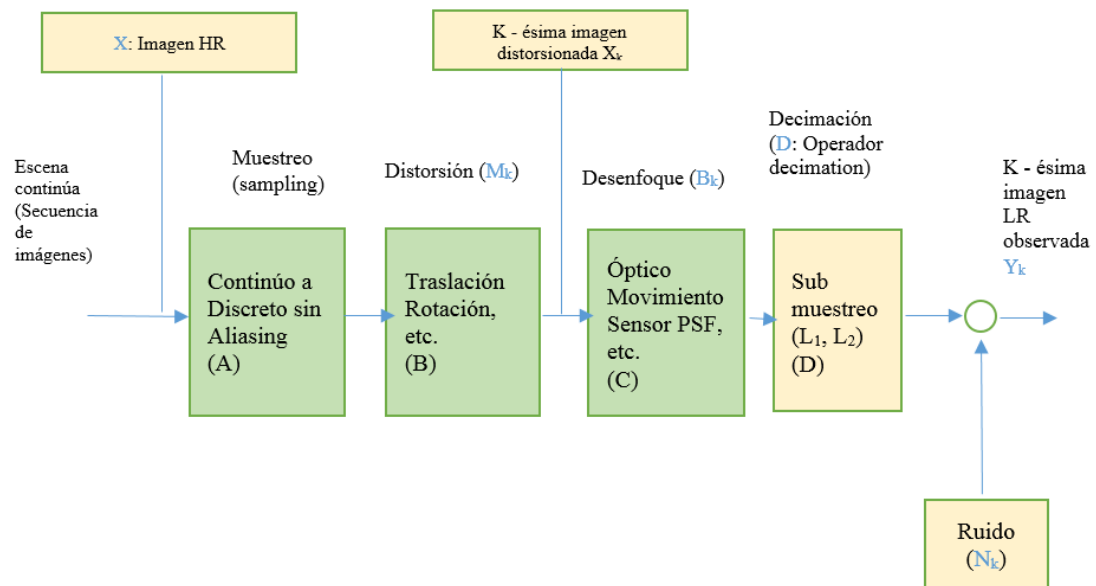


Figura 28: Diagrama de Bloques basado en la ecuación 4.1

Fuente: Súper-resolution mosaicking from digital surveillance video captured by unmanned aircraft systems [6]

Elaboración: Traducido del inglés, basado en [6]

Existen muchos métodos de súper resolución e interpolación que buscan mejorar la calidad de la imagen, en general todos estos métodos buscan minimizar todas o parte de los fenómenos que afectan la calidad de la imagen, cada método puede hacerlo en un orden distinto. En todos los casos, la forma de realizar las pruebas es a través de las pruebas sintéticas teniendo como imagen original real a HR, a partir de la cual se crean las imágenes de entrada LR.

Con la finalidad de minimizar estos fenómenos que afectan la calidad de la imagen, se realiza la operación inversa indicada en la Figura 28, mediante el modelado matemático de estos fenómenos basados en la ecuación 4.1, y esto permite despejar luego la variable de salida de interés, la cual es la imagen en SR. Estos modelos tienen como objetivo lograr la mejor aproximación posible a la imagen HR, la cual está determinada por el PSNR más alto.

Entonces los modelos matemáticos tienen como objetivo “reconstruir” la imagen en la realidad, planteando el problema de reconstrucción de una imagen en Súper Resolución y a partir de ello, su solución.

Por lo tanto, los modelos matemáticos a establecer son:

1. Remover el Ruido
2. Aumentar la frecuencia de muestreo, up sampling.
3. Remover el desenfoque, según el PSF.
4. Registración, para llevar a un mismo sistema de coordenadas en base a los parámetros de desplazamiento, rotación y traslación, esto se aplica en el caso que sea necesarias más de 1 imagen.
5. Construir la imagen en Súper Resolución

Remover el ruido implica usar un tipo de filtro sobre la imagen, el tipo de filtro depende de la naturaleza del ruido y el modelo matemático que se aproxima.

Minimizar el Aliasing, mediante el aumento de la frecuencia de muestreo (up sampling), físicamente el Aliasing puede ser causado por la densidad del sensor, velocidad del obturador, entre otros factores del propio equipo, considerar que la imagen es una escena continua y luego se pasa a un muestreo discreto.

Remover el Desenfoque, para ello determinar el PSF del sistema de adquisición de imágenes, el cual depende del hardware del CCD en sí. Para el entorno médico, el PSF varía según cada CCD, por ejemplo si tenemos un equipo de rayos X de un modelo A, y tenemos otro equipo de rayos X de un modelo B, y si $A \neq B$, entonces $PSF(A) \neq PSF(B)$

Realizar la registración, mediante la determinación de los parámetros de desplazamiento y rotación (proceso intermedio), Siguiendo el orden, es necesario determinar los parámetros de desplazamiento: rotación y desplazamiento entre las imágenes. Para un conjunto de imágenes de

entrada, estos parámetros se determinan respecto a una imagen de referencia, es decir, se realiza una comparación de cada una de las imágenes de entrada con respecto a la imagen de referencia.

El método de registración de imágenes con aliasing en el dominio de las frecuencias de Patrick Vanderwalle, Sabine Susstrunk, Martin Vetterli en [22] ha sido seleccionado, para determinar los parámetros de desplazamiento y rotación de las imágenes de entrada en LR. En este caso utilizamos la imagen de referencia = imagen LR (1), es decir, la primer imagen.

Construir la imagen en Súper Resolución, Una vez determinado los parámetros de desplazamiento, entre un conjunto de imágenes se aplican un método para construir la imagen en Súper Resolución.

La ecuación 4.1 se esquematiza en la Figura 28, en esta figura la imagen original (HR, real image) es una señal continua, cuando se inicia el proceso de obtener la imagen digital, la cual esta formada por 1s y 0s (unos y ceros), y debido al muestreo se pierde cierta informacion (Figura 3, A).

4.3.1 Análisis Efecto – Causa – Solución en el proceso de obtener una imagen en Súper Resolución (SR) a partir de una imagen en baja resolución (LR)

Efectos o Problemas causados en la Imagen	Causas/fenómenos que se presentan	Soluciones	Como medir calidad de la imagen
<p>Distorsión (warp)</p> <p>Borroso (blur) imagen desenfocada imagen rotada y/o trasladada</p> <p>Imagen con Aliasing</p>	<p>Traslación y/o rotación</p> <p>ruido atmosférico interno(propio del equipo)</p> <p>desenfoque óptico movimiento sensores PSF, etc.</p> <p>muestreo (sampling) continuo a discreto (la imagen real tiene una señal continua, al pasar a digital, se pasan a 1s y 0s)</p> <p>decimación (decimación) Disminuir el muestreo (down sampling)</p>	<p>Registración</p> <p>Minimizar ruido Filtros</p> <p>Interpolación</p> <p>Modelar ruido</p> <p>Eliminar, desenfoque minimizar</p> <p>Registración Modelar PSF</p> <p>Filtro</p> <p>Aumentar muestreo de información mediante interpolación, Filtros</p>	<p>PSNR (objetivo) MSE (objetivo) Ojo humano(subjetivo)</p> <p>Como probar un algoritmo</p> <p>Pruebas sintéticas, es decir, pruebas creadas a partir de datos conocidos, usando el PSNR para determinar la calidad (*)</p> <p>(*) Considerar que el PSNR solo puede ser aplicado en caso de pruebas sintéticas</p>

Figura 29: Cuadro Resumen sobre la interrelación entre los efectos, causas y soluciones para obtener una imagen SR a partir de una imagen LR.

Fuente: Enzo Camargo Fernández - Baca

Elaboración: Enzo Camargo Fernández - Baca

La solución consiste en eliminar, minimizar, suavizar las causas desde las imágenes de LR para obtener la imagen de SR a partir de ella (s). La implementación de las soluciones se realiza a través de distintos métodos o algoritmos.

4.3.2 Vista general de los 2 métodos de súper resolución: Iterative Kernel Steering (IKS) e Iterative Back Projecting (IBP)

Los métodos desarrollados en la presente tesis, son el Iterative Back Projecting, el cual se aplica cuando se tienen varias imágenes para lograr la imagen en SR, y el método del Método Iterative Kernel Steering, cuando se tiene una sola imagen

El método de Iterative Back Projecting, se basa en el Back Projecting de las tomografías, el primer paso es realizar la registración, luego de ello, se estima la imagen en Súper Resolución inicial utilizando para ello el promedio entre las imágenes de baja resolución, considerando que no se toma los valores de frecuencias altas para no alterar el resultado de la estimación inicial de la imagen de Súper Resolución. Luego de ello, se inician las iteraciones para cada imagen, que consiste en adicionar información a partir de la estimación inicial de la imagen SR y el aporte de cada imagen en LR. Finalmente toda la información se integra basada en los aportes de cada imagen de LR.

En el caso del método de Iterative Kernel Steering, se enfoca en utilizar la información relevante para cada pixel según sea su vecindad con la finalidad de optimizar el resultado. Es decir, si el pixel pertenece a un borde, se utiliza con prioridad la información del borde para mejorar su contenido. El método IKS puede usar el Filtro Gauss para el cálculo. Es por ello que se llama adaptativo a los datos (data adaptative), el cual es una evolución del método de Kernel tradicional que utiliza de forma homogénea la información cercana

(vecindad cercana al pixel), siendo en el caso del IKS se utiliza la información que pueda ser más relevante para el punto y esto varía de pixel a pixel.

En la Figura 30, se presenta un Diagrama de Flujo General de ambos métodos, observando que:

1. Si se tiene una imagen, solo es posible utilizar el método de Súper Resolución Iterative Kernel Steering , en caso que se tengan más es posible utilizar el método de Súper Resolución IBP
2. En el caso del método IBP, se utiliza el aporte de cada imagen en LR limitando las frecuencias altas, siendo el enfoque principal, tomar lo mejor adicional de cada imagen para lograr una óptima SR.
3. En el caso del método de Súper Resolución IKS, se analiza la vecindad más influyente para cada pixel, a diferencia del Kernel tradicional que selecciona una vecindad regular, para ser utilizada como cálculo para los Subpixeles creados.

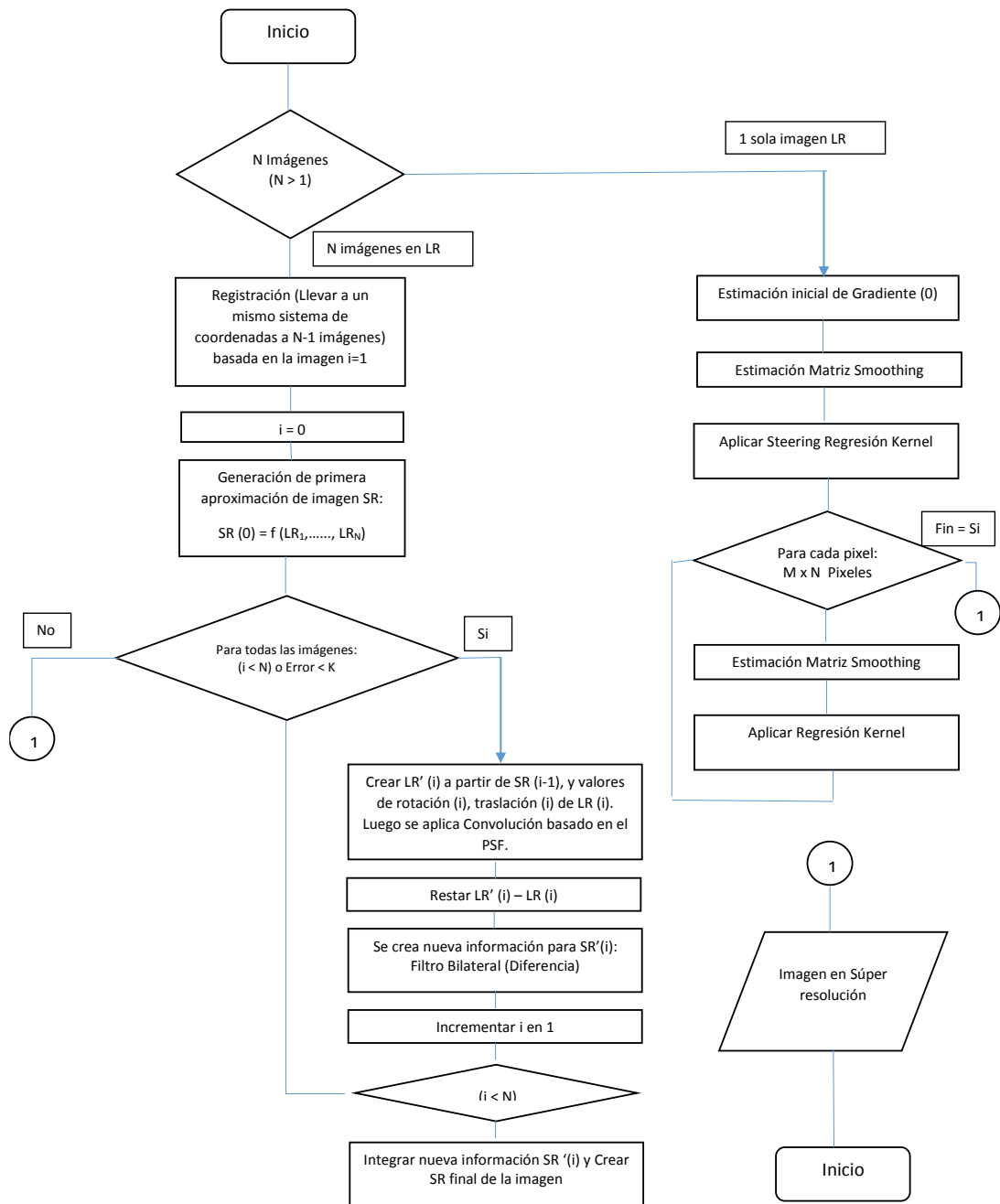


Figura 30: Diagrama de Flujo General de ambos métodos

Fuentes: [12], [17]

Elaboración: Enzo Camargo Fernández - Baca

4.3.2.1 Consideraciones para el desarrollo de ambos métodos

- Las imágenes están en escala de grises, es decir, no se han realizado pruebas con imágenes a color.
- Las imágenes se procesan en $M \times N \times 1$, es decir, en un canal.

4.4 Método de Súper Resolución basado en varias imágenes: Iterative Back Projecting

De acuerdo a [12], la resolución de una imagen puede ser mejorada cuando los relativos desplazamientos de las imágenes de entrada son conocidos con precisión y se tiene algún conocimiento del proceso de adquisición de imágenes. El método propuesto es similar al back projection usado en tomografía. Este método de Súper Resolución es aplicable cuando se tiene varias imágenes de entrada, y nos permite obtener una imagen de mejor calidad hasta en 8x, minimizando el ruido y el desenfoque, aplicado a imágenes digitales médicas y en imágenes digitales en general. Eso se realiza mediante la obtención de información relevante de cada imagen LR y adicionarlo a la imagen final en Súper Resolución.

Desarrollado por Michael Irani and Samuel Peleg [12], inicialmente la investigación llevada a cabo por Tsai y Huang, quienes usan el método en el dominio de la frecuencia. Su trabajo ha sido enfocado en el efecto borroso en el procesamiento de imágenes, ellos usan las imágenes trasladadas para manejar la pérdida de datos, debido a la decimación.

Este método además de obtener resultados relevantes permite ilustrar de forma sencilla el enfoque de un método de Súper Resolución cuando se tienen

varias imágenes (video de una ecografía, toma repetitiva de una zona específica del cuerpo como una mamografía, rayos X) entre otros. Se inicia con la registración (es decir, pasar todas las imágenes a un mismo sistema de coordenadas, debido a que pueden estar rotadas o trasladadas) considerando como la imagen base la primera, como segundo paso, se crea una imagen de Súper Resolución inicial basada en el promedio de las imágenes de baja resolución de entrada; luego como tercer paso, se inicia una iteración por cada imagen hasta un número determinado de iteraciones o hasta que el error sea menor, en la cual se obtiene información relevante de cada imagen en baja resolución, aplicando un Filtro bilateral a la diferencia de imágenes y como cuarto paso finalmente, se adiciona las diferencias encontradas a la imagen de Súper Resolución, obteniendo una imagen de Súper Resolución final.

El algoritmo iterativo planteado por PEREG e IRANI, incrementa la resolución de la imagen, así como realiza la Registración (ajuste del desplazamiento y rotación). Este algoritmo es aplicado para imágenes LR en nivel de grises y en color, con un claro incremento de la resolución luego de algunas iteraciones.

La registración, permite tener las imágenes de baja resolución en el mismo sistema de coordenadas y por lo tanto permitirán añadir información con menor ruido.

Peleg estima una imagen SR inicial y simula el proceso de la obtención de imágenes, para obtener un conjunto de imágenes de LR simuladas. La función de error definida por el actual y la imagen LR sumadas, el cual es minimizado iterativamente hasta que no se pueda mejorar más, o hasta que se llegue a un número máximo de iteraciones.

Este método requiere la estimación de las matrices D_k y H_k , y tiene la característica de disminuir el efecto del ruido \underline{E}_k .

En la tomografía, las imágenes son reconstruidas desde sus proyecciones en varias direcciones. En el caso de SR, cada pixel de baja resolución es una "proyección" de una región en una escena donde el tamaño es determinado por la imagen de entrada. La construcción de la imagen en SR, es usando un método similar al back Projecting usado en Tomografía computarizada [12]

Este método tiene buenos resultados para imágenes libres de ruido, pero es altamente sensible al ruido y lento para converger, lo cual es demostrado en la experimentación realizada en el Capítulo 5.

4.4.1 Esquema del Método IBP

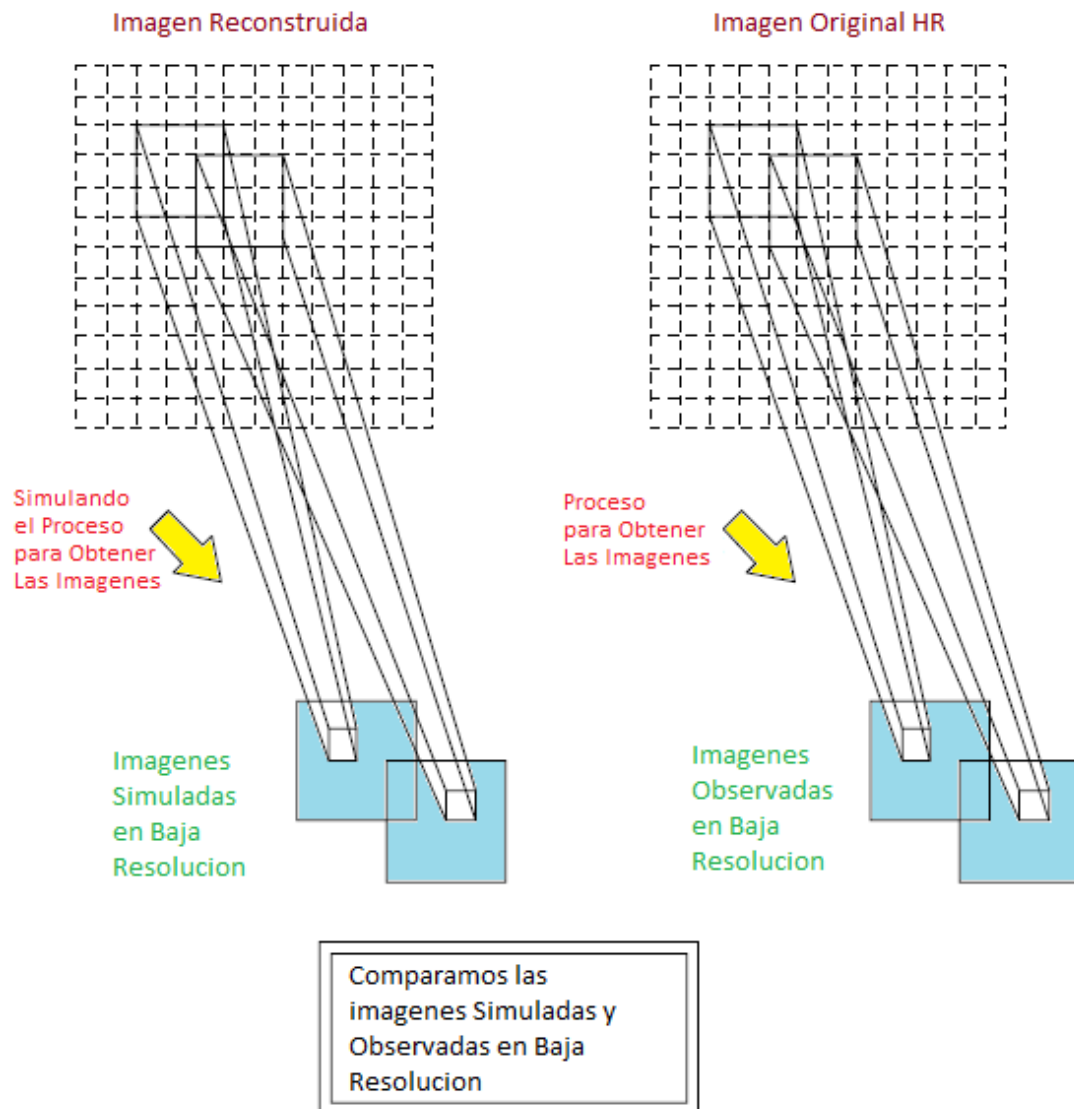


Figura 31: Diagrama esquemático de algoritmo IBP de Súper Resolución

Fuente: Improving Resolution By Image Registration [12]

Elaboración: Michael Irani and Samuel Peleg

Una imagen de alta resolución es reconstruida (izquierda), y provee imágenes simuladas en baja resolución (LR) que son lo más cercano posible a las imágenes de baja resolución observadas.

