

Universidad de las Ciencias Informáticas

Facultad 3