4.4.2 Modelamiento matemático del Método IBP

4.4.2.1 Estimación del Desplazamiento entre dos Imágenes

Estimar el desplazamiento en el dominio de las frecuencias se basa en la propiedad de la transformada de Fourier, expresada en la siguiente Figura:

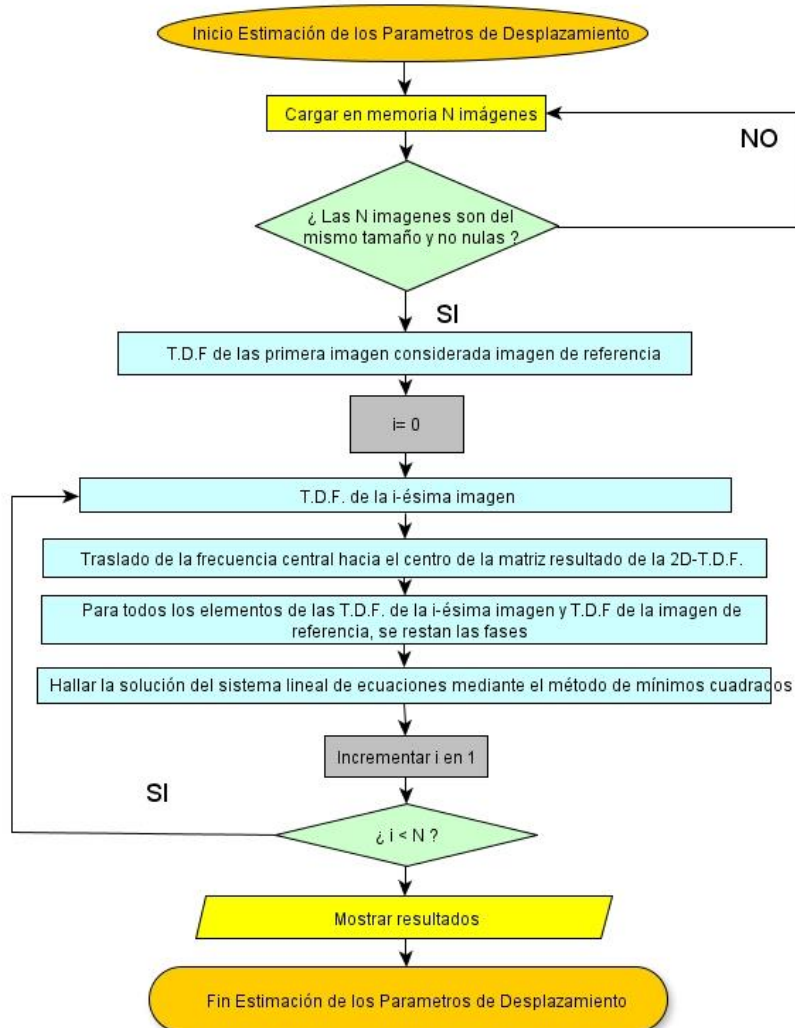


Figura 32: Diagrama de Flujo de la estimación del Desplazamiento entre dos Imágenes

Fuente: [12]

Elaboración: Tomado de [12]

4.4.3 Diagrama de Flujo del método IBP

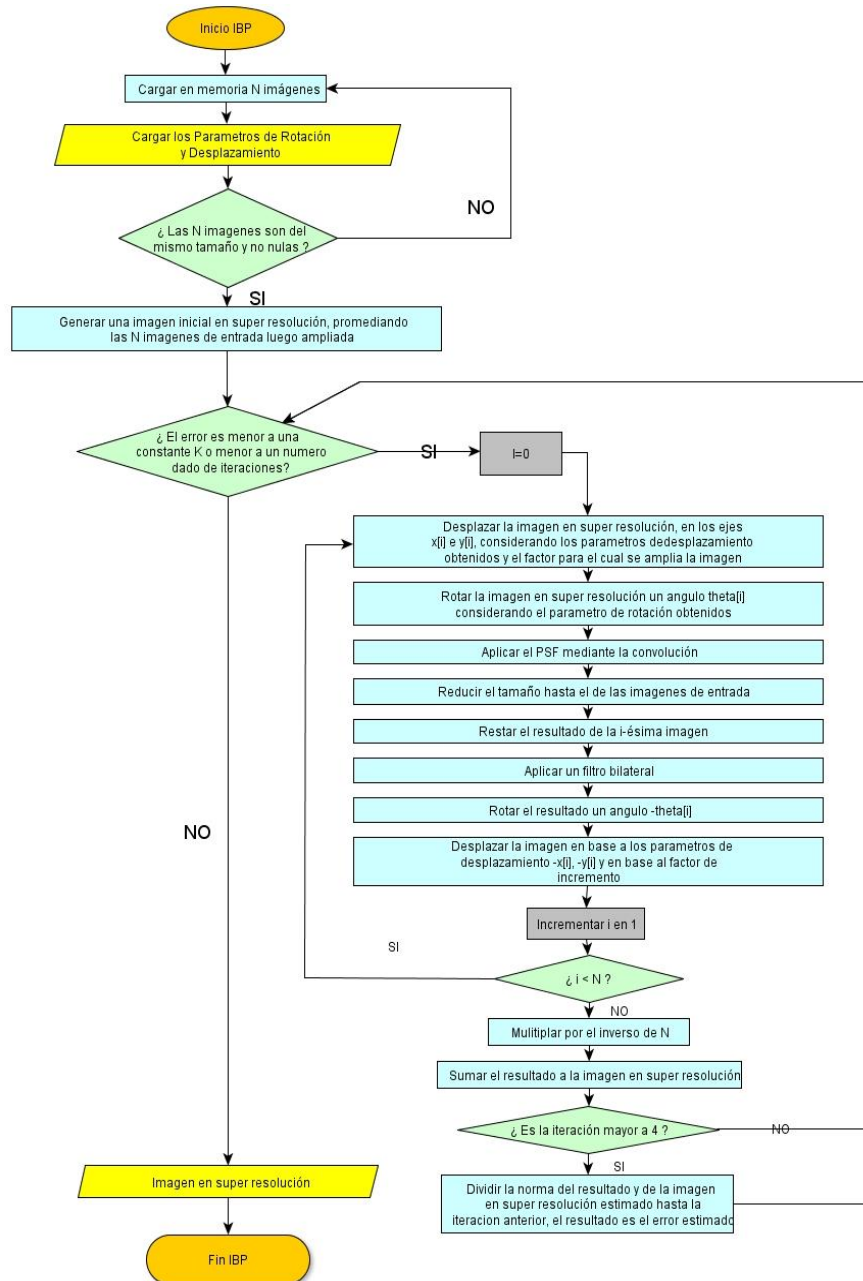


Figura 33: Diagrama de flujo propuesto para el algoritmo IBP.

Fuente: [12]

Elaboración: Tomado de [12]

4.4.4 Ejemplos de aplicación del método IBP

Se inicia el ejemplo de aplicación del método IBP a una imagen digital obtenida desde una cámara fotográfica a colores RGB en la **Figura 34**, iniciando el siguiente procedimiento para crear la prueba sintética:

Primero, utilizamos esta imagen a color para crear una imagen en escala de grises, la cual será la escena original o imagen real HR (Figura 34, b) y esta a su vez la utilizamos para crear imágenes LR como la indicada en (Figura 34, c), que es la data o imagen de entrada.

Luego se procesa esta información de entrada utilizando el algoritmo IBP, y se obtiene la imagen en SR en la Figura 35 (b), en donde se aprecia un acercamiento. Se observa en esta figura bordes mejor definidos y mejor resolución a la vista del ojo humano.

4.4.4.1 Ejemplo de imagen UNI



(a) Imagen Original: Vista desde
Facultad Metalurgia UNI 2015



(b) Imagen Original en escala de grises (HR)



c) Imagen en baja resolución (LR)

Figura 34: Creación de prueba sintética basada en una imagen original en colores(a), luego se convierte a Escala de Grises (b), y finalmente se crea la imagen de menor resolución (c) llamada imagen LR.

Fuente: Imagen fotográfica, 2015.

Elaboración: Enzo Camargo Fernández - Baca

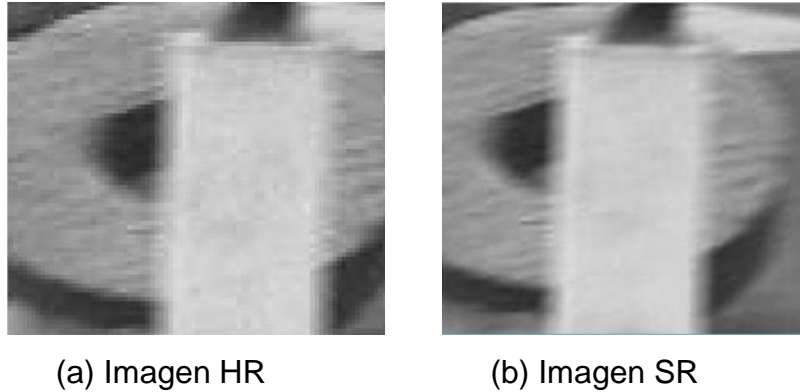


Figura 35: Ejemplo de método de Súper Resolución IBP aplicado a una imagen fotográfica

Fuente: HR: Extracto tomado de Figura 34. Noise = 0, varianza = 0.001. Imágenes sintéticas generadas

Elaboración: Enzo Camargo Fernández - Baca

4.4.4.2 Ejemplo de imagen Médica:

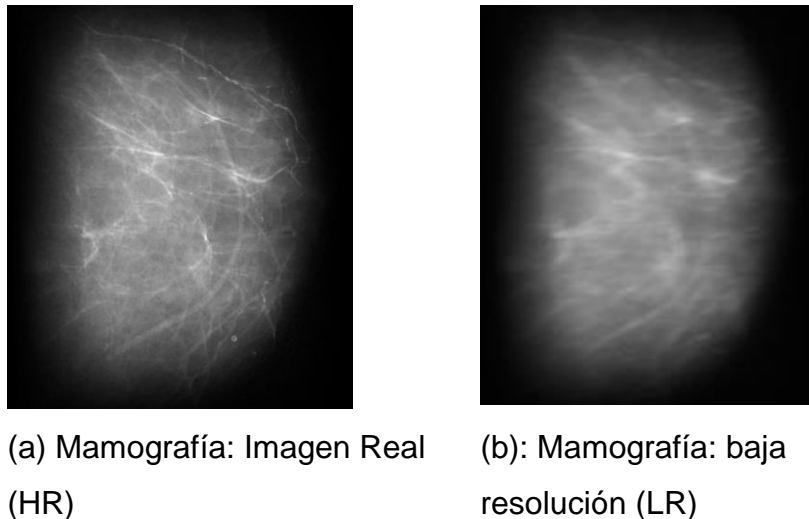
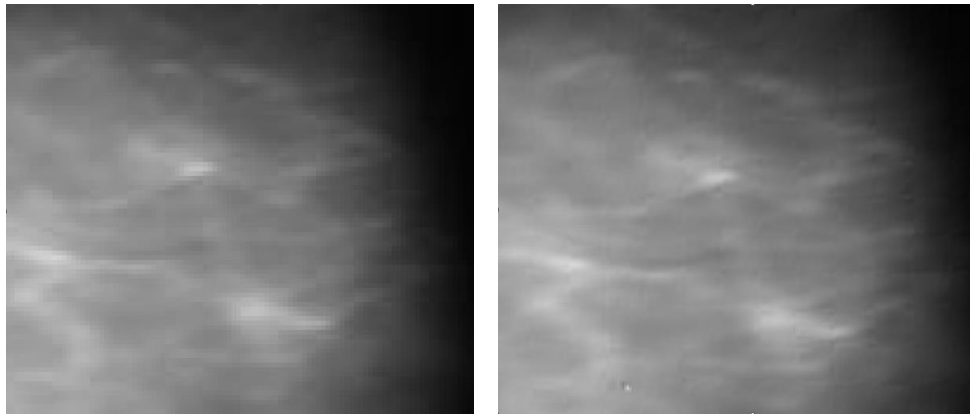


Figura 36: Ejemplo de método de Súper Resolución IBP aplicado a una mamografía

Fuente: Hospital Nacional Cayetano Heredia. [7]

Elaboración: Enzo Camargo Fernández - Baca



*(a) Imagen en baja
resolución tomada de la
Figura 36(a) (LR)*

*(b) Imagen en Súper
Resolución utilizando
IBP, 4 iteraciones*

Figura 37: Acercamiento a Figura 36, se puede observar en la imagen de Súper Resolución (b) que los bordes están mejor definidos.

Fuente: Extracto tomado de Figura 36(a)

Elaboración: Enzo Camargo Fernández - Baca

4.5 Método de Súper Resolución basado en una imagen: Iterative Kernel Steering (IKS)

El método presentado está en el campo de estadísticas no paramétricas y presenta el desarrollo y generalización de herramientas y resultados para el uso en procesamiento y reconstrucción de imágenes. Se propone una extensión al uso del método tradicional del Kernel a uno adaptativo en los datos (data adaptive) para la eliminación de ruido, aumento del muestreo, interpolación, fusión, entre otros. Adicionalmente, se establecen relaciones claves con algunos de los métodos populares existentes como el Filtro bilateral, el cual se plantea como caso especial del framework indicado.

Este método se basa en el trabajo desarrollado por Hiroyuki Takeda y Peyman Milanfar [16] y [17].

La principal contribución de este método es:

- Describir el propósito de regresión Kernel como una efectiva herramienta para ambos (reducir ruido e interpolación) en procesamiento de imágenes,
- Desarrollar un método adaptativo generalizable de la regresión Kernel con excelentes resultados finales, que se enfoca en tomar información relevante para cada pixel

Los métodos de regresión son utilizados en el procesamiento de imágenes, con la finalidad de adicionar información, minimizar ruidos, mejorar el muestreo.

Regresión de Kernel es una técnica no-paramétrica en estadística para estimar el valor esperado de una variable aleatoria. El objetivo es buscar una

relación no lineal entre el par de variables aleatorias X e Y . Es decir, cualquiera que pueda ser la relación o comportamiento entre estas variables se puede aplicar el método de regresión de kernel, para estimar el valor esperado de Y , dado un X determinado.

Esto puede ser representado como:

$$E(Y | X) = m(X) \quad (4.2)$$

Donde m es una función desconocida.

Esto lo podemos ilustrar con el siguiente ejemplo basado en Regresión Kernel Gaussiano:

Suponiendo que se tiene valores aleatorios como X , que se visualizan en la siguiente Figura y se requiere calcular el valor de Y dado un X , además no existe una relación conocida entre X e Y , podríamos aplicar el método de regresión de Kernel, el cual produce valores bastante próximos como se observa en la Figura 38 (b).

En el caso de una imagen, que se expresa en términos similares, el método de regresión de Kernel puede estimar valores similares dados valores de X aleatorio. Un valor aleatorio X en una imagen, conceptualmente se puede asegurar que este método se podría aplicar a cualquier imagen, sea imagen médica o no.

4.5.1 Código de programa en MATLAB de la regresión Kernel tradicional aplicado a una función aleatoria de X

```
clear;close all;clc;

x = 1:100; % training data x
y = sin(x/10)+(x/50).^2+0.2*randn(1,100); % training
data y
h=5; % kernel bandwidth

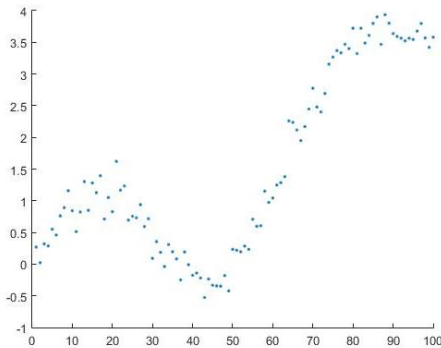
for i=1:100
    xs(i)=i;
    ys(i)=gaussian_kern_reg(xs(i),x,y,h);
end

figure;hold on;
plot(x,y,'.');
plot(xs,ys,'r-');

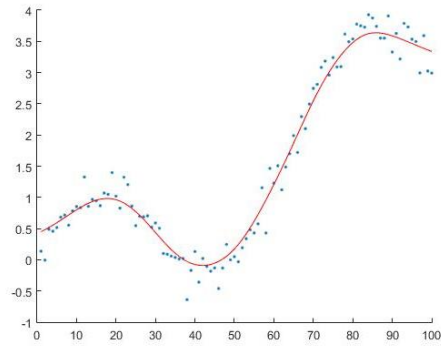
function ys=gaussian_kern_lin_reg(xs,x,y,h)

% Gaussian kernel function
kerf=@(z)exp(-z.*z/2)/sqrt(2*pi);

z=kerf((xs-x)/h);
ys=sum(z.*y)/sum(z);
```



(a) Se tiene valores aleatorios de X



(b) Mediante el método de Regresión Kernel, se estima los valores de Y correspondientes aproximados

Figura 38: Ejemplo de Regresión Kernel aplicado a una función aleatoria
Fuente: Desarrollo de ejemplo con Matlab.

Elaboración: Enzo Camargo Fernandez – Baca

4.5.2 Esquema de la regresión Kernel

Basados en el ejemplo anterior, podemos explicar cómo funciona este método cuando se aplica a imágenes digitales, y de este modo obtener imágenes de mejor resolución.

4.5.2.1 Eliminación de ruido

Como se observa en la Figura 38 (b), existen puntos encima de la línea generada por la Regresión Kernel, estos puntos considerados fuera de la proyección, son considerados como ruido. En el momento que se aplica una

función o método de Súper Resolución, los puntos que quedan por fuera de función, son considerados también como ruido de la imagen.

4.5.2.2 Upsampling

El upsampling tiene como objetivo aumentar la tasa de muestreo, es decir, adicionar información donde no hay, basado en los datos iniciales y dependiendo del algoritmo o método utilizado.

En el caso del ejemplo indicado, tenemos los valores

Como se observa en la figura (a), luego que una imagen es muestreada se tienen valores puntuales, los cuales son aleatorios.

Cuando $X = 31.5$, entonces podemos obtener el valor de Y determinado, es decir tenemos valores continuos aun si la función inicial es discreta con el Dominio inicial de $X = \{1, 2, \dots, 100\}$.

Por lo tanto, se ha aumentado la tasa de muestreo (Upsampling) porque dado $x = 31.5$, se tiene un y determinado. Estos valores aproximados obtenidos mediante interpolación, permitiendo un **Dominio continuo** y por lo tanto un **Rango continuo**, lo que a su vez significa que se podría adicionar información a la imagen de forma aproximada.

4.5.2.3 Interpolación

La interpolación de valores nos permite a su vez aumentar el muestreo, concepto que hemos desarrollado en el párrafo previo.

4.5.3 Modelamiento matemático del Método de Regresión Kernel

4.5.3.1 Técnicas de regresión kernel en imágenes

Tal como se muestra en [19], el modelo del valor de los pixeles en una posición dada de la imagen se puede en 1-D modelar como:

$$y_i = z(x_i) + \varepsilon_i, \quad x_i = [x_{1i}, x_{2i}]^T, \quad i = 1, 2, \dots, P \quad (4.3)$$

Donde x_i es un vector en 2D que representa las coordenadas de un pixel, $z(\cdot)$ es la función que describe el valor de los pixeles en cada posición, y ε_i es un ruido de media cero. Entonces si se desea hacer Súper Resolución nuestra tarea es conocer la forma de la función $z(\cdot)$ con el fin de predecir los valores de los pixeles en cualquier posición de la imagen. Por ejemplo, en la Figura 39, la imagen original es de 3x3(casilleros rojos), y el conocimiento de la función $z(\cdot)$ nos permite conocer el valor de cada pixel en los casilleros negros. Esto nos permite obtener una imagen de 9x9 de más resolución.

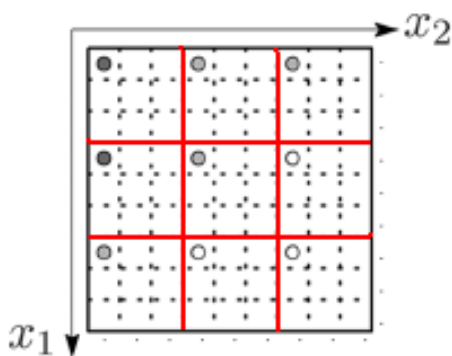


Figura 39: Esquema de Súper Resolución de una imagen de 3x3.

Fuente: Kernel Regression for Image Processing and Reconstruction

Elaboración: Hiroyuki Takeda, Sina Farsiu, y Peyman Milanfar

4.5.3.2 Formulación de Regresión Kernel en 2-D

Una manera de hallar $z(\cdot)$ en cierto punto "x", es mediante la expansión de Taylor la cual está dada por:

$$z(x_i) = z(x) + \{\nabla z(x)\}^T (x_i - x) + \dots \quad (4.4)$$

$$\frac{1}{2!} (x_i - x)^T \{Hz(x)\} (x_i - x) + \dots$$

Donde $\nabla z(x)$ y $Hz(x)$ representa el gradiente y operador Hessian de la función $z(\cdot)$. De donde se demuestra que la ecuación 4.4 es equivalente a:

$$z(x_i) = \beta_0 + \beta_1^T (x_i - x) + \dots \quad (4.5)$$

$$\beta_2^T \text{vech}\{(x_i - x)(x_i - x)^T\} + \dots$$

Como se observa, β_0 representa el valor de $z(\cdot)$ que es el valor de la función, es decir, el valor del pixel (su intensidad) en cualquier posición "x". Los demás "betas" son las aproximaciones de las derivadas de orden superior y están dados por:

$$\beta_1 = \left[\frac{\partial z(x)}{\partial x_1} \Big|_{x=x_i}, \frac{\partial z(x)}{\partial x_2} \Big|_{x=x_i} \right]^T \quad (4.6)$$

$$\beta_2 = \frac{1}{2} \left[\frac{\partial^2 z(x)}{\partial x_1^2} \Big|_{x=x_i}, \frac{\partial^2 z(x)}{\partial x_1 \partial x_2} \Big|_{x=x_i}, \frac{\partial^2 z(x)}{\partial x_2^2} \Big|_{x=x_i} \right]^T \quad (4.7)$$

Los valores de "beta" de (6) se pueden hallar al resolver el siguiente problema de optimización

$$\min_{\{\beta_n\}_{n=1}^N} \sum_{i=1}^P [y_i - (\beta_0 + \beta_1^T (x_i - x)^T + \dots + \beta_2^T \text{vech}\{(x_i - x)(x_i - x)^T\} + \dots)]^2 K_H(x_i - x) \quad (4.8)$$

Donde:

$$K_H(t) = \frac{1}{\det(H)} K(H^{-1}t) \quad (4.9)$$

Es una matriz que representa una función Kernel en 2D que depende de una matriz H de suavizado. Tal como se muestra en la Figura 42, el efecto de aplicar una matriz Kernel es que este permite dar un mayor peso a los pixeles más cercanos a la posición donde se desea hacer la predicción. Esto es muy importante porque los pixeles más cercanos son mucho más relevantes que los pixeles lejanos a la hora de predecir el valor de un pixel en cierta posición.

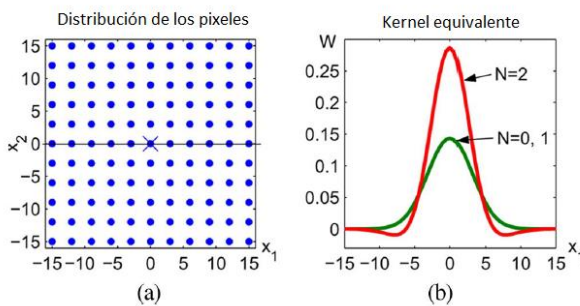


Figura 40: Distribución de los pixeles y Grafico de Kernel gaussiano en 2D.

Fuente: Kernel Regression for Image Processing and Reconstruction

Elaboración: Hiroyuki Takeda, Sina Farsiu, y Peyman Milanfar

El valor de la matriz de suavizado H , que es de 2×2 , es la que controla la forma del Kernel. Tal como se muestra en la Figura 41, es razonable usar Kernels pequeños en áreas de gran densidad de muestras y Kernels Grandes en áreas de poca densidad de muestras. Para mayor detalles ver [17].

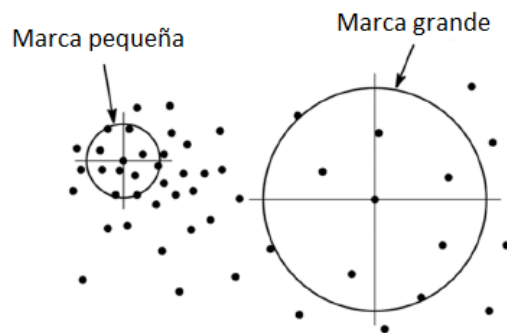


Figura 41: Variación del Kernel en función de la matriz de suavizado.

Fuente: Kernel Regression for Image Processing and Reconstruction

Elaboración: Hiroyuki Takeda, Sina Farsiu, y Peyman Milanfar

4.5.3.3 Técnicas de regresión kernel adaptables en imágenes

Tal como se vio en las ecuaciones 4.3, 4.4 y 4.5 el método de Kernel tradicional solo depende de la distribución espacial de los píxeles, sin considerar el valor que toma cada píxel. Sin embargo, tal como se muestra en [20], se logran muchos mejores resultados usando Kernels que no solo consideren las posiciones de los píxeles sino sus valores radiométricos, es decir el valor de su intensidad. Esta diferencia se puede apreciar claramente en la Figura 42, donde vemos que los Kernels que toman en cuenta la intensidad se pueden adaptar ante la presencia de un flanco.

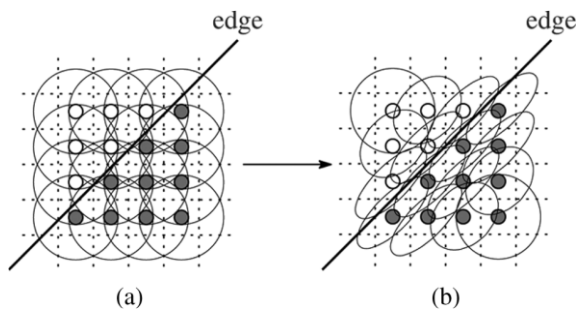


Figura 42: (a) Kernel clásico. (b) Kernel adaptable.

Fuente: Kernel Regression for Image Processing and Reconstruction

Elaboración: Hiroyuki Takeda, Sina Farsiu, y Peyman Milanfar

Estos Kernels se consiguen mediante el uso de matrices de suavizado H que no solo dependen de la distancia sino del valor de los pixeles. La forma de estos Kernels adaptables está dada por:

$$K_{adaptable}(x_i - x, y_i - y) = \frac{\sqrt{\det(C_i)}}{2\pi h^2 u_i^2} \exp\left\{-\frac{(x_i - x)^T C_i (x_i - x)}{2\pi h^2 u_i^2}\right\} \quad (4.10)$$

Donde la matriz C esta basadas en las diferencias de los valores de las intensidades de los pixeles. En la Figura 43, se puede ver la forma de los Kernels adaptables en diferentes regiones de una misma imagen. Para mayor detalle sobre el cálculo de las matrices C y otros, ver [20].

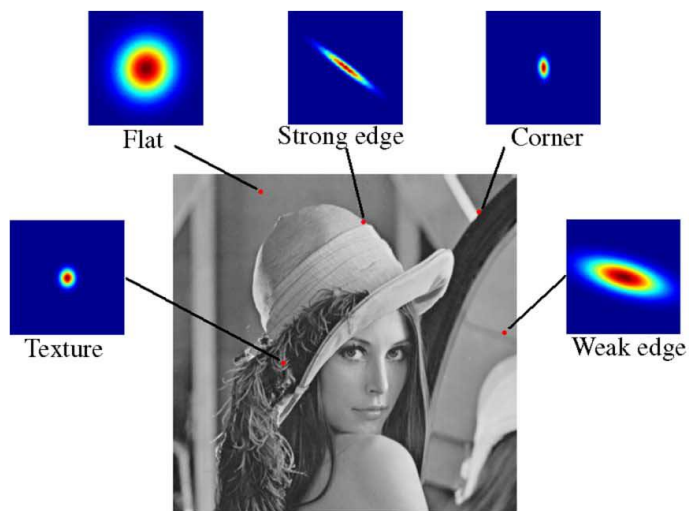


Figura 43: Ejemplos de Kernels adaptables, donde se aprecia diferentes matrices de suavizado según la posición del píxeles

Fuente: Kernel Regression for Image Processing and Reconstruction

Elaboración: Hiroyuki Takeda, Sina Farsiu, y Peyman Milanfar

4.5.4 Diagrama de Flujo

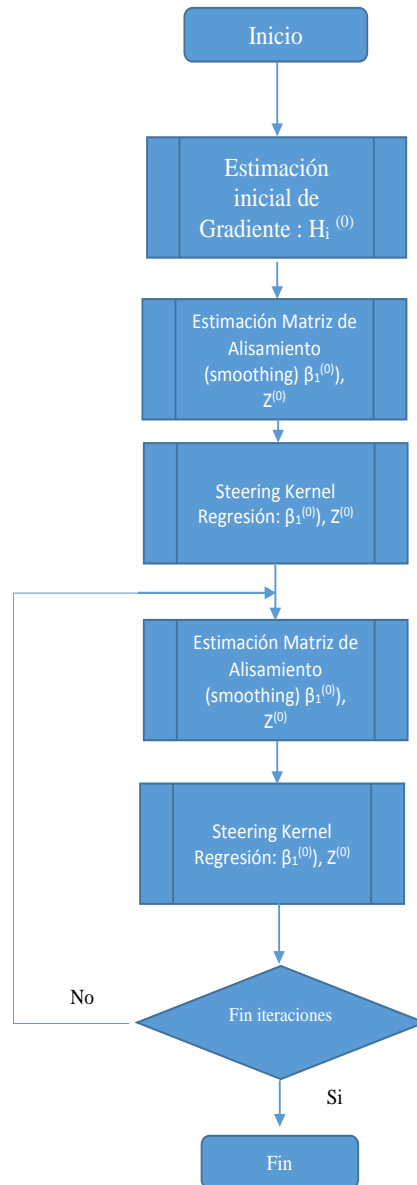


Figura 44: Diagrama de Flujo de Metodo IKS para obtener Súper Resolución a partir de 1 sola imagen

Fuente: Kernel Regression for Image Processing and Reconstruction

Elaboracion: Enzo Camargo Fernández - Baca

El método IKS se realiza en 2 pasos:

1. Determinar el área relevante para cada pixel, basada en la orientación dominante, indicado en Figura 43, mediante la estimación inicial de los gradientes de las imágenes (utilizando un estimador de gradientes, el cual puede ser de segundo orden del método clásico de regresión Kernel).
2. Filtrar la información utilizando la orientación en cada región de píxeles basado en la adaptación dirigida (adaptively “steer”), es decir, según la orientación podría resultar necesario alongar, contornos elípticos, circulares a través de dicha región, esto conlleva a aprovechar la vecindad del pixel (o región de píxeles: imagen) que se está analizando y determinar con mejor aproximación la información y aumento del muestreo correspondiente (upsampling), por ejemplo en bordes entre otros (Figura 45).

Con esta información local de Kernels adaptativos, la eliminación de ruido es mucho más efectiva, a través de los contornos resultando en una mejor preservación de detalles en la salida.

4.5.5 Ejemplos de aplicación de Método IKS

Se aplica el IKS a la Figura 34 b) y se obtiene la Figura 45(b), considerando que en este caso solo se requiere una sola imagen de entrada. Se observa que a la vista del ojo humano se observan bordes mejor definidos y de mejor resolución en (b) que en (a).

4.5.5.1 Ejemplo de imagen fotográfica:

En este caso aplicamos IKS a la imagen de entrada.



(a) Imagen en baja resolución (LR)



(b) Imagen en Súper Resolución (SR) utilizando el método de Kernel Adaptativo

Figura 45: Ejemplos de aplicación de Método de Regresión de Kernel Adaptativo

Fuente: Extracto de Figura 34

Elaboración: Enzo Camargo Fernández - Baca

CAPÍTULO V: EXPERIMENTACIÓN

Para la realización de las experimentaciones, se toma como fuente de datos las imágenes indicadas en el Capítulo 0, en donde se especifican sus atributos como tamaño, tipo de imagen, número de canales e información necesaria de base para desarrollar el presente Capitulo, con la finalidad de probar la hipótesis, considerando los fenómenos y parámetros descritos previamente.

5.1 Pruebas sintéticas

Para la evaluación del desempeño de los métodos de Super Resolución IBP e IKS, se realiza mediante pruebas sintéticas, el cual consiste en un entorno de pruebas controlado que permita medir la imagen en Súper Resolución SR en comparación con la imagen real HR mediante el PSNR y el RMSE.

En la pruebas sintéticas se crea las imágenes de entrada (imágenes de baja resolución) del modelo mostrado en Figura 36(b), a partir de la imagen real (HR) indicado en Figura 36(a). En cada imagen de baja resolución, se simula los fenómenos que afectan la calidad de la imagen como desenfoque, el ruido, Aliasing, entre otros, luego en base a estas imágenes de baja resolución se procesa y se obtiene la imagen en Súper Resolución (SR), finalmente se determina el desempeño de cada método utilizando el indicador objetivo el PSNR.

La finalidad principal de las pruebas sintéticas, en este caso, es comparar la imagen de Súper Resolución obtenida de un método con la imagen original; y esto se realiza mediante el PSNR y el RMSE, y por lo tanto permite comparar métodos distintos basados en los mismos indicadores. Este ambiente controlado de pruebas también permite trabajar con las diferentes variables del modelo, analizando sus efectos, así como para el desarrollo de los modelos de mejora, los cuales se van comparando y evaluando dentro de este escenario. Asimismo las pruebas sintéticas, en general, es un método ampliamente utilizado en el desarrollo e investigación en el campo del procesamiento de imágenes medicas

El PSF se asume el mismo valor, por considerarse que se usa el mismo equipo para capturar las imágenes.

En la Figura 46, se muestra el esquema de cómo se trabajan las pruebas sintéticas y su relación con la realización de una aplicación real. Se utilizan las pruebas sintéticas por la facilidad de medir resultados con el PSNR y RMSE.

Importante:

En el caso de aplicaciones reales, no es posible medir estos indicadores, debido a que la imagen de referencia no existe.

En el siguiente esquema, se muestra la relación entre las pruebas sintéticas y la aplicación de los métodos de Super Resolución en el mundo real.

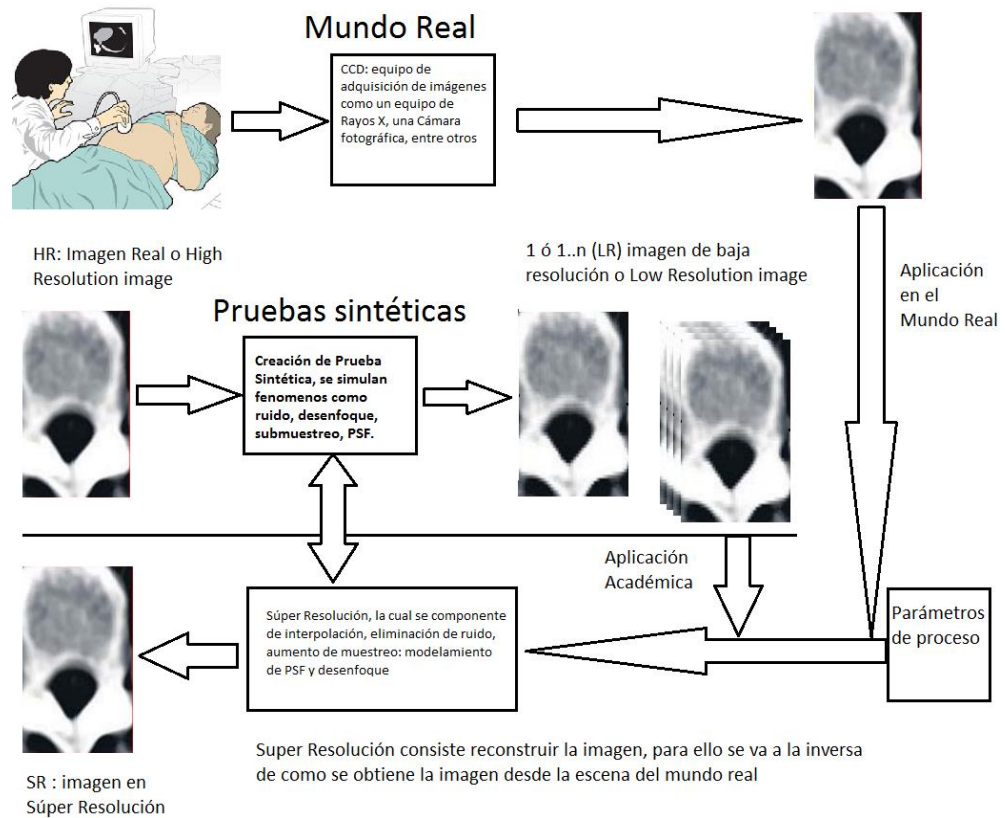
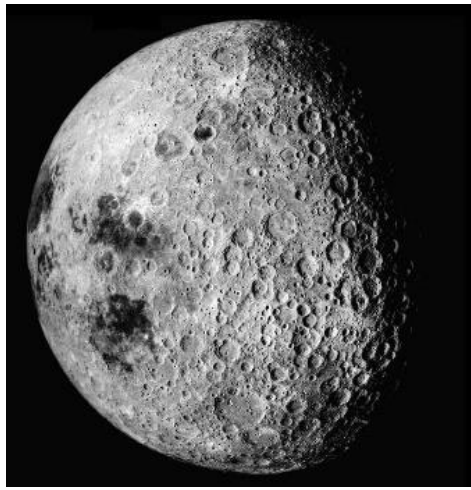


Figura 46: Esquema de las pruebas sintéticas y su relación con una aplicación real

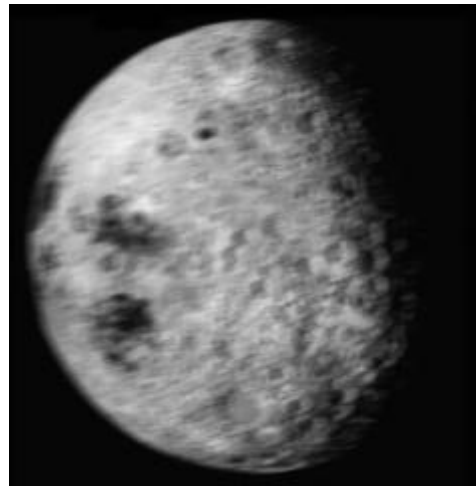
Elaboración: Enzo Camargo Fernández Baca

Fuente: Enzo Camargo Fernández Baca

En las Figura 47 y Figura 48, se muestran cómo se han creado las imágenes LR (b) a partir de la imagen (a).



a. (HR) Imagen Original



b. (LR) Imagen de baja resolución

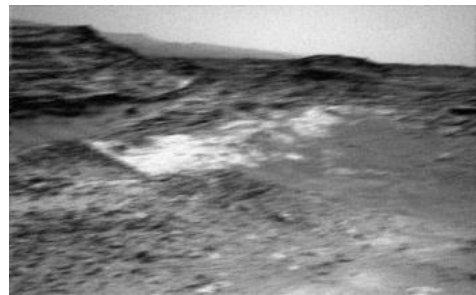
Figura 47: Pruebas sintéticas aplicadas a imagen: Cara oculta de la luna y la creación de la imagen en baja resolución LR

Fuente: Imagen original tomada de NASA [3].

Elaboración: Enzo Camargo Fernández Baca



a. (HR) Imagen Original



b. (LR) Imagen de baja resolución

Figura 48: Imagen capturada por Vehículo de exploración de la NASA, exposición a contacto geológico y la creación de la imagen en baja resolución LR.

Fuente: Imagen original tomada de NASA [3].

Elaboración: Enzo Camargo Fernández Baca

5.2 Interpolación de una imagen con varios métodos

En la Figura 49 y Figura 50, son ejemplos donde se presentan resultados utilizando métodos como la Interpolación bicúbica, Bilineal, Nearest y Lanczos3, adicionalmente a IBP e IKS.

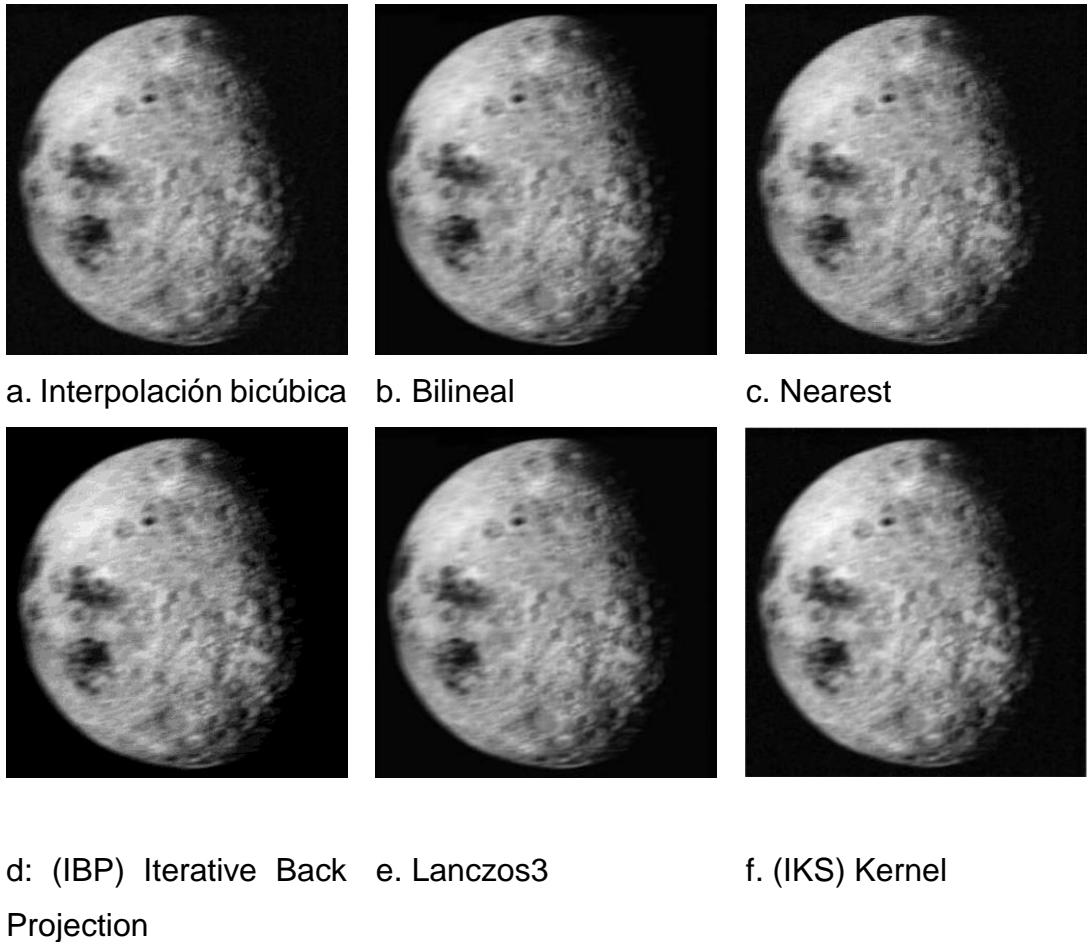


Figura 49: Basados en la figura 47, se han aplicado diversos métodos para la mejora de las imágenes (a),(b), (c) y (e) obtenidas mediante interpolaciones respectivas; (d) y (f) son el resultado de métodos de súper resolución.

Fuente: Imagen original tomada de NASA [3].

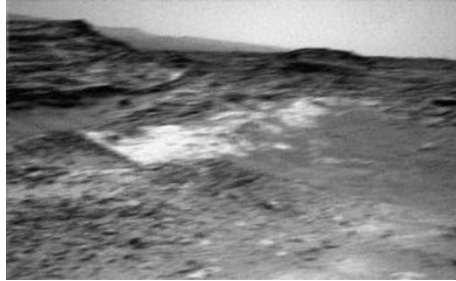
Elaboración: Enzo Camargo Fernández Baca

Estos métodos son de interpolación, que a diferencia de un método de súper resolución solo aumentan el muestreo usualmente no eliminan el ruido existente.

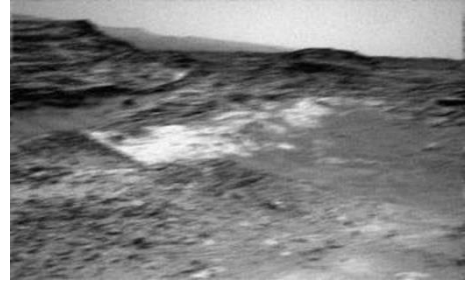
Método	PSNR
Interpolación Bicúbica	29.8812 dB
Bilineal	30.8036 dB
Nearest	29.8927 dB
Iterative Back Projection (IBP)	30.4831 dB
Lanczos3	29.8594 dB
Kernel (IKS)	26.8642 dB

Tabla 8: Valores PSNR para la aplicación de métodos de interpolación y súper resolución a la imagen LR de la Figura 47 (b) de la cara oculta de la luna.

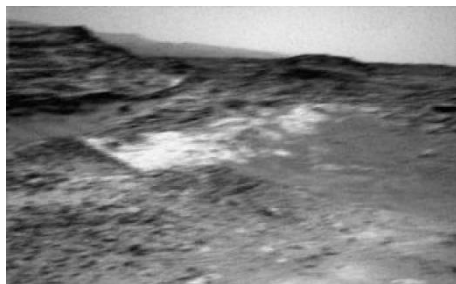
Elaboración: Enzo Camargo Fernández Baca



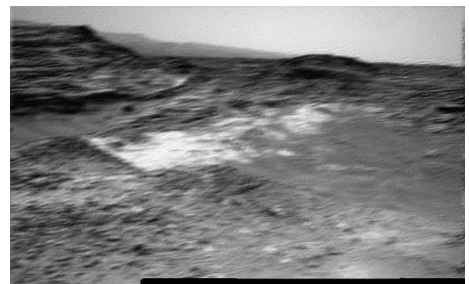
a. Interpolación bicúbica



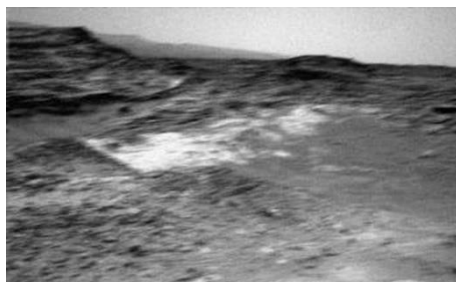
b. Bilineal



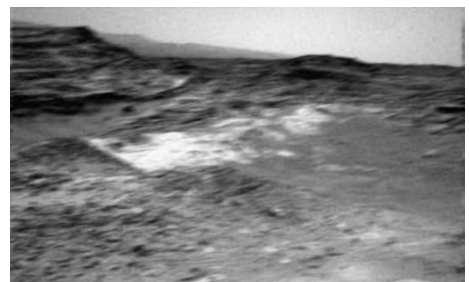
c. Nearest



d: Iterative Back Projection (IBP)



e. Lanczos3



f. Kernel (IKS)

Figura 50: Mejora de la imagen de la Figura 48 aplicando 6 métodos interpolación y Súper Resolución.

Fuente: basado en Figura 48

Elaboración: Enzo Camargo Fernández Baca

Método	PSNR
Interpolación Bicúbica	29.7887 dB
Bilineal	29.7600 dB
Nearest	29.8270 dB
Iterative Back Projection (IBP)	29.9841 dB
Lanczos3	29.7031 dB
Iterative Kernel Steering(IKS)	29.8015dB

Tabla 9: Tabulación de resultados de los diferentes métodos de SR
Elaboración: Enzo Camargo Fernández Baca

5.3 Eliminar el ruido de las Imágenes (Image Denoising)

En este primer conjunto de experimentos y presentación de los resultados, utilizamos las imágenes indicadas en la fuente de datos del Capítulo 0, obtenidas del Hospital Cayetano Heredia, lo que se compara es la performance de ambos métodos para eliminar solo el ruido. Para ello se adiciona ruido blanco Gaussiano con una desviación estándar $\sigma = 25$, según [30] a la imagen de la **Tabla 7-1**, la cual se reproduce en la Figura 51(a), luego el ruido de la imagen se elimina utilizando los 2 métodos en evaluación, iniciamos aplicando el IKS con $h = 2.4$ como parámetro global de suavizado, 15 iteraciones, $\lambda = 1$; parámetro de regularización de elongación, $\alpha = 0.5$; como parámetro de estructura sensitiva.

Asimismo los resultados obtenidos en la Figura 51, se muestran en acercamiento en la Figura 52, con la finalidad de apreciar mejor visualmente las diferencias.

En el análisis se puede apreciar que se obtiene el mejor resultado utilizando el Iterative Kernel Steering que con el IBP, esto también se comprueba con el

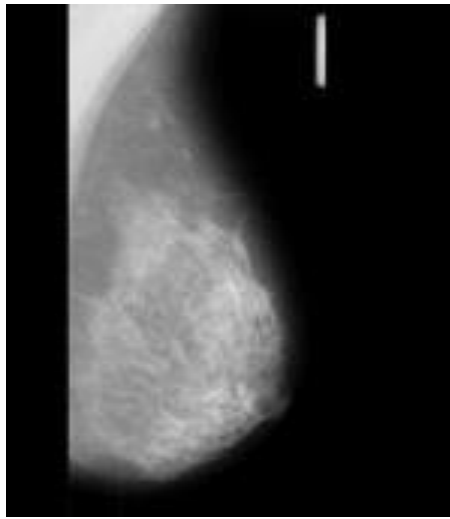
PSNR (IKS) > PSNR (IP), así con el RMSE (IKS) < RMSE (IBP). Este hallazgo es muy importante, lo cual usaremos más adelante.

5.4 Eliminar el ruido combinando métodos

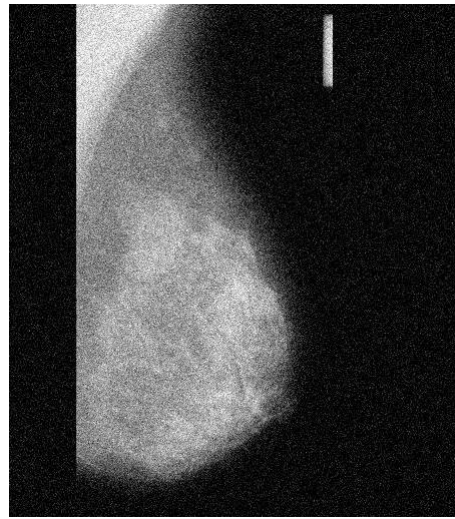
En esta experimentación, planteamos eliminar el ruido de una imagen con la combinación los 2 métodos, entonces para ello, se plantea el problema:

Se sub muestrea la imagen Figura 51 a, 3 veces menor para obtener una imagen en LR. A esta imagen en LR, le aplicamos ruido blanco gaussiano similar al experimentación anterior y se obtiene Figura 53 a) y a esta imagen le aplicamos ahora los 2 métodos, el IKS para eliminar el ruido y obtenemos Figura 53 b) y luego a esta imagen le aplicamos el IBP y obtenemos la Figura 53 d).

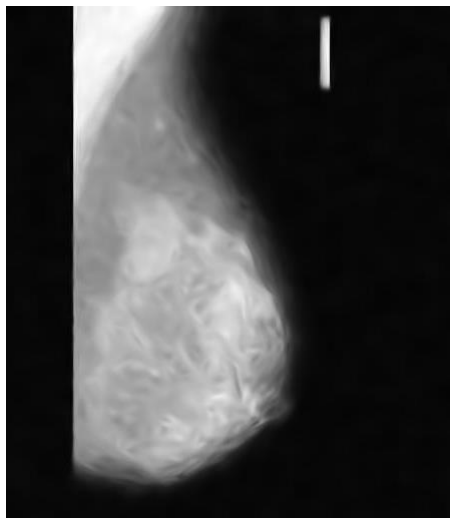
Como vemos en los resultados, se ha obtenido una mejora sustancial en la eliminación de ruido en Figura 53 d) que combina los 2 métodos respecto a aplicación de IBP solamente Figura 53 c) , esto es debido a que los métodos son independientes entre sí. Aunque esto todavía no indica si el método de Kernel es mejor que el IBP para la Súper Resolución, si indica que el IKS es más eficiente en eliminar el ruido.



(a) HR



(b) (LR) Imagen con ruido, $N=0$, no iteraciones, $STD = 25$, $RMSE = 20.3877$, $PSNR = 21.6118$



(c) IKS se aplica a imagen (a), $N=2$, 12 iteraciones, $RMSE = 8.7069$, $PSNR = 27.2573$

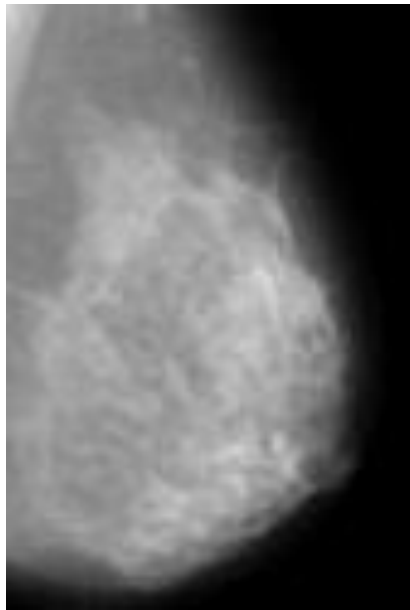


(d) IBP se aplica a imagen (a), $RMSE = 21.9520$, $PSNR = 21.3013$

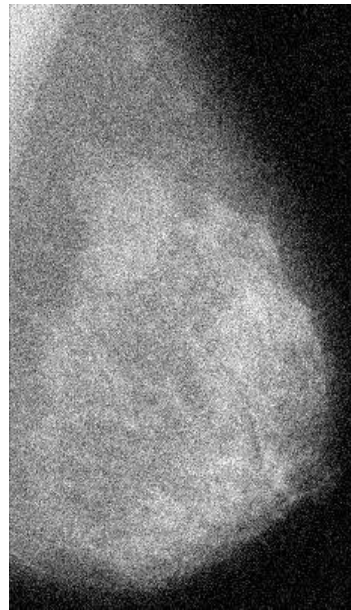
Figura 51: Performance de los métodos para eliminar el ruido en imágenes médicas de mamografía. RMSE de imágenes (b) – (f) es respectivamente y el PSNR

Fuente: Hospital Cayetano Heredia

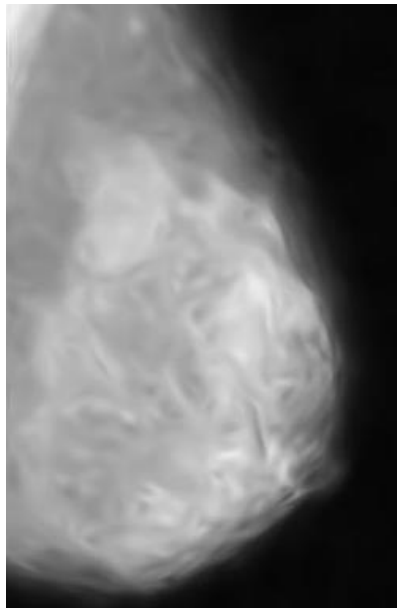
Elaborado por Enzo Camargo Fernández - Baca



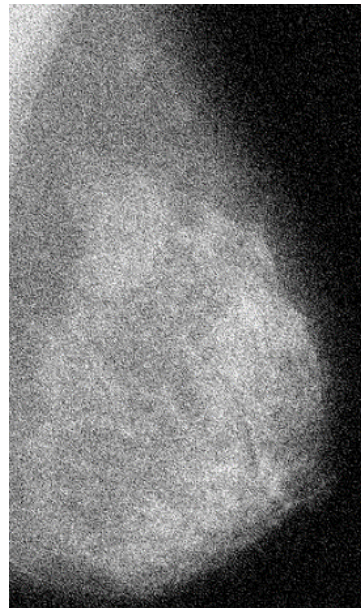
(a)



(b)

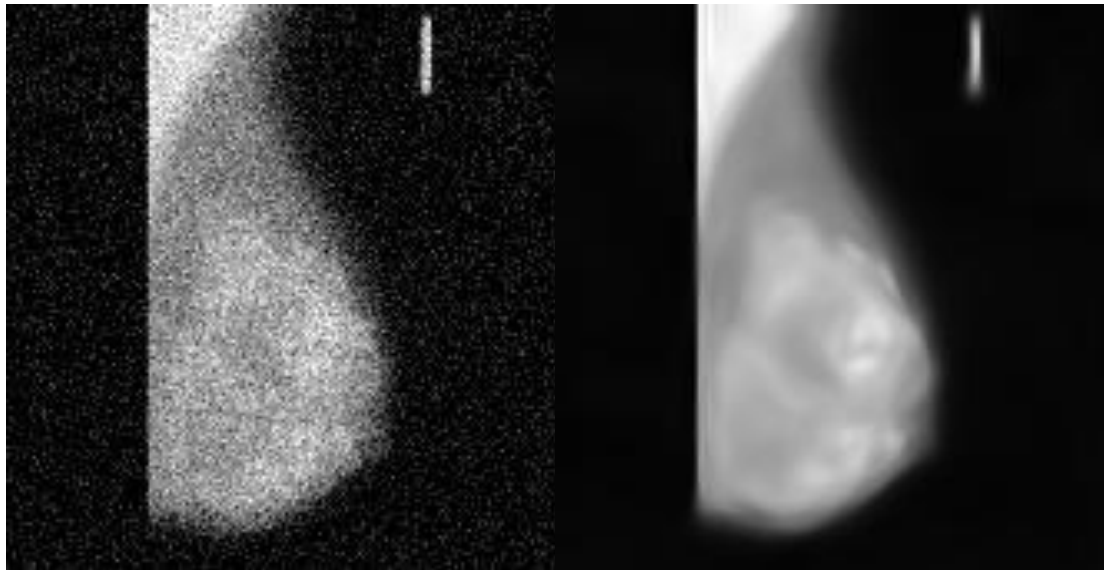


(c)



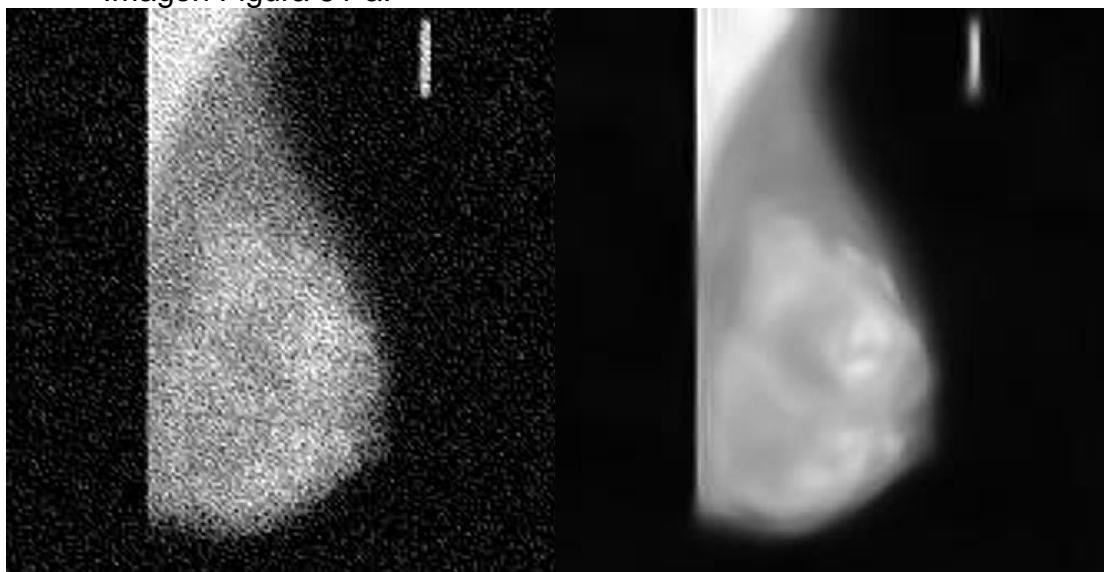
(d)

Figura 52: Acercamiento a Figuras 51 (a), (b), (c) y (d) respectivamente
Elaborado por Enzo Camargo Fernández – Baca



(a) LR creada con ruido ($\sigma = 25$), $r = 3$ (nro de veces de submuestreo) aplicado a Imagen Figura 51-a.

(b) IKS aplicada a LR



(c) IBP a LR
PSNR = 21.3013, RMSE = 21.9520

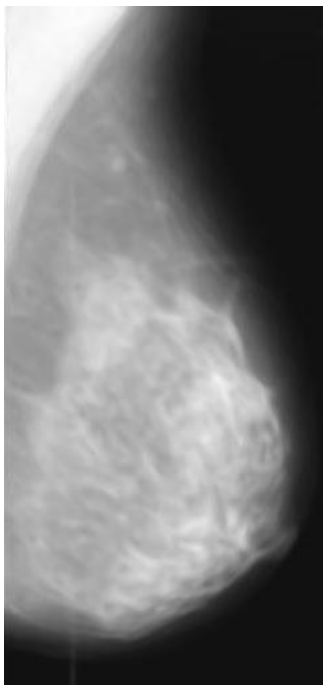
(d) IBP a imagen (b).
PSNR = 26.6345, RMSE = 11.8800

Figura 53: LR se obtiene submuestreando HR. Combinación de métodos, se aprecia una mejora en lo obtenido en (d) respecto a lo obtenido en (c).

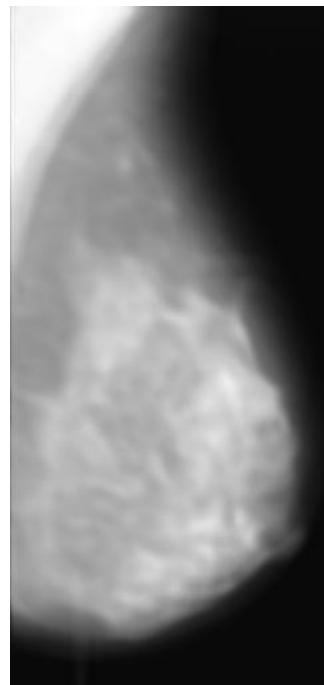
Elaborado por Enzo Camargo Fernández - Baca

5.5 Interpolación de imágenes

En esta tercera experimentación, se sub muestrea la imagen Figura 51 a), en 3 veces en cada dirección (X e Y), se compara la técnica de Iterative Kernel Steering con parámetros diferentes, la primera con $N=0$, $h=1$ y no iteraciones, y la Figura 54 b) se obtiene considerando los parámetros $N=0$, $h=2.4$ y no iteraciones.



(a) IKS a Figura 51(a), parámetros $N=0$, $h=1$, no iteraciones.



(b) IKS a Figura 51(a), parámetros $N=0$, $h=2.4$, no iteraciones.

PSNR= 25.7372, RMSE= 19.3581 PSNR=23.4408, RMSE = 14.8607

Figura 54: Interpolación a la mamografía, en este caso, solo se le hace sub muestreo en 3 veces, luego se aplica IKS según los parámetros indicados en (a) y (b)

Elaborado por Enzo Camargo Fernández - Baca

En los resultados, se puede observar que la Figura 54(a) (PSNR y MSE que es 25.7372, 19.3581), es visualmente mejor aunque de forma ligera, que la Figura 54(b) (PSNR y MSE = 23.4408, 14.8607) y esto se comprueba los valores de PSNR y MSE respectivamente.

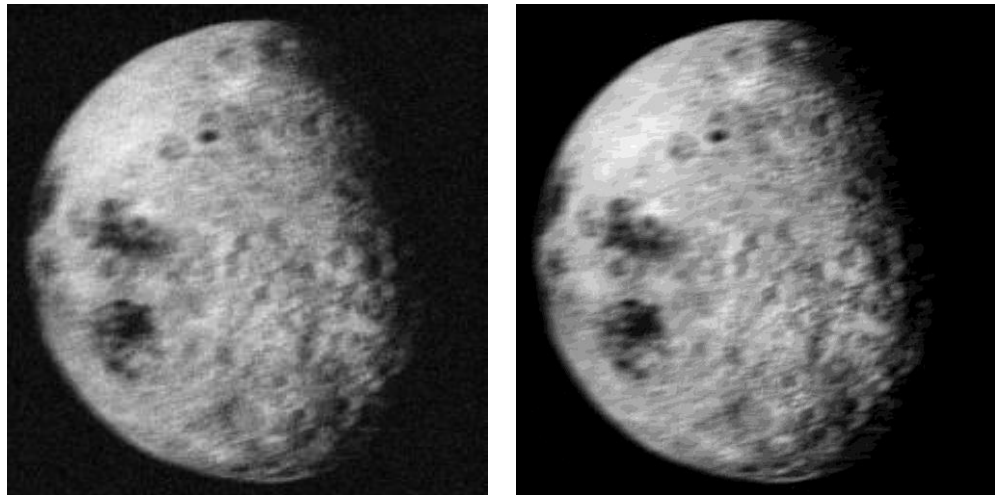
5.6 Súper Resolución con sub muestreo y ruido

En la cuarta experimentación que se muestra en la Figura 55, se ha añadido ruido y se sub muestrea la imagen HR de la Figura 47 (a), obteniéndose la imagen en LR de la Figura 55 (a), en estas experimentaciones se utiliza los mismos parámetros y se varía las imágenes de aplicación, considerando las imágenes satelitales. La imagen original (HR) se le aplica las pruebas sintéticas, simulando los fenómenos como sub muestreo y ruido.

Entonces se evalúa el desempeño de ambos métodos para minimizar el ruido, aumentar muestreo (o upsampling, en idioma inglés). La prueba sintética se inicia adicionando ruido blanco Gaussiano con $\sigma^2 = \{0.0001 \dots 0.0016\}$; y $\mu = 0$, donde σ^2 es la varianza y μ es la media, y se obtiene la imagen en LR. Se aplica el Iterative Kernel Steering con $h= 0.75$, $N=0$, no iteraciones, $\lambda = 1$; parámetro de regularización de elongación, $\alpha = 0.05$; como parámetro de estructura sensitiva, $r= 3$ como factor de aumento de muestreo.

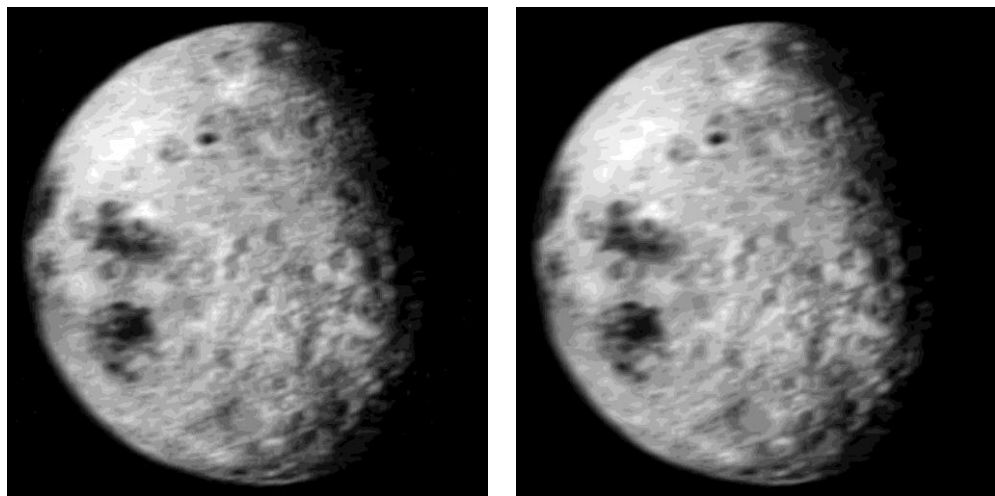
Analizando los resultados, se puede apreciar que el resultado utilizando el IBP que con el Iterative Kernel Steering, es muy similar, esto también se comprueba con el PSNR (IBP) \sim PSNR (IBP), así con el RMSE (IKS) \sim RMSE (IBP) en el caso de las Figura 55(b) y (d) respectivamente. En este caso se presenta un resultado que se debe resaltar, en este caso que el ruido no es alto (se ha simulado un ruido bajo), los resultados con el IBP ahora son

similares a el IKS. Esto demuestra adicionalmente a lo indicado en 5.3 que cuando se tienen imágenes con ruido, es conveniente el IKS que el IBP.



(a) LR

(b) IBP aplicado a LR



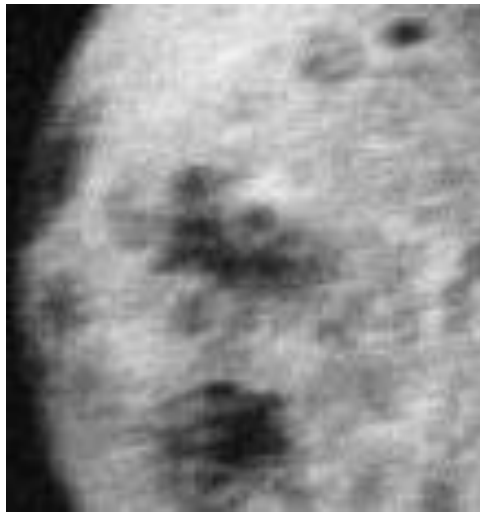
(c) IKS a LR, con $N=0$, $h=0.75$, no iteraciones. (d) IKS, con $N=0$, $h=1$, no iteraciones.

Figura 55: Súper Resolución con sub muestreo y ruido, a la cara oculta de la luna, con los 2 métodos IBP e IKS.

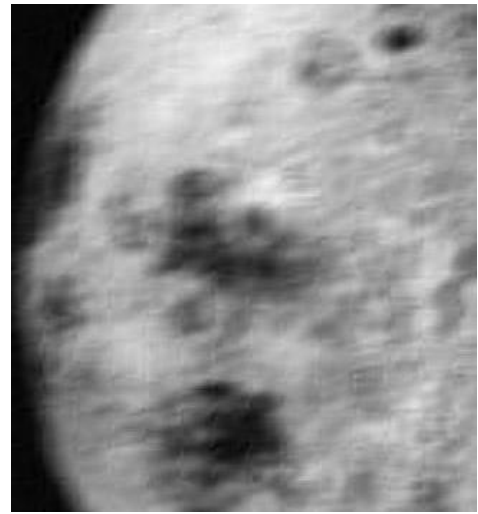
De las figuras (b)-(d) se obtiene PSNR = 18.6977, 8.4320, 18.5354 respectivamente; RMSE = 29.6246, 96.5919, 30.1836 respectivamente

Fuente: Pruebas basadas en Figura 47 (a).

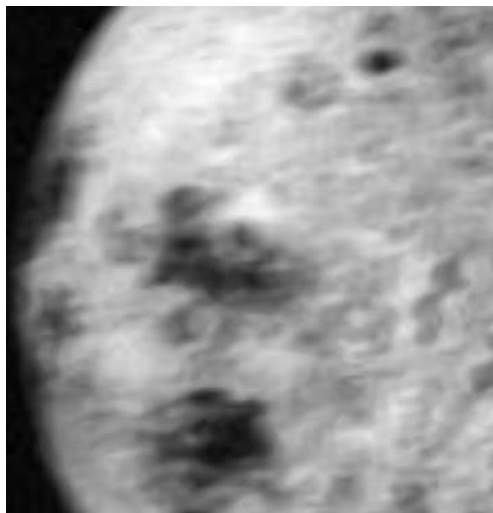
Elaboración: Enzo Camargo Fernández Baca



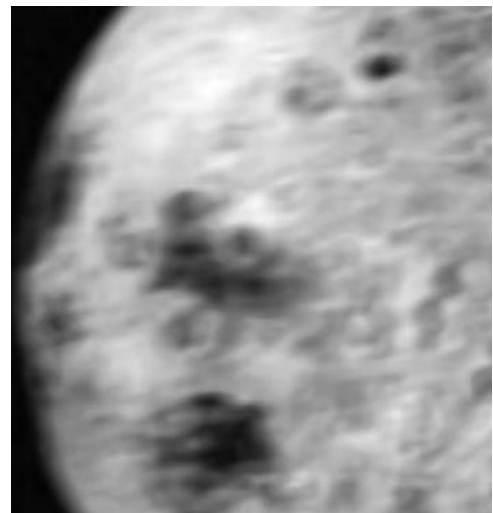
(a) Imagen en LR



(b) IBP



(c) IKS , con $N= 0$, $h = 0.75$, no iteraciones



(d) IKS, con $N= 0$, $h = 1$, no iteraciones.

Figura 56: Acercamiento a Imágenes de la Luna en LR y en SR aplicando Iterative Back Projecting e Iterative kernel Steering

Fuente: Acercamiento a Figura 55

Elaboración: Enzo Camargo Fernández Baca

5.7 Súper Resolución adicionando el PSF

En esta quinta experimentación, se adiciona el PSF además de los otros fenómenos simulados en la cuarta experimentación, para crear las pruebas sintéticas.

Se utiliza como imagen de entrada la indicada en Tabla 7-5, la creación de la prueba sintética Se inicia adicionando ruido blanco Gaussiano con $\sigma^2 = \{0.0001 \dots 0.0016\}$; $\mu = 0$, donde σ^2 es la varianza y μ es la media, y se obtiene la imagen en LR. La Figura 56 c) y d) se obtiene aplicando el Iterative Kernel Steering con $h = 0.75$, $N=0$, no iteraciones; y con los parámetros $N=2$, $h=2.4$, 15 iteraciones respectivamente, y para ambos casos: $\lambda = 1$; parámetro de regularización de elongación, $\alpha = 0.05$; como parámetro de estructura sensitiva, $r = 3$ como factor de aumento de muestreo.

Asimismo se obtiene la imagen Figura 56 b) aplicando el Iterative Back Projecting.

Según los resultados, se puede observar que se tiene ligeramente mejores resultados utilizando el método IKS que el IBP, comprobado por los resultados obtenidos mediante el $PSNR(IKS) > PSNR (IBP)$ y $RMSE (IKS) < RMSE (IBP)$

5.7.1 Ejecución de Iterative Back Projecting y resultados de corrida

Para ejecutar esta rutina, debe seguir 2 pasos:

Paso 1: Localizar ruta de ejecución y copiar archivos de entrada

En este caso es: D:\SR\Iterative Back Projecting>

En esta ruta se copia los archivos de trabajo, que son los archivos LR, en el caso que K=4, es decir se tienen 4 imágenes de entrada se copia:

Paso 2: ejecución

D:\SR\Iterative Back Projecting> iterative_back_projecting_gpu.exe

Obteniéndose lo siguiente.

```
%*****  
1 main  
2 main  
3 main  
Ingresando las imagenes del registro  
2  
PARAMETROS DE ROTACION:  
Angulo[0] : 0.0000000000000000  
Angulo[1] : 10.200001597404482  
Angulo[2] : 1.800001263618471  
Angulo[3] : -29.999999999999996  
3  
  
PARAMETROS DE DESPLAZAMIENTO:  
x: 0.0000000000000000 , y: 0.0000000000000000  
x: -0.015383898775746 , y: 0.089494212633246  
x: 0.281609426688878 , y: -0.171323457126121  
x: -0.401502760748156 , y: -0.052155118945607  
4  
  
ITERATIVE BACK PROJECTING:
```

```
5
6

Delta: 0.014743
Delta: 0.011053
Delta: 0.008350
Delta: 0.006346
Delta: 0.004964
Delta: 0.004158
Delta: 0.003921
Delta: 0.003895
Delta: 0.003923
Delta: 0.003894 7

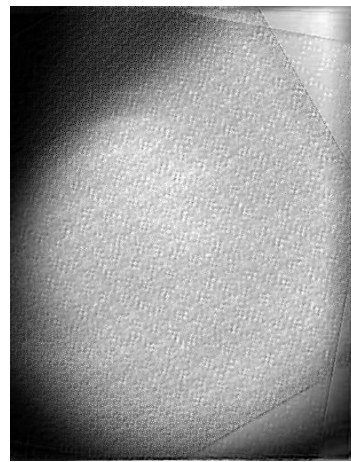
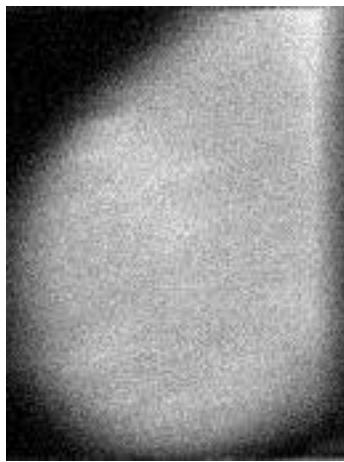
4 main
5 main

%*****
```

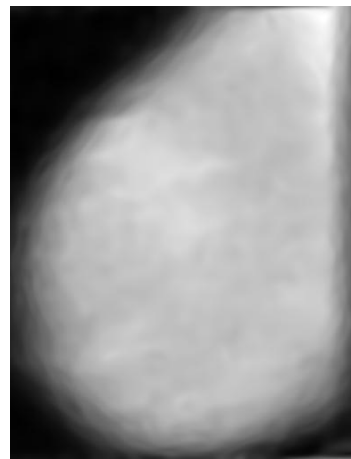
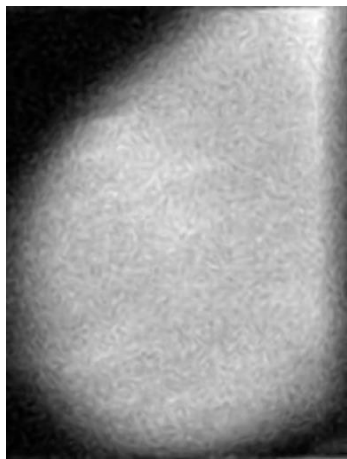
Los resultados del programa mostrados se describen a continuación:

1. Leer imágenes de entrada, en este caso son 4 imágenes de entrada, creadas mediante pruebas sintéticas
2. Realiza la registración o alineamiento de las imágenes a un mismo sistema de coordenadas, tomando la imagen 1, como la imagen de referencia
3. Se calcula los ángulos de rotación $\text{Angulo}[0]$ hasta $\text{Angulo}[3]$ para cada imagen. Como la primera imagen es la referencia, se tiene $\text{Angulo}[0] = 0.0000$
4. De la misma forma, calcula el desplazamiento para cada imagen. Para el caso de la primera imagen es la imagen de referencia, se tiene $(X, Y) = (0.0000, 0.0000)$

5. Luego de ello se realiza el proceso específico del algoritmo teniendo las imágenes registradas, lo cual es explicado en
6. En cada iteración (Iterative), se tiene diferencias cada vez menores lo que denota que la imagen resultante para cada iteración es mejor que la anterior, esto se realiza hasta un número de iteraciones determinado, en este caso son 10 iteraciones.
7. Finalmente se obtiene la imagen en SR con el método IBP.



a) Imagen LR obtenida de Tabla 7-5 (b) IBP



(c) IKS aplicado a LR, con $N=0$, $h=1$, no iteraciones (d) IKS aplicado a imagen (c), con $N=2$, $h=2.4$, 15 iteraciones

Figura 57: Se toma la imagen de Tabla 7-5 y se obtiene LR, de las figuras (b)-(d) se obtiene PSNR = 17.1520, 18.3677, 17.6071 respectivamente; RMSE = 35.3950, 30.7718, 33.5882 respectivamente

Fuente: Hospital Nacional Cayetano Heredia [7]

Elaboración: Enzo Camargo Fernández Baca

5.8 Súper Resolución con PSF, sub muestreo y ruido

En este sexto conjunto de experimentación, se crean 2 tipos de prueba sintéticas, se incluye el PSF y el sub muestreo, en una se adiciona ruido y en otra no, esto se aplican a 2 imágenes de la Tabla 7-2 y Tabla 7-3.

El ruido adicionando es ruido blanco Gaussiano con $\sigma^2 = \{0.0001 \dots 0.0016\}$; $\mu = 0$, donde σ^2 es la varianza y μ es la media.

Vamos a comparar la performance de ambos métodos, para estas 2 pruebas, entonces:

Primera prueba sintética:

1. Adicionando el sub muestreo
2. Se aplica una función de Gauss para simular el PSF a la imagen obtenida en 1 y se obtiene LR sin considerar el ruido obteniéndose una conjunto de imágenes en LR (1..16).
3. Se toma LR y se aplica IKS con $N=0$, $h=1$, no iteraciones,
4. Se procesa con el IBP

Segunda prueba sintética:

1. Se adiciona el ruido blanco Gaussiano con los parámetros indicados previamente,
2. Se sub muestrea la imagen obtenida en 1.
3. Se adiciona el desenfoque a la imagen obtenida en 2, y se obtiene LR

4. Se toma LR y se aplica IKS con $N=0$, $h=1$, no iteraciones,
5. Se procesa con el IBP

Se observa lo siguiente:

Sin ruido:

Figura 58 (b) – (c): PSNR= 27.8491, 30.1967 RMSE = 10.3296, 7.8833 respectivamente

Figura 59 (b) – (c): PSNR= 29.5583, 29.5211 RMSE = 8.4845, 8.5209 respectivamente

Resultado: PSNR (IBP) > PSNR (IKS)

Con ruido:

Figura 58(e) – (f): PSNR=22.5077 y 22.0858, RMSE=19.1054 y 20.0563 respectivamente

Figura 59(e) – (f): PSNR=20.0778, 18.5905, RMSE=25.2726 y 29.9928 respectivamente

Resultado: PSNR (IBP) < PSNR (IKS)

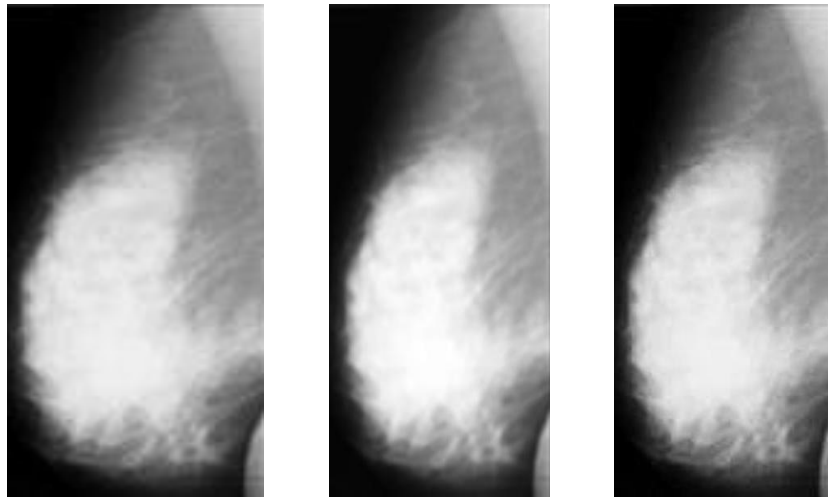
Asimismo, se observa que ante la presencia del ruido el Iterative Kernel Steering presenta mejor performance que el método IBP, lo cual es el mismo resultado en experimentaciones previas.

En cambio cuando no hay ruido el IBP es mejor o ligeramente mejor que el IKS.

Considerando que se aplica este método IKS con $N=0$, es decir un nivel de interpolación lineal, y no con $N=2$ se tiene una interpolación de mejor precisión mostrado en la Figura 51 (c), la cual da mejores resultados que Figura 51 (b), esto debido a que este último solo está implementado cuando $r=1$, es decir, se utiliza básicamente para eliminar ruido y todavía no ha sido implementado para aumentar muestreo, razón por la cual no es aplicada en la presente experimentación.

Se observa en los resultados de **Figura 58 (b) - (c)** y **Figura 58 (e) y (f)**, visualmente que las imágenes se suavizan luego de aplicarles los métodos de súper resolución, pese a tener ruido o no, independientemente de la imagen LR de entrada.

Los resultados de la **Figura 59**, son similares a lo obtenido en la **Figura 58**.



(a) LR (PSF + sub
muestreo)

(b) IKS (SR), N= 0,
h=1, no iteraciones

(c)
IBP (SR)



(d) LR (ruido + PSF +
sub muestreo)

(e) IKS (SR), N= 0,
h=1, no iteraciones

(f) IBP (SR)

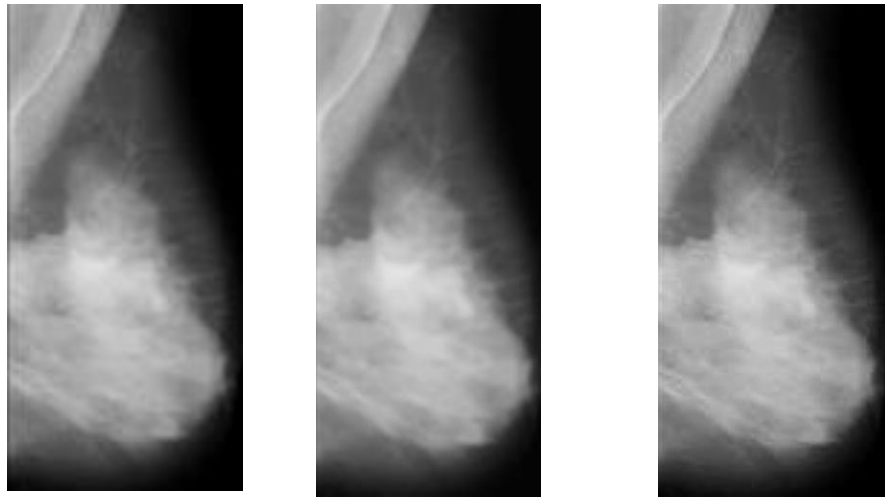
Figura 58: Aplicación con ruido y sin ruido, adicionando sub muestreo y PSF.

Se obtiene (b)-(c) PSNR= 27.8491, 30.1967 RMSE = 10.3296, 7.8833 (e) – (f)

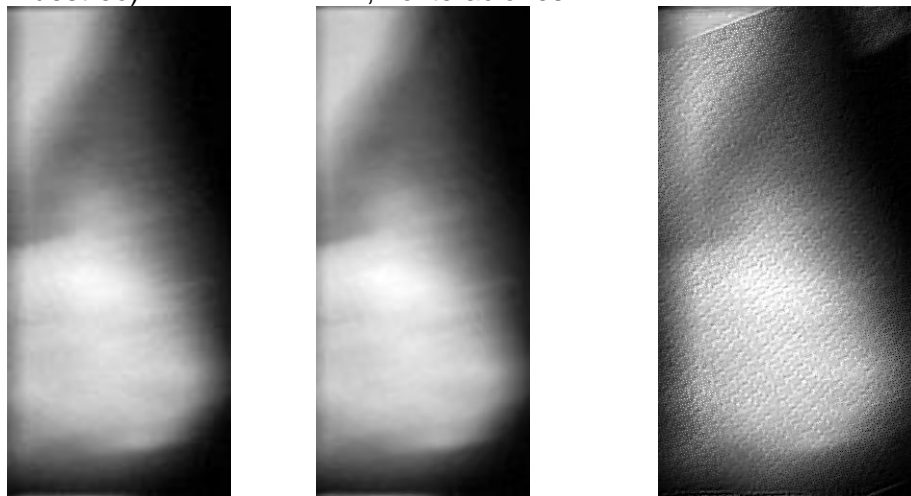
PSNR= 22.5077 y 22.0858, RMSE= 19.1054 y 20.0563 respectivamente

Fuente: Hospital Nacional Cayetano Heredia [7]

Elaboración: Enzo Camargo Fernández Baca



(a) LR (PSF + sub muestreo) (b) IKS (SR), N= 0, h=1, no iteraciones (c) IBP (SR)



(a) LR (PSF + sub muestreo) (b) IKS (SR), N= 0, h=1, no iteraciones (c) IBP (SR)

Figura 59: Aplicación con ruido y sin ruido, adicionando sub muestreo y PSF. Se obtiene (b)-(c) PSNR= 29.5583, 29.5211 RMSE = 8.4845, 8.5209 (e) – (f) PSNR= 20.0778, 18.5905, RMSE=25.2726 y 29.9928 respectivamente

Fuente: Hospital Nacional Cayetano Heredia [7]

Elaboración: Enzo Camargo Fernández Baca

5.9 Resumen de resultados obtenidos

La forma objetiva de medir la calidad de una imagen es calculando el RMSE y PSNR entre dos imágenes, explicado en la ecuación 2.4 y 2.5. Este método requiere de una imagen original de referencia, por ello se realizan pruebas sintéticas para cada experimentación como se ha mostrado previamente.

En la Tabla 10, se resume el resultado de las pruebas sintéticas realizadas.

Se observa en todos los casos en los que la imagen es visualmente mejor, el PSNR tiene el valor más alto que el PSNR de la imagen comparada, lo cual comprueba que es un indicador objetivo confiable, esto también es ampliamente tratado en [13] y [15].

También se observa en los casos que los valores de PSNR son aproximadamente iguales, la diferencia entre las imágenes comparadas no es perceptible visualmente.

Nro. Experimento	Descripción	Ref.	Parámetros	Iterative Kernel		Iterative Back	
				Steering (IKS)	PSNR	RMSE	Projecting (IBP)
1	Eliminación sólo ruido en una imagen	Figura 51	ruido blanco Gaussiano con $r \sigma=25$	27.2573	8.7069	21.3013	21.952
				1x		3x	
2	Eliminación sólo ruido Combinación de métodos	Figura 53, 3X	N.A.	26.6345	11.88	21.3013	21.952
3	Interpolación de imágenes	Figura 54, 3X	h=1, N=0	25.7372	19.3581	N.A.	
			h=2.4, N=0	23.4408	14.8607		
4	Súper Resolución con sub muestreo y ruido	Figura 55, 3x	h=0.75, N=0	8.432	96.5919	18.6977	29.6246
			h=1, N=0	18.5354	30.1836	N.A.	
5	Súper Resolución adicionando el PSF	Figura 57, 3x	h=1, N=0	18.3677	30.7718	17.152	35.395
			h=2.4, N=2	17.6071	33.5882	N.A.	
6	Súper Resolución con PSF y sub muestreo (Figura 58)	Sin ruido, 3x	h=1, N=0	27.8491	10.3296	30.1967	7.8833
		Con ruido, 3x	h=1, N=0	22.5077	19.1054	22.086	20.0563

Tabla 10: Resumen de resultados obtenidos basados en experimentaciones realizadas con ambos métodos de Súper Resolución IBP e IKS

Fuente: Propia

Elaboración: Enzo Camargo Fernández Baca

Leyenda: N.A. significa que No aplica.

5.10 Contrastación de la Hipótesis

Considerando la Figura 51, Figura 54 y Figura 57 y los resultados consolidados en la Tabla 10, se observa que la hipótesis planteada es VERDADERA, dado que en todos estos casos $PSNR (IBP) > PSNR (IKS)$, asimismo esto se comprueba visualmente debido a que la imagen obtenida con el método IBP es mejor que lo obtenido con el método IKS. En estos casos no se ha adicionado ruido o el ruido ha adicionado un muy bajo (a la imagen sintética para hacer las pruebas).

Cuando se tiene ruido, adicionalmente a los otros fenómenos que afectan la calidad de una imagen, como se indica observa en la Figura 51, Figura 53 y Figura 57 y la Tabla 10, se concluye que la hipótesis es FALSA ya que $PSNR (IBP) < PSNR (IKS)$, y esto se comprueba también visualmente.

En la Figura 54 y Figura 57 y Tabla 10, se observa que la hipótesis también es FALSA, es decir, la calidad de la imagen resultante en ambos métodos tienen diferencias que no son perceptibles visualmente y esto se manifiesta con los valores de los PSNR aproximados entre sí en ambos casos.

CONCLUSIONES

1. La hipótesis indicada que establece $PSNR(\text{Iterative Back Projecting}) > PSNR(\text{Iterative Kernel Steering})$ es verdadera, cuando la prueba sintética no considera ruido a la imagen o imágenes de entrada. En el caso que la prueba sintética considera ruido, entonces la hipótesis planteada es FALSA, debido a que el método IBP presenta limitaciones para minimizar ruido así como el método IKS tiene un eficiente rendimiento para minimizar el ruido resultando que $PSNR(\text{IBP}) < PSNR(\text{IKS})$.
2. Para la aplicación en imágenes médicas en el diagnóstico médico, deben realizarse pruebas a nivel clínicas, según los exigentes protocolos médicos. Considerar que es necesario evaluar la aparición de falsos negativos y falsos positivos de patologías. Este trabajo representa un primer paso en el objetivo de desarrollar un método alternativo adicional para mejorar la resolución de las imágenes médicas en la etapa de post procesamiento.
3. Se cumplen los objetivos planteados al demostrarse para cada método, que es posible minimizar los fenómenos que afectan la calidad de las imágenes como el ruido, el desenfoque, entre otros, los cuales pueden ser optimizados de forma independiente o de forma conjunta, y en orden indistinto, dependiendo del método de Súper Resolución utilizado.
4. Ambos métodos de Súper Resolución son aplicables a cualquier a todo tipo de imagen, ya que se enfocan en solucionar principalmente el problema por cada pixel o región de pixeles y no se enfocan en la imagen en sí misma. Es por ello que un método puede ser aplicable a medicina o a imágenes satelitales de forma indistinta, con la debida configuración de parámetros previa, que se optimiza para cada caso.

5. El método de Súper Resolución IKS es flexible ya que permite pasarle parámetros a través de los cuales se puede incidir de forma personalizable a resolver la mejora de la resolución de imágenes, o solo resolver el ruido.
6. Los métodos de Súper Resolución evolucionaron a identificar una estrategia de mejora, a partir de los problemas de la imagen y según ello, aplicar los métodos correspondientes basados en el pixel, el entorno del pixel y los patrones de la forma de la imagen a recuperar. Asimismo podrían mejorar la imagen dinámicamente conforme se realice el aumento. El progreso de estos métodos avanza a medida que evoluciona las matemáticas, la óptica, física, el procesamiento de imágenes y la capacidad computacional de procesamiento, siendo el enfoque principal el recuperar la información de la imagen como lo hace el cerebro humano.
7. En todos los casos que la imagen obtenida es visualmente de mejor resolución, el valor del PSNR es más alto, esto comprueba que el PSNR es un indicador efectivo para la evaluación del desempeño de métodos en el campo del procesamiento de imágenes. En los que no hay una diferencia visual importante se tienen PSNR similares. El cálculo del PSNR solo puede ser realizado, cuando se realizan pruebas sintéticas.
8. Se demuestra que es posible mejorar la calidad de una imagen médica hasta en 3X con los modelos propuestos, evaluándose cada método de forma objetiva mediante el PSNR y visualmente.
9. La Súper Resolución, como parte del procesamiento de imágenes, proporciona herramientas muy valiosas para mejorar la calidad de una imagen, en las cuales los métodos tradicionales tienen limitaciones.
10. En la industria se observa aplicaciones de procesamiento de imágenes en la medicina, las cuales deben cumplir regulaciones a nivel médico

como la FDA que solicita el MSE a cada imagen compresada para su revisión posterior. La súper resolución actualmente solo se usa a nivel académico en la medicina.

11. El método de Súper Resolución (IBP) desarrollado en una siguiente versión aceptara parámetros. Este método tiene limitaciones para eliminar el ruido como principal desventaja, ello es debido a que al tomar lo mejor de cada imagen y en el caso que todas tienen problemas de ruido, lo que resulta es imagen final en SR con ruido, es por ello recomendable que los métodos que toman varias de imágenes de entrada, consideren la eliminación de ruido.
12. El método IKS minimiza el ruido de forma más eficiente y por ello los mejores resultados demostrados por un valor PSNR más elevado en el caso del IKS que el IBP.
13. Esta tecnología se encuentra en el estado del arte, por lo que seguirán mejorándose los métodos, pudiéndose obtener mejores resultados en el futuro, en la medida que avancen los modelos matemáticos de interpolación, modelos matemáticos de predicción, reconocimiento de patrones y el modelamiento de los fenómenos físicos propios del hardware del CCD que afectan la calidad de la imagen como PSF del CCD, así como optimización así como la velocidad de procesamiento.
14. La aplicación a imágenes médicas, imágenes satelitales e imágenes de fotografías, obteniéndose en todos los casos una mejora en la calidad de las imágenes, indicado por el PSNR
15. Las pruebas sintéticas son importantes para determinar la eficiencia de un método en el campo del procesamiento de imágenes, debido a que con su aplicación es posible medir el nivel de performance, variar y controlar cada fenómeno en análisis y comparar objetivamente sus resultados y luego pueden ser aplicados luego de pasar estas pruebas a aplicaciones reales.

16. La presente tesis se ha enfocado en trabajar en imágenes en escala de grises, en el caso de su aplicación en imágenes a colores es recomendable pasarlos al formato YCrCb y procesarlo canal por canal, siendo necesario más recursos de procesamiento.
17. Es posible lograr mejores resultados, si se tiene un modelo más preciso del PSF, el cual siendo un fenómeno físico, es representado como un Filtro Gauss para el presente trabajo,
18. En el caso de una aplicación comercial, se requiere alto poder de procesamiento así como procesamiento en paralelo, para imágenes de mayor tamaño y videos, y esto es posible implementándolo en C++, como es el caso del IBP. En el caso de usar MATLAB tiene limitaciones para procesamiento en 1 hilo y en paralelo.
19. Sobre los algoritmos de Súper Resolución, se tiende a desarrollar entornos o "*framework*" que usualmente absorben y toman lo mejor de varios métodos, mejorando substancialmente la aplicación a problemáticas específicas, como el Método de Kernel Steering, el cual se basa en la generalización del Método del Kernel tradicional, lo cual da una ventaja muy importante de escalabilidad y especialización de los métodos.

RECOMENDACIONES

Las siguientes recomendaciones a seguir están orientadas para mejorar los métodos existentes u otros métodos, a fin de desarrollar un proceso de mejora continua:

1. Se podría utilizar comercialmente estos métodos de súper resolución, en la medida que cumplan las normas según el campo de aplicación, y para lo cual se podría requerir una adaptación previa.
2. La solución del problema de la súper resolución está enfocado en los fenómenos que afectan a la imagen y no a una particularidad de su aplicación, lo que significa que los métodos pueden ser aplicados además de imágenes médicas a imágenes fotográficas, satelitales y en general cualquiera de los campos de aplicación de las imágenes.
3. Para ser utilizada y probada a nivel de usuario final es recomendable desarrollarlo en una siguiente etapa en una plataforma Web, y con un servidor con capacidad de procesamiento paralelo y/o a nivel de hardware, que también son utilizados por aplicaciones comerciales en procesamiento de imágenes por su rapidez y accesibilidad, siendo necesario un ancho de banda óptimo según el tamaño de imágenes.
4. Se recomienda estandarizar y desarrollar un procedimiento de pruebas sintéticas que permita asegurar la eficiencia de los métodos desarrollados.
5. Se recomienda desarrollar, probar y validar los modelos y los métodos de Súper Resolución, inicialmente en MATLAB, y luego de ello, se puede empezar el desarrollo en un lenguaje de bajo nivel como el C++
6. Los métodos se pueden mejorar o combinarse, tomando lo mejor de cada uno para solucionar un fenómeno y combinarlos a través de

framework, y así dar robustez a los nuevos métodos de super resolución basados en el aprendizaje y mejora de los existentes.

7. En el caso de trabajar con imágenes a color, es recomendable realizar pruebas en cada canal, empezar por escala de grises, si funciona con ella, será más sencillo trasladar los resultados a los otros canales. Para trabajar con imágenes a color como la imagen en RGB se recomienda pasar al formato YCrCb. Los 2 métodos propuestos, se pueden extender a imágenes a colores e imágenes de mayores dimensiones, así como medir su desempeño usando el PSNR y el RMSE.
8. Se recomienda utilizar el PSNR como indicador primario. En el presente trabajo se ha calculado el RMSE también como indicador.
9. Para los principiantes, se recomienda iniciar y realizar pruebas de los métodos de Súper Resolución, con imágenes en escala de grises de $M \times N \times 1$, y luego extender a otros formatos y/o colores.
10. El Perú está adquiriendo tecnología satelital, por lo cual este tipo de tecnología podría ser muy útil para el análisis y mejora de las imágenes obtenidas por el Satélite. UNI ha lanzado su satélite y este tipo de tecnologías, también podría ser utilizado de forma complementaria para las fotografías tomadas y por lo tanto, en el futuro el país, ser un referente en procesamiento de imágenes satelitales.
11. Se recomienda desarrollar en forma masiva, la tecnología de procesamiento de imágenes de forma sostenida en el país, desde la universidad, empresa y el estado y usar esta tecnología en los sectores como salud, imágenes satelitales, imágenes del instituto geográfico nacional, seguridad militar, civil, entre otros, ya que la dependencia tecnológica es costosa para el país, y es totalmente factible desarrollar este campo. Existen aplicaciones de gestión de imágenes médicas que utilizan intensamente el procesamiento de imágenes como las aplicaciones PACS, de las cuales nuestro país utiliza en todo el

Sistema Hospitalario y Clínico, en donde una licencia podría costar hasta 300,000 dolares por Hospital, es posible sustituir esta tecnología de forma gradual.

12. Desarrollar, implementar otras aplicaciones de procesamiento de imágenes, como reconocimiento de patrones, compresión, búsqueda entre otros que son requeridos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Shai Bagon, Daniel Glasner and Michal Irani. Superresolution from a single image. 2009
- Jacob Beutel, Harold L. Kundel, Richard L. Van Meter, Handbook of Medical Imaging. Volume 1. SPIE Press. Bellingham, Washington, USA. 2000. Society of Photo-Optical Instrumentation Engineers.
- [2] NASA, Imágenes de la NASA (www.nasa.gov)
- [3] N. K. Bose. Superresolution from image sequence. In Applied Imagery Pattern Recognition Workshop, pages 81.IEEE, 2003
- [4] Aldo Camargo, Enzo Camargo, Christian Tinoco, Iván Calle, Roger León, Carlos Carrasco, Súper Resolución de imágenes y videos con aplicación a la medicina, Proyecto Súper Rivam: Contrato N° 054-FINCYT-PITEI-2010, PERU, 2010.
- [5] Aldo Camargo Fernández Baca; Súper-resolution mosaicking from digital surveillance video captured by unmanned aircraft systems (UAS), Agosto 2010. Ph.D. Dissertation University of North Dakota,USA.
- [6] Aldo Camargo Fernández Baca, Enzo Camargo Fernández Baca; SuperRivam Software para el Cálculo de la Súper-resolución de Imágenes Médicas y Video. SPIE Medical Imaging 2013. USA, Orlando, Florida.

- [7] Michael Elad and Arie Feuer. Súper-resolution reconstruction of continuous image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:459-463, 1999.
- [8] Feng Liu, Jinjun Wang, Shenghuo Zhu, Michael Gleicher, and Yihong Gong. Noisy video súper-resolution. *ACM Multimedia*. ACM, 2008.
- [9] Marrero C. Gustavo and Barreto D. Dacil et al. Técnicas de Súper Resolución para la mejora de secuencias de video comprimido. ULPGC, Instituto de Microelectrónica Aplicada, pages 34 - 44, 2006.
- [10] T.S. Huang and R.Y. Tsai, Multi-frame image restoration and registration, in *Advances in Computer Vision and Image Processing* (T.S. Huang Ed), Vol.1, pp. 317-339, JAI Press, Greenwich, CT, 1984
- [11] Michael Irani and Samuel Peleg, *Improving Resolution By Image Registration*, Jerusalem, Israel: CVGIF: Graphical Models and Image Processing, 25 de mayo de 1990, Vol. 53.
- [12] Monica Penedo and William A Pearlman, *Medical Imaging Systems Technology: Methods in Diagnosis Optimization* (Medical Imaging Systems Technology) Dec 30, 2005 by CORNELIUS T LEONDES, Universidad of California, Los Angeles , USA 161-194 .
- [13] S. Prasad. Digital superresolution and the generalized sampling theorem. *Journal of the Optical Society of America*, 24(2):311{325, 2007.

- [14] F. Rousseau, Medical Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2010: 13th International Conference, Beijing, China, September 20-24, 2010, Proceedings, Part 2. Pag 357.
- [15] Hiroyuki Takeda, student member, IEEE, Sina Farsiu, IEEE y Peyman Milanfar, Senior Member, IEEE. . Kernel Regression for Image Processing and Reconstruction. IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 16, NO. 2, FEBRUARY 2007
- [16] Hiroyuki Takeda UNIVERSITY OF CALIFORNIA, SANTA CRUZ. KERNEL REGRESSION FOR IMAGE PROCESSING AND RECONSTRUCTION. The Thesis of Hiroyuki Takeda. MASTER OF SCIENCE in ELECTRICAL ENGINEERING, March 2006.
- [17] Sabhal Mor Ostaig, Isle of Skye, Tree-Encoded Conditional Random Fields for Image Synthesis. Information Processing in Medical Imaging: 24th International Conference, IPMI 2015, UK, June 28 - July 3, 2015, Proceedings. Pag 733.
Amod Jog, Aaron Carass, Dzung Pham and Jerry Prince. Image Analysis and Communications laboratory, The Johns Hopkins University, Baltimore, USA (amodjogjhu.edu) Henry Jackson Foundation for the Advancement of Military Medicine, Bethesda, USA
- [18] Euclid Seeram, Patrick C Brennan; Radiation Protection In Diagnostic X-Ray Imaging, Pagina 74 a la 77. Jan 27, 2016
- [19] United Nations, 2010, The world's women 2010: trends and statistics. New York :. Paginas 25 a Paginas 26

- [20] Patrick Vandewalle, Luciano Sbaiz, J. Vandewalle, and Martin Vetterli. Súper-resolution from unregistered and totally aliased signals using subspace methods. *IEEE Transactions on Signal Processing*, 55(7-2):3687-3703, 2007.
- [21] Patrick Vanderwalle, Sabine Susstrunk, Martin Vetterli A Frequency Domain Approach to Registration of Aliased Images with Application to Súper-resolution, Ecole Polytechnique Federal de Laussane, University of California, 2005.

GLOSARIO DE TÉRMINOS

Súper Resolución (SR): Es obtener una imagen de mejor calidad a partir de una imagen o varias imágenes de baja calidad, por la minimización de los fenómenos que afectan la calidad de la imagen como el ruido, desenfoque, entre otros.

PSNR: Es un método objetivo de medir la calidad de una imagen digital respecto a la imagen original, realizada mediante cálculo matemático. Cuanto más alto es el valor PSNR, mejor es su calidad. Se utiliza para pruebas sintéticas.

Subpixeles: Es la información de los nuevos pixeles calculados en la imagen en SR, creada a partir de los pixeles de la imagen en LR. También puede emplearse este término cuando se adiciona información a una imagen inicial.

Imágenes médicas digitales: Son obtenidas mediante equipos de adquisición de imágenes médicas, como rayos X, ecógrafos, tomógrafos, MRI entre otros.

Imágenes digitales: Son imágenes obtenidas mediante un CCD, el cual puede ser una cámara fotográfica, un celular, un equipo de rayos X.

CCD: Es un equipo de adquisición de imágenes como una cámara fotográfica, un equipo médico, entre otros;

Imagen Original Real (HR): Es la escena o imagen original del mundo real (original high-resolution imagen, por sus siglas en inglés, ver [6]),

Imagen de baja resolución (Low resolution image: LR): Es la imagen obtenida desde un CCD;

Imagen Súper Resolución (SR): Es las mejor aproximaciones posibles a la imagen Real, lograda por técnicas de Súper Resolución a partir de una o varias imágenes de baja resolución (LR).

DICOM (Digital Imaging and Communications in Medicine): Es un estándar reconocido mundialmente de formato de archivo, protocolo de comunicaciones para el intercambio de imágenes médicas considerando su manejo, visualización, almacenamiento, impresión y transmisión. Fue creado por NEMA y la AAR. El protocolo de comunicaciones que usa es el TCP/IP para comunicarse entre sistemas. Los archivos DICOM pueden intercambiarse entre diferentes instituciones para enviar y recibir datos de un paciente.

Lo más importante de DICOM es que es un formato de archivo y un protocolo de red:

DICOM formato de archivo: Todo las imágenes médicas son almacenadas en formato DICOM. Los equipos de imágenes médicas crean archivos DICOM, los doctores usan visualizadores de DICOM, las aplicaciones de software pueden mostrar imágenes DICOM, para diagnosticar sus hallazgos en las imágenes. Archivos DICOM contienen más que solo imágenes.

Cada archivo DICOM almacena:

- información del paciente (nombre, identificación, sexo, fecha de nacimiento, etc.)

- datos importantes de la adquisición (tipo de equipo utilizado, su configuración);
- contexto del estudio de la imagen que es usado para ligar la imagen con el tratamiento médico correspondiente.

DICOM Protocolo de red: Todas las imágenes médicas que están conectadas con la red del hospital utiliza el protocolo DICOM para intercambiar información, principalmente imágenes DICOM pero también el paciente e información del procedimiento. El protocolo de red DICOM es usado para buscar por estudios de imágenes en el archivo y restaurar los estudios de imágenes para mostrarlos en la estación de trabajo. Existen también comandos de red avanzados que son usados para controlar y seguir el tratamiento, programación en el tiempo del procedimiento, estatus de los reportes y compartir carga de trabajo entre los doctores y dispositivos de imágenes.

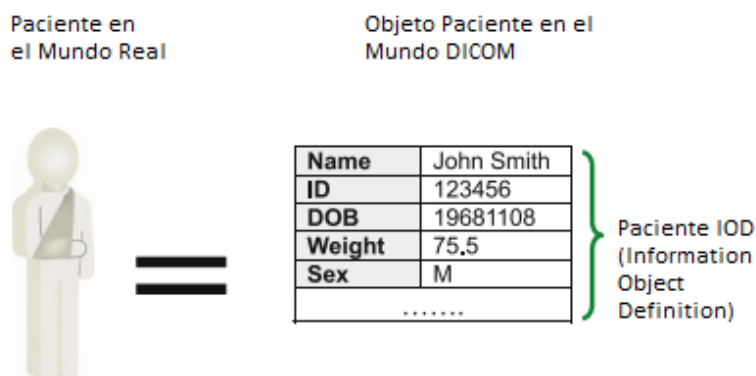


Figura A.1 Desde los datos reales a DICOM

MPRAGE: Magnetization Prepared Gradient Echo, es una variedad en secuencias de pulso para contrastación de imágenes. Fuente [18]

FLAIR: Fluid Attenuated Inversion Recovery. Fuente [18]

ANEXOS

A. Programas desarrollados en Matlab

A.1 Fuentes de modulo principal del IKS

```
% *****  
% Elaborado por : Enzo Camargo Fernandez Baca  
% *****  
% IKS : interpolacion  
% Modificado de programa desarrollado por Hiro [17]  
% *****  
% lero se asegura que la imagen sea MxNx8 bits o se pasa a  
GRAYSCALE  
  
myFolder='D:\sr\progs\01                                pruebas  
sinteticas\Medicas\seleccionados\110';  
HR = '110HR_image_BN.JPG';  
LR = 'bscap0000.JPG';  
SR1 = 'SR1.jpg';  
% SR2 = 'SR2.jpg';  
r=3;  
  
% load image  
img = double(imread(fullfile(myFolder, HR)));  
[N,M] = size(img);  
  
y = double(imread(fullfile(myFolder,LR)));  
  
% downsample the original image  
% y = downsample2(img, r);  
% y2 = y;
```

```

%
*****
*****
% 1. Interpolacion, N= 0, h=1, no iteraciones
%
*****
*****
    % N = 0

    % pilot estimation by classic kernel regression
    h1 = 0.5;
    ksize = 5;
    [zc, zx1c, zx2c] = ckr2_regular(y, h1, 1, ksize);

    % obtain orientation information
    wsize = 3;      % the size of local analysis window
    lambda = 1;     % the regularization for the elongation
parameter
    alpha = 0.05;  % the structure sensitive parameter
    C = steering(zx1c, zx2c, ones(size(y)), wsize, lambda,
alpha);

    % upscaling by steering kernel regression
    h = 1;
    ksize = 7;
    [zs, zx1s, zx2s] = skr2_regular(y, h, C, r, ksize);

    % root mean square error
    %     error = img - zs(1:512, 1:512);

    error = img - zs(1:N, 1:M);
    rmse = sqrt(mymse(error(:)));
    rmse
    peaksnr1 = psnr(zs(1:N, 1:M),img);
    peaksnr1

```

```
mini=min(min(zs)');  
zs= zs-mini;  
maxi=max(max(zs)');  
zs=zs/maxi;  
zs=uint8(zs*255);  
imwrite(zs,fullfile(myFolder,SR1));
```

A.2 Minimizar ruido con IKS

```
%
*****
***
% Elaborado por : Enzo Camargo Fernandez Baca
% *****
% Denoising : Steering Kernel (Matlab)(1), Splime Smoother
(Matlab)(2), IBP (C++)
% *****
% (1),(2) Basados en algoritmos de Takeda [17]
% *****
% lero se asegura que la imagen sea MxNx8 bits o se pasa a GRAYSCALE

myFolder          =          'D:\sr\progs\01          pruebas
sinteticas\Medicas\seleccionados\105';
f1 = 'bscap0000.JPG';
f2_con_ruido      = '100HR_image_BN_con_ruido.JPG';
f2_limpio         = '100HR_image_BN_limpio.JPG';

% myFolder = 'D:\sr\progs\TakedaToolbox\TestImages';
% f1 = 'lena.png';

% A denoising example by iterative steering kernel regression
%
% written by hiro, June 17 2007

% load image
% img = double(imread(f1));
img = double(imread(fullfile(myFolder,f1)));

[N,M] = size(img);

% add white Gaussian noise
sigma = 25;          % standard deviation
```

```

randn('state', 0); % initialization
y = round0_255(img + randn(size(img)) * sigma);

% pilot estimation by second order classic kernel regression
h = 0.5; % the global smoothing parameter
r = 1; % the upscaling factor
ksize = 5; % the kernel size
[zc, zx1c, zx2c] = ckr2_regular(y, h, r, ksize);

% iterative steering kernel regression (second order)
IT = 15; % the total number of iterations
wsize = 11; % the size of the local orientation analysis window
lambda = 1; % the regularization for the elongation parameter
alpha = 0.5; % the structure sensitive parameter
h = 2.4; % the global smoothing parameter
ksize = 21; % the kernel size
z = zeros(N, M, IT+1);
zx1 = zeros(N, M, IT+1);
zx2 = zeros(N, M, IT+1);
rmse = zeros(IT+1, 1);
z(:, :, 1) = y;
zx1(:, :, 1) = zx1c;
zx2(:, :, 1) = zx2c;
error = img - y;
rmse(1) = sqrt(mymse(error(:)));

for i = 2 : IT+1
    % compute steering matrix
    C = steering(zx1(:, :, i-1), zx2(:, :, i-1), ones(size(img)),
wsize, lambda, alpha);
    % steering kernel regression
    [zs, zx1s, zx2s] = skr2_regular(z(:, :, i-1), h, C, r, ksize);
    z(:, :, i) = zs;
    zx1(:, :, i) = zx1s;
    zx2(:, :, i) = zx2s;
end

```

```

    % root mean square error
    error = img - zs;
    rmse(i) = sqrt(mymse(error(:)));
    rmse(i)
    %figure(99); imagesc(zs); colormap(gray); axis image; pause(1);
end

% display images
figure; imagesc(y); colormap(gray); axis image;
title(['The noisy image, STD=25, RMSE=', num2str(rmse(1))]);
figure; imagesc(z(:, :, 13)); colormap(gray); axis image;
title(['ISKR, 12 iterations, RMSE=', num2str(rmse(13))]);

% Pasar a archivos y grabar las imagenes
zz_noise =y;
zz      =z(:, :, 13);

mini=min(min(zz_noise)');
zz_noise= zz_noise-mini;
maxi=max(max(zz_noise)');
zz_noise=zz_noise/maxi;
zz_noise=uint8(zz_noise*255);
imwrite(zz_noise,fullfile(myFolder,f2_con_ruido));

mini=min(min(zz)');
zz= zz-mini;
maxi=max(max(zz)');
zz=zz/maxi;
zz=uint8(zz*255);
imwrite(zz,fullfile(myFolder,f2_limpio));

```

A.3 Función de soporte: Cálculo de PSNR # 1

```
% *****
% Objetivos:
% Obtener Imagenes con super resolucion y sus medicion de calidad
mediante el PSNR, y RMSE respectivos,
% a traves de metodos de interpolacion (bicubica, lineal, lanczos3
) y super resolucion usando
% Kernel e IBP
% INPUT1: LR images(1..n) imagenes LR
% OUTPUT: SR imagen ( para cada metodo utilizado)
%          PSNR(SR imagen, ref), RMSE para cada imagen generada
% *****
% PARAMETROS DE ENTRADA:
myFolder    =    'D:\sr\progs\02    Pruebas    sinteticas    por
etapas\ImagenesTest';
HR_image = '100HR_image_BN.JPG';
LR_image = 'bscap0000.JPG';

SR_PNG1 = 'Img_En_SuperResolution 1 con ruido.PNG';
SR_PNG2 = 'Img_En_SuperResolution 2 limpio.PNG';
HR1 = '100HR_image_BN_limpio.JPG';
HR2 = '100HR_image_BN_con_ruido.JPG';

img = double(imread(fullfile(myFolder, HR_image)));
[XR, YR, ZR] = size (img);

% Cargando imagenes : HR y SR
HR1F = double(imread(fullfile(myFolder, HR1)));
HR2F = double(imread(fullfile(myFolder, HR2)));
SR1F = double(imread(fullfile(myFolder, SR_PNG1)));
SR2F = double(imread(fullfile(myFolder, SR_PNG2)));
% SR_KERNEL = double(imread(fullfile(myFolder, 'SR_Kernel.JPG')));

SR11 = imresize(SR1F, [XR YR]);
```



```

SR21 = imresize(SR2F, [XR YR]);

%
% % 3. Obtener SR por interpolacion bicubica
% SR_IBICUBIC = imresize(LR, [x y], 'bicubic'); % 'bicubic' is
default
% imwrite(SR_IBICUBIC, fullfile(myFolder, '01SR BICUBIC 001.jpg'));

% =====
% Calculando el PSNR Y RMSE
% =====
peaksnr1 = psnr(SR11, img);
peaksnr2 = psnr(SR21, img);
peaksnr11 = psnr(HR1F, img);
peaksnr12 = psnr(HR2F, img);

error = img - SR11;
rmse(1) = sqrt(mymse(error(:)));
error = img - SR21;
rmse(2) = sqrt(mymse(error(:)));
error = img - HR1F;
rmse(3) = sqrt(mymse(error(:)));
error = img - HR2F;
rmse(4) = sqrt(mymse(error(:)));

fprintf('\n The Peak-SNR -1 (SR11 con ruido) value is %0.4f',
peaksnr1);
fprintf('\n The Peak-SNR -2 (SR11 limpio) value is %0.4f',
peaksnr2);
fprintf('\n The Peak-SNR -11 (HR1F) value is %0.4f', peaksnr11);
fprintf('\n The Peak-SNR -12 (HR2F) value is %0.4f', peaksnr12);

for i = 1 : 4
    fprintf('\n RMSE value is %0.4f', rmse(i));
end

```

A.4 Funciones de Soporte: Cálculo de PSNR # 2

```
% *****  
% Objetivos:  
% Obtener Imagenes con super resolucion y sus medicion de calidad  
mediante el PSNR, y RMSE respectivos,  
% Ejercicio Kernel interpolacion  
% INPUT1: SR1, SR2  
% OUTPUT: PSNR(SR imagen, ref), RMSE para cada imagen generada  
% *****  
% PARAMETROS DE ENTRADA:  
myFolder = 'D:\sr\progs\01 pruebas  
sinteticas\Medicas\seleccionados\110';  
HR = '110HR_image_BN.jpg';  
SR11 = 'SR_KERNEL.JPG';  
SR12 = 'Img_En_SuperResolution.jpg';  
SR21 = 'SR1.JPG';  
SR22 = 'Img_En_SuperResolution.PNG';  
  
img = double(imread(fullfile(myFolder, HR)));  
[XR, YR, ZR] = size (img);  
  
% Cargando imagenes : HR y SR  
SR1F = double(imread(fullfile(myFolder, SR11)));  
SR2F = double(imread(fullfile(myFolder, SR12)));  
SR3F = double(imread(fullfile(myFolder, SR21)));  
SR4F = double(imread(fullfile(myFolder, SR22)));  
  
% SR_KERNEL = double(imread(fullfile(myFolder, 'SR_Kernel.JPG')));
```

```

SR1 = imresize(SR1F, [XR YR]);
SR2 = imresize(SR2F, [XR YR]);
SR3 = imresize(SR3F, [XR YR]);
SR4 = imresize(SR4F, [XR YR]);

% =====
% Calculando el PSNR Y RMSE
% =====

peaksnr(1) = psnr(SR1,img);
peaksnr(2) = psnr(SR2,img);
peaksnr(3) = psnr(SR3,img);
peaksnr(4) = psnr(SR4,img);

error = img - SR1;
rmse_b(1) = sqrt(mymse(error(:)));
error = img - SR2;
rmse_b(2) = sqrt(mymse(error(:)));
error = img - SR3;
rmse_b(3) = sqrt(mymse(error(:)));
error = img - SR4;
rmse_b(4) = sqrt(mymse(error(:)));

for i = 1 : 4
    fprintf('\n The Peak-SNR -2 (SR2) value is %0.4f',
peaksnr(i));
    fprintf('\n RMSE value is %0.4f', rmse_b(i));
end

```

A.5 Pasar a Escala de grises

```
% ecfb

% pasar a grayscale una imagen

myFolder          =          'D:\sr\progs\01          pruebas
sinteticas\Medicas\seleccionados\t';
f1 = 'mdb105_orig.BMP';
f2 = 'mdb110_orig.BMP';
f3 = 'mdb122_orig.BMP';

fg1 = '105HR_image_BN.JPG';
fg2 = '110HR_image_BN.JPG';
fg3 = '122HR_image_BN.JPG';

h1 = imread(fullfile(myFolder,f1));
h2 = imread(fullfile(myFolder,f2));
h3 = imread(fullfile(myFolder,f3));

imshow(h1);
imshow(h2);
imshow(h3);

L1 = rgb2gray(h1);
L2 = rgb2gray(h2);
L3 = rgb2gray(h3);

imshow(L1,fullfile(myFolder,fg1));
imshow(L2,fullfile(myFolder,fg2));
imshow(L3,fullfile(myFolder,fg3));

imshow(L1);
```

```
imtool(L2);  
imtool(L3);
```

B. Programas C++ del método de super resolución IBP [5]

```
#include "GenSuperResolucion.h"

int main(){

const int N = 4;

printf(" 1 main \n ");
std::vector<char *>input(N);

printf(" 2 main \n ");

input[0] = "bscap0000.jpg";
input[1] = "bscap0001.jpg";
input[2] = "bscap0002.jpg";
input[3] = "bscap0003.jpg";

printf(" 3 main \n ");
GenerarSR( input, N);

printf(" 4 main \n ");
input.~vector();

printf(" 5 main \n ");

}

#include "GenSuperResolucion.h"

void GenerarSR( std::vector<char *>input, const int &N){

//SYSTEMTIME t1;
//SYSTEMTIME t2;

int i;
```

```

//*****
/
// Especificacion el numero de imagenes a ingresar, variable a enlazr con
// la interfaz grafica

//
//*****

/*Definiendo la variables necesarias para ejecutar cada uno de los 3
algoritmos*/
std::vector<IplImage *>pblur(N);
std::vector<IplImage *>tmp(N);
IplImage *w;
//IplImage *pblur[N],*w;
//IplImage *tmp[N];

/*Definiendo la variables para calcular las rotaciones entre cada par de
imagenes*/

double *d;
float exc;
int n = 25;
double factor;
/*Definiendo la variable donde se almacenara la rotacion estimada*/
CvMat *angle_est;
IplImage *delta, *phi;

//*****
/
//*****
/
// Ingresando las imagenes del registro
//
// Rutas a asignar mediante la interfaz grafica

std::cout << " Ingresando las imagenes del registro \n";

std::vector<IplImage *>reg(N);
//IplImage *reg[N];

    reg[0] = cvLoadImage( input[0] , CV_LOAD_IMAGE_GRAYSCALE );
    reg[1] = cvLoadImage( input[1], CV_LOAD_IMAGE_GRAYSCALE );
    reg[2] = cvLoadImage( input[2], CV_LOAD_IMAGE_GRAYSCALE );
    reg[3] = cvLoadImage( input[3], CV_LOAD_IMAGE_GRAYSCALE );

```

```

//*****
/
//*****
/
/*
    IplImage *ass = cvLoadImage("Koala1.tif", CV_LOAD_IMAGE_GRAYSCALE);
    IplImage *dst = cvCreateImage(cvSize(ass->width,ass-
>height),IPL_DEPTH_32F,1);
    IplImage *dst2 = cvCreateImage(cvSize(2*ass->width,2*ass-
>height),IPL_DEPTH_32F,1);
    F8to64f(ass,dst);
    int width = ass->width;
    int height = ass->height;
    int elemL = width*height;
    int elemH = dst2->width*dst2->height;
    size_t datasize = elemL*sizeof(float);
    int j;
    float *d123 = (float *)malloc(elemL*sizeof(float));//entrada, se
borra al ejecutar opencLrotarblurr
    float *outputhigh = (float *)malloc( elemH*sizeof(float));//entrada,
se borra al ejecutar opencLrotarblurr

    for( i=0; i<height; i++){
        float *p_src = (float *)(dst->imageData + i*dst->widthStep);
        for( j=0; j<width; j++){
            d123[j+i*width] = p_src[j];
        }
    }
    cvNamedWindow("rot",CV_WINDOW_AUTOSIZE);
    cvShowImage("rot", dst);
    cvWaitKey(0);
    OpenCLResize(d123, outputhigh, width, height, dst2->width, dst2-
>height, elemH, elemL);

    for( i=0; i<dst2->height; i++){
        float *pc = (float *)(dst2->imageData + i*dst2->widthStep);
        for( j=0; j<dst2->width; j++){
            pc[j] = outputhigh[dst2->width*i+j];
        }
    }
    cvNamedWindow("rotan",CV_WINDOW_AUTOSIZE);
    cvShowImage("rotan", dst2);
    cvWaitKey(0);

    cv::namedWindow("original",CV_WINDOW_AUTOSIZE);
    cv::imshow("original", ass);
    cv::waitKey(0);

    cv::gpu::GpuMat sd;
    sd.upload(ass);
    sd.convertTo(sd,CV_32FC1,1,0);
    GetSystemTime(&t1);

```



```

gc_rotarImagen(sd, sd , 0.789 );
GetSystemTime(&t2);

    delta_tiempo(t1, t2);*/

/*****
*****/
/*****
*****/
// Calculo de las estimaciones de rotacion
std::vector<IplImage *>h_A(N);
std::vector<IplImage *>H_AI(N);
std::vector<IplImage *>H_AR(N);//
//IplImage *h_A[N];
//IplImage *H_AI[N];
//IplImage *H_AR[N];

/* Creando la imagen w, que se empleara para filtrar cada imagen del
registro y */
/* eliminar el fenomeno de aliasing*/

    w = cvCreateImage(cvGetSize(reg[0]),IPL_DEPTH_64F,reg[0]->nChannels);
    tukeywin(w,1);/*
    cvNamedWindow("tukey",CV_WINDOW_AUTOSIZE);
    cvShowImage("tukey",w);
    cvWaitKey(0);*/
    for(i=0; i<N; i++){
        pblur[i]=cvCreateImage(cvGetSize(reg[0]),IPL_DEPTH_32F,reg[0]-
>nChannels);
    }

    for(i=0; i<N; i++){
        cvMul(reg[i],w,pblur[i],1);
    }

    angle_est = cvCreateMat(1,N,CV_64FC1);
    /* Cargando los Parametros para iniciar el Algoritmo de Rotacion */
    d = (double*) malloc( 1*2*sizeof(int) );
        d[0]=0.515;
        d[1]=0.585;
    exc = (float)0.15;

std::cout << " 2 \n";
    est_rotacion(pblur,d,exc,angle_est, N, h_A, H_AI, H_AR );

    fprintf(stdout,"PARAMETROS DE ROTACION: \n ");

    for(i=0; i<N; i++){
        fprintf( stdout,"Angulo[%i] : %.15f \n ",i
,cvGetReal2D(angle_est,0,i) );
    }

```

```

/*****
*****/
/*****
*****/

/*****
*****/
/*****
*****/
// Caculo de las estimaciones de desplazamiento
//

std::cout << " 3 \n";

for(i=0; i<N; i++){
    tmp[i]=cvCreateImage(cvGetSize(reg[0]),IPL_DEPTH_32F,1);
    rotarImagen( pblur[i] , tmp[i] , -cvGetReal2D(angle_est,0,i)
);
}

/*Definiendo la variable donde se almacenara los desplazamientos
estimados*/
CvMat* delta_est;

/* Obteniendo la estimacion por desplazamientos */
delta_est = estimacion_desplazamiento( tmp, n, N);

fprintf(stdout, "\nPARAMETROS DE DESPLAZAMIENTO:\n");
for( i=0; i<N; i++){
    fprintf(stdout, " x: %.15f , y:
%.15f\n",CV_MAT_ELEM(*delta_est,double,i,0) ,
CV_MAT_ELEM(*delta_est,double,i,1) );
}
//cvReleaseImage(tmp);

/*****
*****/
/*****
*****/

/*****
*****/
/*****
*****/
// Algoritmo iterative Back Projecting para generar la imagen en super
resolucion //

```

```

        std::cout << " 4 \n ";

std::vector<IplImage *>tmp1(N);
for(i=0; i<N; i++){
    tmp1[i] = cvCreateImage(cvGetSize(reg[0]),IPL_DEPTH_32F,1);
}

/*Convirtiendo la imagen a formato double */
for(i=0; i<N; i++){
    F8to64f( reg[i], tmp1[i] );
}

fprintf( stdout, "\nITERATIVE BACK PROJECTING:\n" );

// Especificando el factor de incremento de la imagen en superresolucion
// respecto de//
// la imagen en baja resolucion
//
factor = 3;

delta = cvCreateImage(cvSize(2,N),IPL_DEPTH_64F,1);
phi = cvCreateImage(cvSize(1,N),IPL_DEPTH_64F,1);

for(i=0; i<phi->height ; i++){
    cvSetReal2D( phi, i, 0, cvGetReal2D(angle_est,0,i) );
}
std::cout << " 5 \n ";
//GetSystemTime(&t1);
std::cout << " 6 \n";
IterativeBackProjecting(tmp1 , delta , phi , factor, N);

/*****
*****/
/*****
*****/

//GetSystemTime(&t2);

std::cout << " 7 \n ";

//delta_tiempo(t1, t2);

reg.~reg();
reg.~pblur();
reg.~tmp();
reg.~tmp1();
reg.~h_A();
//cvReleaseImage(reg);
//cvReleaseImage(pblur);
cvReleaseImage(&w);
//cvReleaseImage(tmp);

```

```

        //cvReleaseImage(tmp1);
        //cvReleaseImage(h_A);

    }

#include "LibDesplazamiento.h"

CvMat *estimacion_desplazamiento( std::vector<IplImage *>reg, int n , const
int N)
{
    int i, mx, my,m;
    static const float pi = (float)3.141592653589793;

    /* Estableciendo las propiedades de la imagen */
    int width = reg[0]->width;
    int height = reg[0]->height;
    int step = reg[0]->widthStep;
    int fft_size = width * height;

    /* Definiendo las variables necesarios para aplicar el filtro */
    int beginy;
    int endy;
    int beginx;
    int endx;

    /* Definiendo la matriz para calcular la diferencia de fases */
    CvMat* Ang = cvCreateMat(height,width,CV_32FC1);

    /* Definiendo las variables para la transformada de Fourier de la
referencia*/
    IplImage * image_Re1 = cvCreateImage( cvSize(width, height),
IPL_DEPTH_32F, 1);
    IplImage * image_Im1 = cvCreateImage( cvSize(width, height),
IPL_DEPTH_32F, 1);

    /* Definiendo las variables para la transformada de Fourier de la
imagen del registro*/
    IplImage * image_Re2 = cvCreateImage( cvSize(width, height),
IPL_DEPTH_32F, 1);
    IplImage * image_Im2 = cvCreateImage( cvSize(width, height),
IPL_DEPTH_32F, 1);

    /* Transformada de Fourier de la imagen de referencia */
    DFT(reg[0],image_Re1,image_Im1);

```

```

        /*Creando la ventana para filtrar las bajas frecuencias y eliminar el
aliasing */
        beginy = (int)floor((float)(height/2))-n+1;
        endy   = (int)floor((float)height/2)+n+1;
        beginx = (int)floor((float)width/2)-n+1;
        endx   = (int)floor((float)width/2)+n+1;

        my = (endy - beginy + 1);
        mx = (endx - beginx + 1);
        m = mx*my;

        CvMat *delta_est;

        delta_est = cvCreateMat(N,2,CV_64FC1);
        /* Iniciando los valores estimados */
        CV_MAT_ELEM(*delta_est,double,0,0) = 0;
        CV_MAT_ELEM(*delta_est,double,0,1) = 0;

        CvMat *B;
        B = cvCreateMat( m, 1, CV_32FC1);

        /*Desarrollando las coordenadas x e y de los pixeles */
        float *x, *y;
        x = (float*) malloc( m *sizeof(int) );
        y = (float*) malloc( m *sizeof(int) );

        ventanaxy(x, y, mx, my,beginx, beginy);

        for(i=1 ; i<N; i++){

            DFT(reg[i],image_Re2,image_Im2);

            g_angulo(image_Im1, image_Re1, image_Im2, image_Re2, Ang);

            /*Definiendo la matriz v que son los elementos de angle en las
coordenadas establecidas por x e y */
            matriz_v(beginx,beginy,Ang,mx,my,B);

            /* Resolviendo el sistema lineal de ecuaciones*/
            prueba(x, y, B ,beginx, endx, beginy, endy, width, height,
delta_est, i);
        }

        cvReleaseMat(&Ang);
        cvReleaseMat(&B);
        free(x);
        free(y);

        cvReleaseImage(&image_Re1);
        cvReleaseImage(&image_Re2);
        cvReleaseImage(&image_Im1);
        cvReleaseImage(&image_Im2);

        return delta_est;

```

```

}

void DFT(IplImage *src, IplImage *dst, IplImage *dstIm){
    int w, h;

    w = src->width;
    h = src->height;

    std::vector<cv::gpu::GpuMat>entrada(2);
    cv::gpu::GpuMat g_complexIn1(h,w,CV_32FC2);
    cv::gpu::GpuMat g_complex(h,w,CV_32FC2);

    entrada[0].upload(src);
    entrada[0].convertTo(entrada[0],CV_32FC1,1,0);
    entrada[1].create(h,w,CV_32FC1);
    entrada[1].setTo(0);

    //calculo de la transformada discreta de fourier
    cv::gpu::merge(entrada,g_complexIn1);
    cv::gpu::dft(g_complexIn1 , g_complex , cv::Size(w,h) ,
CV_DXT_FORWARD);
    cv::gpu::split(g_complex,entrada);

    //Desplazando la frecuencia fundamental al centro
    g_cvShiftDFT(entrada[0],entrada[0]);
    g_cvShiftDFT(entrada[1],entrada[1]);

    //Copiando al formato IplImage
    ArrtoIplImageD((cv::Mat)entrada[0], dst);
    ArrtoIplImageD((cv::Mat)entrada[1], dstIm);

    entrada.~vector();
}

void prueba(float *x, float *y, CvMat *v ,int &beginx, int &endx, int
&beginy, int &endy, int &width, int &height, CvMat *delta_est, int &index){
    int i,l;
    CvMat* A ;
    CvMat* Xr ;

    l = (endy - beginy +1)*(endx - beginx + 1);
    A = cvCreateMat(1,3,CV_32FC1);
    Xr = cvCreateMat(3,1,CV_32FC1);

    /* Ingresando los Valores de la Matriz A*/
    for(i=0;i< l ;i++)
    {
        CV_MAT_ELEM(*A,float,i,0) = y[i];
        CV_MAT_ELEM(*A,float,i,1) = x[i];
        CV_MAT_ELEM(*A,float,i,2) = 1;
    }
}

```

```

cvSolve(A, v, Xr, CV_SVD);
solucion(Xr,width,height,delta_est, index);

cvReleaseMat(&A);
cvReleaseMat(&Xr);
}

void solucion(CvMat *Xr,int &width,int &height,CvMat *delta_est, int
&index){
    float x1,x2,x3;
    static const float pi = (float)3.1415926535;

    x1    = CV_MAT_ELEM(*Xr, float, 0, 0);
    x2    = CV_MAT_ELEM(*Xr, float, 1, 0);
    x3    = CV_MAT_ELEM(*Xr, float, 2, 0);

    CV_MAT_ELEM(*delta_est, double, index, 0) = -x1*width/( 2*(float)pi
);
    CV_MAT_ELEM(*delta_est, double, index, 1) = -x2*height/( 2*(float)pi
);
}

void coreccionang(CvMat *Ang, int i, int j){
    static const double pi = 3.14159265358979323846;
    double k;

    k = cvGetReal2D(Ang, i, j);
    if( k < -pi){
        CV_MAT_ELEM(*Ang, double, i, j) = k + (double)2*pi;
    }

    if( k > pi){
        CV_MAT_ELEM(*Ang, double, i, j) = k - (double)2*pi;
    }
}

void ventanaxy(float *x, float *y, int &mx, int &my, int &beginx, int
&beginy){
    int i,j;

    for( i=0 ; i< my ; i++){
        for( j=0; j< mx ; j++ ) {
            x[ (j + i * mx) ] = (float)beginx + (float)j;
        }
    }

    for( i=0 ; i< my ; i++){
        for( j=0; j< mx ; j++ ) {
            y[ (j + i*mx) ] = (float)i + (float)beginy;
        }
    }
}

```

```

    }
}

void error_ref(){
    IplImage *err = 0;
    err = cvLoadImage( "ERROR_REF.jpg", CV_LOAD_IMAGE_COLOR );
    cvNamedWindow("ERROR",CV_WINDOW_AUTOSIZE);
    cvShowImage("ERROR",err);
    cvWaitKey(5000);
}

void error_reg(){
    IplImage *err = 0;
    err = cvLoadImage( "ERROR_REG.jpg", CV_LOAD_IMAGE_COLOR );
    cvNamedWindow("ERROR",CV_WINDOW_AUTOSIZE);
    cvShowImage("ERROR",err);
    cvWaitKey(5000);
}

void mostrar_matriz(int filas, int columnas, float *matriz){
    int i,j;
    for( i=0 ; i< filas ; i++){
        for( j=0; j< (columnas-1) ; j++ ) {
            fprintf(stdout, "%f ", matriz[j + i*columnas ]);
        }
        fprintf(stdout, "%f\n", matriz[j + i*columnas ]);
    }
    fprintf(stdout, "\n\n");
}

void cvShiftDFT(CvArr * src_arr, CvArr * dst_arr )
{
    CvMat * tmp;
    CvMat q1stub, q2stub;
    CvMat q3stub, q4stub;
    CvMat d1stub, d2stub;
    CvMat d3stub, d4stub;
    CvMat * q1, * q2, * q3, * q4;
    CvMat * d1, * d2, * d3, * d4;

    CvSize size = cvGetSize(src_arr);
    CvSize dst_size = cvGetSize(dst_arr);
    int cx, cy;

    if(dst_size.width != size.width || dst_size.height != size.height){
        cvError( CV_StsUnmatchedSizes, "cvShiftDFT", "Source and Destination
arrays must have equal sizes",__FILE__, __LINE__ );
    }

    if(src_arr==dst_arr){

```



```

        tmp = cvCreateMat(size.height/2, size.width/2,
cvGetElemType(src_arr));
    }

    cx = size.width/2;
    cy = size.height/2; // image center

    q1 = cvGetSubRect( src_arr, &q1stub, cvRect(0,0,cx, cy) );
    q2 = cvGetSubRect( src_arr, &q2stub, cvRect(cx,0,cx,cy) );
    q3 = cvGetSubRect( src_arr, &q3stub, cvRect(cx,cy,cx,cy) );
    q4 = cvGetSubRect( src_arr, &q4stub, cvRect(0,cy,cx,cy) );
    d1 = cvGetSubRect( src_arr, &d1stub, cvRect(0,0,cx,cy) );
    d2 = cvGetSubRect( src_arr, &d2stub, cvRect(cx,0,cx,cy) );
    d3 = cvGetSubRect( src_arr, &d3stub, cvRect(cx,cy,cx,cy) );
    d4 = cvGetSubRect( src_arr, &d4stub, cvRect(0,cy,cx,cy) );

    if(src_arr!=dst_arr){
        if( !CV_ARE_TYPES_EQ( q1, d1 )){
            cvError( CV_StsUnmatchedFormats, "cvShiftDFT", "Source and
Destination arrays must have the same format", __FILE__, __LINE__ );
        }
        cvCopy(q3, d1, 0);
        cvCopy(q4, d2, 0);
        cvCopy(q1, d3, 0);
        cvCopy(q2, d4, 0);
    }
    else{
        cvCopy(q3, tmp, 0);
        cvCopy(q1, q3, 0);
        cvCopy(tmp, q1, 0);
        cvCopy(q4, tmp, 0);
        cvCopy(q2, q4, 0);
        cvCopy(tmp, q2, 0);
    }
}

void g_cvShiftDFT(cv::gpu::GpuMat &src_arr,cv::gpu::GpuMat &dst_arr ){

    int cx, cy;

    cx = src_arr.cols;
    cy = src_arr.rows;

    cv::gpu::GpuMat q1 = src_arr.operator()( cv::Rect(0,0,cx/2,cy/2) );
    cv::gpu::GpuMat q2 = src_arr.operator()( cv::Rect(cx/2,0,cx/2,cy/2)
);
    cv::gpu::GpuMat q3 = src_arr.operator()(
cv::Rect(cx/2,cy/2,cx/2,cy/2) );
    cv::gpu::GpuMat q4 = src_arr.operator()( cv::Rect(0,cy/2,cx/2,cy/2)
);

    cv::gpu::GpuMat tmp(cy,cx,src_arr.type());

```

```

    q3.copyTo(tmp);
    q1.copyTo(q3);
    tmp.copyTo(q1);
    q4.copyTo(tmp);
    q2.copyTo(q4);
    tmp.copyTo(q2);

    tmp.~GpuMat();
    q1.~GpuMat();
    q2.~GpuMat();
    q3.~GpuMat();
    q4.~GpuMat();
}

void espectro_magnitud(IplImage *image_Re2, IplImage *image_Im2 ){
    double m2, M2;
    cvPow( image_Re2, image_Re2, 2.0);
    cvPow( image_Im2, image_Im2, 2.0);
    cvAdd( image_Re2, image_Im2, image_Re2, NULL);
    cvPow( image_Re2, image_Re2, 0.5 );
    cvAddS( image_Re2, cvScalarAll(1.0), image_Re2, NULL ); // 1 +
Mag
    cvLog( image_Re2, image_Re2 ); // log(1 + Mag)
    cvNamedWindow("Espectro de Magnitud de la Imagen del
Registro",CV_WINDOW_AUTOSIZE);
    cvShiftDFT( image_Re2, image_Re2 );
    cvMinMaxLoc(image_Re2, &m2, &M2, NULL, NULL, NULL);
    cvScale(image_Re2, image_Re2, 1.0/(M2-m2), 1.0*(-m2)/(M2-m2));
    cvShowImage("Espectro de Magnitud de la Imagen del Registro",
image_Re2);
}

void angulo(IplImage *image_Im1, IplImage *image_Re1, IplImage *image_Im2,
IplImage *image_Re2, CvMat *Ang){

    int i,j;
    int height, width;

    height = image_Re1->height;
    width = image_Re1->width;

    /*inciando la matriz angle cuyos elementos son resultados de
la diferencia de fase en cada elemento de la
imagen*/
    for( i = 0; i < height ; i++ ) {
        for( j = 0 ; j < width ; j++ ) {

```

```

        cvSetReal2D(Ang, i, j, atan2(
cvGetReal2D(image_Im1,i,j),cvGetReal2D(image_Re1,i,j) ) - atan2(
cvGetReal2D(image_Im2,i,j),cvGetReal2D(image_Re2,i,j) ) );
    }
}
/*Factor de correccion */
for( i = 0; i < height ; i++ ) {
    for( j = 0 ; j < width ; j++) {
        coreccionang(Ang,i,j);
    }
}
}

void g_angulo(IplImage *image_Im1, IplImage *image_Re1, IplImage *image_Im2,
IplImage *image_Re2, CvMat *Ang){

    int i,j,k;
    int height, width;

    float *Re1 ; // Input array
    float *Im1 ; // Input array
    float *Re2 ; // Input array
    float *Im2 ; // Input array
    float *R ; // Output array

    height = image_Re1->height;
    width = image_Re1->width;

    // numero de elementos en cada arreglo
    int elements = height*width;
    // tamaño de los datos
    size_t datasize = sizeof(float)*elements;
    Re1 = (float*)malloc(datasize);
    Im1 = (float*)malloc(datasize);
    Re2 = (float*)malloc(datasize);
    Im2 = (float*)malloc(datasize);
    R = (float*)malloc(datasize);

    for( i = 0; i < height ; i++ ) {
        float* ptr1 = (float*)(image_Re1->imageData +
i*image_Re1->widthStep);
        float* ptr2 = (float*)(image_Im1->imageData +
i*image_Im1->widthStep);
        float* ptr3 = (float*)(image_Re2->imageData +
i*image_Re2->widthStep);
        float* ptr4 = (float*)(image_Im2->imageData +
i*image_Im2->widthStep);
        //for( j = 0 ; j < width ; j++) {
            memcpy( Re1 + i*width ,ptr1,width*sizeof(float)
);
            memcpy( Im1 + i*width ,ptr2,width*sizeof(float)
);
        }
    }
}

```

```

);
memcpy( Re2 + i*width ,ptr3,width*sizeof(float)
);
memcpy( Im2 + i*width ,ptr4,width*sizeof(float)
);
//k = j+i*width;
//Re1[k]= ptr1[j];
//Im1[k]= ptr2[j];
//Re2[k]= ptr3[j];
//Im2[k]= ptr4[j];
//}
}
g_tang(Re1, Im1,Re2,Im2, R, elements);

/*inciando la matriz angle cuyos elementos son resultados de
la diferencia de fase en cada elemento de la imagen*/

}

void matriz_v(int &beginx, int &beginy, CvMat *Ang, int &mx, int &my, CvMat
*B){
int i,j;
for( i=0 ; i< my ; i++){
for( j=0; j< mx ; j++ ) {
CV_MAT_ELEM(*B, float, i + j*mx, 0 ) =
CV_MAT_ELEM(*Ang, float, i+beginy-1, j+beginx-1 ) ;
}
}
}

void ArrtoIplImageD(cv::Mat src, IplImage *dst){
IplImage dst1 = src;
cvScale(&dst1,dst,1,0);
}

#include "LibRotacion.h"

void est_rotacion( std::vector<IplImage *>reg, double *d_front, float
precision,CvMat *angle_est, const int N, std::vector<IplImage *>h_A,
std::vector<IplImage *>H_AI, std::vector<IplImage *>H_AR )
{
/* Variables de Entrada */
int i, j ;

```

```

double d ;
IplImage *m ;
IplImage *n ;
static const float pi = (float)3.141592653589793 ;

/* Estableciendo las propiedades de la imagen de referencia */
int width      = reg[0]->width ;
int height     = reg[0]->height;

/* Definiendo las variables CvArray para la conversion de coordenadas
*/
CvMat* th;
CvMat* th1;
CvMat* ra;
CvMat* ix;
CvMat* T ;

/* Definiendo las variables para la transformada de Fourier de la
primera Imagen */
IplImage * image_Re1;           //Salida, parte real
IplImage * image_Im1;          //Salida, parte imaginaria

//indica la cantidad de elementos que se obtieron debido a un
promedio
float nep;

/* Estableciendo el centro de la imagen y la matriz que sera el
dominio de las frecuencias(coordenadas)*/
d = 1*pi/180;

//Creando ra y th
th = cvCreateMat(width,height,CV_32FC1);
ra = cvCreateMat(width,height,CV_32FC1);
CPolares( ra, th, width, height);

//Pasando a OpenCL
float *cl_th; //Entrada
float *cl_ra; //Entrada
float *cl_tt; //Salida
float *cl_thR; //Entrada
float *dd;

// Numero de elementos en cada arreglo
int elements = height*width;
size_t datasize = sizeof(float)*elements;
cl_th  = (float*)malloc(datasize);
cl_ra  = (float*)malloc(datasize);
cl_tt  = (float*)malloc(datasize);
cl_thR = (float*)malloc(datasize);
dd     = (float*)malloc(sizeof(float)*2);

for(i=0 ; i < width;i++)

```

```

{
    float *pth = (float *)(th->data.ptr + i *th->step);
    float *pra = (float *)(ra->data.ptr + i *ra->step);
    //for(j=0; j< height ;j++){
        memcpy( cl_th + i*height , pth, height*sizeof(float) );
        memcpy( cl_ra + i*height , pra, height*sizeof(float) );
        //    cl_th[j+i*height] = pth[j];
        //    cl_ra[j+i*height] = pra[j];
        //}
    }

    /*El rango th calculado es de [0;2*pi], se realiza una traslacion
para que rango sea
    desde [ -pi ; pi ]*/
    g_factor1(cl_th, cl_thR, elements);

    /*Seleccionando los valores de "ra" que estan dentro del intervalo [
d_front(0), d_front(1) ]
    Cambiando algunos elementos de la tt segun DB*/
    dd[0] = (float)d_front[0];
    dd[1] = (float)d_front[1];
    g_selec(cl_ra, cl_tt , cl_thR, dd, elements);

    /* Antes de ordenar los angulos, se les reubicara en una sola fila */
    th1 = cvCreateMat(1,height*width,CV_32FC1);
    cvReleaseMat(&th);
    cvReleaseMat(&ra);

    /* Aplicando un factor de correcion con respecto a los resultados que
da Matlab, debido al valor de -pi */
    float kk = (float)-pi+(float)0.000001;
    for(i=0 ; i < width*height;i++)
    {
        float *ptt = (float *)(th1->data.ptr);
        if ( cl_tt[i] < kk ){
            ptt[i] = cl_tt[i] +2*(float)pi;
        } else {
            ptt[i] = cl_tt[i];
        }
    }
}

/* Ordenando los elementos de la tt en forma ascendente */
ix    =    cvCreateMat(1,height*width,CV_32SC1);
T     =    cvCreateMat(1,height*width,CV_32FC1);
cvSort(th1,T,ix,CV_SORT_ASCENDING);

cvReleaseMat(&th1);

/* Calculando el valor promedio de la transformada de Fourier para
cada elemento: m, n no son cualquier valor */
nep = (360/precision + 1);

```

```

m = cvCreateImage(cvSize((int)nep,1), IPL_DEPTH_32F, 1);
n = cvCreateImage(cvSize((int)nep,1), IPL_DEPTH_32F, 1);

float *ptt = (float *)(m->imageData);
float *ptn = (float *)(n->imageData);
for(i=0; i < (int)nep; i++){
    ptt[i] = -pi+i*pi*(float)precision/180;
    ptn[i] = (float)i;
}

image_Re1 = cvCreateImage( cvSize(height, width), IPL_DEPTH_32F,
1);
image_Im1 = cvCreateImage( cvSize(width, height), IPL_DEPTH_32F,
1);

for(i=0; i < N; i++){

    /* Transformada de Fourier de la imagen */
    g_fftshift_abs_DFT(reg[i], image_Re1, image_Im1);

    /*Se obtiene la funcion h_A*/
    h_A[i] = cvCreateImage(cvSize((int)nep,1),
IPL_DEPTH_32F, 1);
    promedio_dft(image_Re1, height, width, nep, h_A[i], ptt,
T, ix, d, ptn);
}

cvReleaseMat(&T);
cvReleaseMat(&ix);
cvReleaseImage(&image_Re1);
cvReleaseImage(&image_Im1);
cvReleaseImage(&m);
cvReleaseImage(&n);

/* Especificando las variables para el calculo de la dft */
H_AR[0] = cvCreateImage(cvSize((int)nep,1), IPL_DEPTH_32F, 1);
H_AI[0] = cvCreateImage(cvSize((int)nep,1), IPL_DEPTH_32F, 1);

/* Obteniendo la fft de h(f)=h_A1 para el calculo de la correlacion
*/
g_DFT1(h_A[0], H_AR[0], H_AI[0]);

/* Especificando el valor estiamdo de la primera imagen respecto de
su propia imagen */
cvSetReal2D(angle_est,0,0,0);

for(i=1; i < N; i++){

    /*Obteniendo la fft de h(f)=h_A2 para el calculo de la
correlacion*/
    H_AR[i] = cvCreateImage(cvSize((int)nep,1),
IPL_DEPTH_32F, 1);

```

```

        H_AI[i] = cvCreateImage(cvSize((int)nep,1),
IPL_DEPTH_32F, 1);
        g_DFT1Rev(h_A[i], H_AR[i], H_AI[i]);

        /*Calculo de la correlacion*/
        correlacion(H_AR[0], H_AI[0], H_AR[i], H_AI[i],
precision, nep, angle_est,i);
    }

    //cvReleaseImage(h_A);
    //cvReleaseImage(H_AR);
    //cvReleaseImage(H_AI);
    h_A.~vector();
    H_AR.~vector();
    H_AI.~vector();

}

void CPolares(CvMat *ra, CvMat *th, int &width, int &height){
    /* Definiendo las variables CvArray para la conversion de coordenadas
*/

    //GPU
    cv::gpu::GpuMat g_x;
    cv::gpu::GpuMat g_y;
    cv::gpu::GpuMat g_raT( height, width, CV_32FC1);
    cv::gpu::GpuMat g_thT( height, width, CV_32FC1);
    cv::gpu::GpuMat g_ra ( width, height, CV_32FC1);
    cv::gpu::GpuMat g_th ( width, height, CV_32FC1);

    /* Creacion de las coordenadas x e y */
    g_x.create(height,width,CV_32FC1);
    g_y.create(height,width,CV_32FC1);

    cv::Mat xx ;
    cv::Mat yy ;
    int i;

    xx.create(height,width,CV_32FC1);
    yy.create(height,width,CV_32FC1);

    cv::Mat qy(height,1,CV_32FC1);
    cv::Mat qx(1,width,CV_32FC1);

    //Ingresando las componentes x de la matriz
    for(i=0 ; i < width; i++)
    {
        qx.at<float>(0,i) = (float)-1+(float)i*(float)2/(float)width;
    }
}

```



```

cv::Mat tmp;
for(i=0 ; i < height; i++)
{
    tmp = xx.row(i) ;
    qx.copyTo(tmp);
    //Ingresando las componentes y de la matriz
    qy.at<float>(i,0) = (float)-1+(float)i*(float)2/(float)height;
}

for(i=0 ; i < width; i++)
{
    tmp = yy.col(i) ;
    qy.copyTo(tmp);
}

g_x.upload(xx);
g_y.upload(yy);
//Se procedera a hacer un cambio en el sistema de coordenadas de
cartesiano a polar

cv::gpu::cartToPolar(g_x, g_y, g_raT, g_thT);
g_x.~GpuMat();
g_y.~GpuMat();

cv::gpu::transpose(g_thT,g_th);
cv::gpu::transpose(g_raT,g_ra);

ArrtoIplImageRM((cv::Mat)g_th, th);
ArrtoIplImageRM((cv::Mat)g_ra, ra);
}

void Coord_cartesianas(cv::gpu::GpuMat &g_x,cv::gpu::GpuMat &g_y, int
&width,int &height ){
    cv::Mat xx ;
    cv::Mat yy ;
    int i;

    xx.create(height,width,CV_32FC1);
    yy.create(height,width,CV_32FC1);

    cv::Mat qy(height,1,CV_32FC1);
    cv::Mat qx(1,width,CV_32FC1);

    //Ingresando las componentes x de la matriz
    for(i=0 ; i < width; i++)
    {
        qx.at<float>(0,i) = (float)-1+(float)i*(float)2/(float)width;
    }

    cv::Mat tmp;
    for(i=0 ; i < height; i++)

```

```

    {
        tmp = xx.row(i) ;
        qx.copyTo(tmp);
    }

    //Ingresando las componentes y de la matriz
    for(i=0 ; i < height; i++)
    {
        qy.at<float>(i,0) = (float)-1+(float)i*(float)2/(float)height;
    }

    for(i=0 ; i < width; i++)
    {
        tmp = yy.col(i) ;
        qy.copyTo(tmp);
    }

    g_x.upload(xx);
    g_y.upload(yy);
}

void promedio_dft(IplImage *TDF_image_Re2,int &width, int &height, float
&len, IplImage *h_A, float *ptt, CvMat *T, CvMat *ix, double &d, float
*ptn){

    int ihigh, ilow, ik, i, j;
    float prom;
    float k1, k2;

    float *cl_img;          //Input Array

    //Numero de elementos en cada arreglo
    int elements = height*width;

    //Tamaño de los datos
    size_t datasize = sizeof(float)*elements;
    cl_img = (float*)malloc(datasize);

    /* Reordenando los elementos de la imagen en una sola fila */
    for(i=0 ; i < height ;i++)
    {
        float* ptr = (float*)(TDF_image_Re2->imageData +
i*TDF_image_Re2->widthStep);

        //for(j=0; j< width ;j++){
            memcpy( cl_img + i*width , ptr, width*sizeof(float) );
            //cl_img[j+i*width] = ptr[j] ;
        //}
    }
}

```

```

/*Calculando el valor de h(alfa) en funcion de la frecuencia, esto es
necesario para hallar la correlacion */
ilow=0;
ihigh=0;
ik=0;

float * pta = (float*)(h_A->imageData );
float * pT = (float*)(T->data.ptr );

for(i=0 ; i < (int)len ; i++){
    ik=ilow;
    prom=0;
    k1 = ( ptt[i] - (float)d );
    k2 = ( ptt[i] + (float)d );

    while( k1 > pT[ik] ){
        ik++;
    }

    ilow = ik;

    if ( ik<ihigh ){
        ik = ihigh;
    }

    while( k2 > pT[ik] ){
        ik++;
        if ((ik > elements) ||( pT[ik] >= 1000 )){
            break;
        }
    }
    ihigh = ik;

    if (ihigh -1 >ilow){
        g_mean(ihigh, ilow, cl_img, elements, ix, pta, i, ptn);
    }
    else {
        pta[i] = 0;
    }
}

}

void mean(int ihigh, int ilow, IplImage *img, CvMat *ix, IplImage *h, int k,
IplImage *n){

    /* Variables */
    int w;
    float prom;

    /* Valores iniciales */

```

```

    prom = 0;
    w = ilow;

    while( ihigh > w ){
        prom = prom + (float)(img->imageData)[
CV_MAT_ELEM(*ix,int,0,w) ];
        w++;
    }
    prom = prom/( (float)ihigh - (float)ilow );

    float* ptr = (float*)(h->imageData );

    ptr[ (int)cvGetReal2D(n,0,k) ] = prom;
}

void g_mean(int ihigh, int ilow, float *cl_img, int &elements, CvMat *ix,
float *h, int &k, float *ptn){

    /* Variables */
    int w;
    float prom;

    /* Valores iniciales */
    prom = 0;
    w = ilow;

    /* Datasize */
    size_t datasize = sizeof(float)*elements;

    while( ihigh > w ){
        prom = prom + cl_img[ CV_MAT_ELEM(*ix,int,0,w)]; ;
        w++;
    }
    prom = prom/( (float)ihigh - (float)ilow );

    h[ (int)ptn[k] ] = prom;
}

void correlacion(IplImage *H_AR, IplImage *H_AI, IplImage *H_2AR,
IplImage *H_2AI, float precision, float size, CvMat *angle_est, int index){

    /* Variables */
    IplImage *H_CR, *H_CI;
    int len, width;
    static const double pi = 3.14159265358979323846;

    /* Variables para la transformada inversa */
    IplImage * image_Re;
    IplImage * image_Im;

```

```

//Numero de elementos en cada arreglo
int elements = H_AR->width;
width = H_AR->width;
//Tamaño de los datos
size_t datasize = sizeof(float)*elements;

H_CR    = cvCreateImage( cvSize( (int)size, 1), IPL_DEPTH_32F, 1);
H_CI    = cvCreateImage( cvSize( (int)size, 1), IPL_DEPTH_32F, 1);
//Asignando las direcciones de memoria en el host
float* Re1 = (float*)(H_AR->imageData );
float* Im1 = (float*)(H_AI->imageData );
float* Re2 = (float*)(H_2AR->imageData );
float* Im2 = (float*)(H_2AI->imageData );
float* pt1 = (float*)(H_CR->imageData );
float* pt2 = (float*)(H_CI->imageData );

g_calc(Re1, Im1, Re2, Im2, pt1, pt2, elements);

/* Especificando las variables */
len    = (int)(60/precision+1);

/* Definiendo las variables para calcular la IDFT */
image_Re = cvCreateImage( cvSize((int)size,1), IPL_DEPTH_32F, 1);
image_Im = cvCreateImage( cvSize((int)size,1), IPL_DEPTH_32F, 1);

/* Calculo de la transformada inversa de H_C*/
g_DFT1Inv(H_CR, H_CI, image_Re, image_Im);

/* Seleccionando un intervalo en funcion de la precision */
MinMax(angle_est, index, precision, image_Re, len);

/* Eliminando Variables */
cvReleaseImage(&H_CR);
cvReleaseImage(&H_CI);
cvReleaseImage(&image_Re);
cvReleaseImage(&image_Im);
}

void MinMax(CvMat *angle_est, int &index, float &precision, IplImage
*image_Re,int &len){

    int i;

    CvPoint  minloc, maxloc;
    double   minval, maxval;
    IplImage * pf;

    pf          = cvCreateImage( cvSize( len, 1), IPL_DEPTH_64F, 1);
    /* Seleccionando un intervalo en funcion de la precision */

```

```

double *p_pf = (double *) (pf->imageData);
float *p_Re = (float *) (image_Re->imageData);
for(i=0; i < len; i++){
    p_pf[i] = p_Re[i+(int)(150/precision)] ;
}

/* Localizando el valor maximo y el angulo */
cvMinMaxLoc( pf, &minval, &maxval, &minloc, &maxloc, 0 );
cvSetReal2D(angle_est ,0 ,index , (maxloc.x-30/precision)*precision
);

cvReleaseImage(&pf);
}

void g_fftshift_abs_DFT(IplImage *src, IplImage *dst, IplImage *dstIm){

    std::vector<cv::gpu::GpuMat> entrada(2);
    cv::gpu::GpuMat g_complexIn1(src->height,src->width,CV_32FC2);
    cv::gpu::GpuMat g_complex(src->height,src->width,CV_32FC2);

    entrada[0].upload(src);
    entrada[0].convertTo(entrada[0],CV_32FC1,1,0);

    entrada[1].create(src->height,src->width,CV_32FC1);
    entrada[1].setTo(cv::Scalar(0));

    //caculo de la transformada discreta de fourier
    cv::gpu::merge(entrada,g_complexIn1);
    cv::gpu::dft(g_complexIn1 , g_complex , cv::Size(src->width,src-
>height) , CV_DXT_FORWARD);
    cv::gpu::split(g_complex,entrada);

    // Calculando la magnitud del espectro Mag = sqrt(Re^2 + Im^2)
    cv::gpu::pow(entrada[0],2.0,entrada[0]);
    cv::gpu::pow(entrada[1],2.0,entrada[1]);
    cv::gpu::add(entrada[0],entrada[1],entrada[0]);
    cv::gpu::pow(entrada[0],0.5,entrada[0]);

    //Desplazando la frecuencia fundamental al centro
    gr_cvShiftDFT(entrada[0],entrada[0]);
    cv::gpu::GpuMat dstt(src->width,src->height,CV_32FC1);
    cv::gpu::transpose(entrada[0],dstt);

    //Copiando al formato IplImage

    ArrtoIplImageR( (cv::Mat)dstt, dst);
    ArrtoIplImageR( (cv::Mat)entrada[1], dstIm);

    entrada.~vector();
}

```

```

void gr_cvShiftDFT(cv::gpu::GpuMat &src_arr, cv::gpu::GpuMat &dst_arr ){
    int cx, cy;

    cx = src_arr.cols;
    cy = src_arr.rows;

    cv::gpu::GpuMat q1 = src_arr.operator()( cv::Rect(0,0,cx/2,cy/2) );
    cv::gpu::GpuMat q2 = src_arr.operator()( cv::Rect(cx/2,0,cx/2,cy/2)
);
    cv::gpu::GpuMat q3 = src_arr.operator()(
cv::Rect(cx/2,cy/2,cx/2,cy/2) );
    cv::gpu::GpuMat q4 = src_arr.operator()( cv::Rect(0,cy/2,cx/2,cy/2)
);

    cv::gpu::GpuMat tmp(cy,cx,src_arr.type());

    q3.copyTo(tmp);
    q1.copyTo(q3);
    tmp.copyTo(q1);
    q4.copyTo(tmp);
    q2.copyTo(q4);
    tmp.copyTo(q2);
}

void factor1(CvMat *A, int i, int j){
    /* Constante pi*/
    const double pi = 3.141592653589793;

    if ( cvGetReal2D(A,i,j) < -pi + 0.0000001 ){
        cvSetReal2D( A, i, j, -cvGetReal2D(A,i,j) );
    }

    if ( cvGetReal2D(A,i,j) > pi + 0.0000001 ){
        cvSetReal2D( A, i, j, cvGetReal2D(A,i,j)-(double)2*pi);
    }

    if ( cvGetReal2D(A,i,j) < -pi - 0.0000001 ) {
        cvSetReal2D( A, i, j, cvGetReal2D(A,i,j)+(double)2*pi);
    }
}

void g_DFT1(IplImage *src, IplImage *dst, IplImage *dstIm){
    std::vector<cv::gpu::GpuMat>entrada(2);
    cv::gpu::GpuMat g_complexIn1(src->height,src->width,CV_32FC2);
    cv::gpu::GpuMat g_complex(src->height,src->width,CV_32FC2);

    entrada[0].upload(src);

```

```

    entrada[0].convertTo(entrada[0],CV_32FC1,1,0);
    entrada[1].create(src->height,src->width,CV_32FC1);
    entrada[1].setTo(0);

    //calculo de la transformada discreta de fourier
    cv::gpu::merge(entrada,g_complexIn1);
    cv::gpu::dft(g_complexIn1 , g_complex , cv::Size(src->width,src-
>height) , CV_DXT_FORWARD);
    cv::gpu::split(g_complex,entrada);

    //Copiando al formato IplImage
    ArrtoIplImageR( (cv::Mat)entrada[0], dst);
    ArrtoIplImageR( (cv::Mat)entrada[1], dstIm);

    entrada.~vector();
}

void g_DFT1Rev(IplImage *src, IplImage *dst, IplImage *dstIm){

    /* Variables */
    int i,w;
    IplImage * realIn1;

    std::vector<cv::gpu::GpuMat>entrada(2);
    cv::gpu::GpuMat g_complexIn1(src->height,src->width,CV_32FC2);
    cv::gpu::GpuMat g_complex(src->height,src->width,CV_32FC2);

    /* especificando las variables */
    realIn1 = cvCreateImage( cvGetSize(src), IPL_DEPTH_32F, 1);
    w = src->width;

    /* Invirtiendo el orden de los elementos de la matriz src */
    float* ptr = (float*)(realIn1->imageData );
    float* gsrc = (float*)(src->imageData );
    for(i=0; i<w ;i++){
        ptr[w-i-1] = gsrc[i];
    }

    entrada[0].upload(realIn1);
    entrada[0].convertTo(entrada[0],CV_32FC1,1,0);

    /* Poniendo a cero la parte imaginaria de la entrada */
    entrada[1].create(src->height,src->width,CV_32FC1);
    entrada[1].setTo(0);

    /* Ingresando los valores de complexIn1 */
    cv::gpu::merge(entrada,g_complexIn1);

    /* Calculando la DFT de complexIn1 */
    cv::gpu::dft(g_complexIn1 , g_complex , cv::Size(src->width,src-
>height) , CV_DXT_FORWARD);

    // Dividiendo la DFT en su parte real e imaginaria

```



```

cv::gpu::split(g_complex, entrada);

//Copiando al formato IplImage
ArrtoIplImageR( (cv::Mat)entrada[0], dst);
ArrtoIplImageR( (cv::Mat)entrada[1], dstIm);

cvReleaseImage(&realIn1);
entrada.~vector();
}

void g_DFT1Inv(IplImage *src, IplImage *srcIm, IplImage *dst, IplImage
*dstIm){

    /* Variables */
    std::vector<cv::gpu::GpuMat>entrada(2);
    cv::gpu::GpuMat g_complexIn1(src->height, src->width, CV_32FC2);
    cv::gpu::GpuMat g_complex(src->height, src->width, CV_32FC2);

    /* Especificando las variables */
    entrada[0].upload(src);

    entrada[1].upload(srcIm);

    /* ingresando los valores de complexIn1 */
    cv::gpu::merge(entrada, g_complexIn1);

    // Dividiendo la IDFT en su parte real e imaginaria
    cv::gpu::split(g_complex, entrada);

    //Copiando al formato IplImage
    ArrtoIplImageR( (cv::Mat)entrada[0], dst);
    ArrtoIplImageR( (cv::Mat)entrada[1], dstIm);

    entrada.~vector();
}

void ArrtoIplImageR(cv::Mat src, IplImage *dst){
    IplImage dst1 = src;
    cvScale(&dst1, dst, 1, 0);
}

void ArrtoIplImageRM(cv::Mat src, CvMat *dst){
    IplImage dst1 = src;
    cvScale(&dst1, dst, 1, 0);
}
}

```

C. Resultados de programas en C++ ejecución en IBP

C.1 Resultados de ejecución con IBP #1

```
1 main
  2 main
    3 main
      Ingresando las imagenes del registro
2
PARAMETROS DE ROTACION:
Angulo[0] : 0.0000000000000000
Angulo[1] : 0.000001192092897
Angulo[2] : 0.000001192092897
Angulo[3] : 0.000001192092897
3

PARAMETROS DE DESPLAZAMIENTO:
x: 0.0000000000000000 , y: 0.0000000000000000
x: -0.0000000000000000 , y: -0.0000000000000000
x: -0.0000000000000000 , y: -0.0000000000000000
x: -0.0000000000000000 , y: -0.0000000000000000
4

ITERATIVE BACK PROJECTING:
5
  6

Delta: 0.013807
Delta: 0.009576
Delta: 0.006647
Delta: 0.004618
Delta: 0.003211
Delta: 0.002234
Delta: 0.001555
Delta: 0.001084
Delta: 0.000755
```

```
Delta: 0.000527
Delta: 0.000368
Delta: 0.000257
Delta: 0.000180
Delta: 0.000126
Delta: 0.000088 7
  4 main
  5 main
```

C.2 Resultados de ejecución con IBP #2

```
1 main
  2 main
    3 main
      Ingresando las imagenes del registro
        2
          PARAMETROS DE ROTACION:
            Angulo[0] : 0.0000000000000000
            Angulo[1] : -1.199998855590818
            Angulo[2] : -2.549998909234999
            Angulo[3] : -21.299999654293057
          3
            PARAMETROS DE DESPLAZAMIENTO:
              x: 0.0000000000000000 , y: 0.0000000000000000
              x: -0.136224554700138 , y: -0.276165906144096
              x: -0.107101865527382 , y: 0.046018516813524
              x: 0.237792818561437 , y: 0.331210616578946
            4
              ITERATIVE BACK PROJECTING:
                5
                  6
                    Delta: 0.030502
                    Delta: 0.022505
                    Delta: 0.016639
                    Delta: 0.012357
                    Delta: 0.009246
                    Delta: 0.007018
                    Delta: 0.005508
                    Delta: 0.004568
                    Delta: 0.004163
                    Delta: 0.004068
                    Delta: 0.004160
```

```
Delta: 0.004111 7
```

```
  4 main
```

```
  5 main
```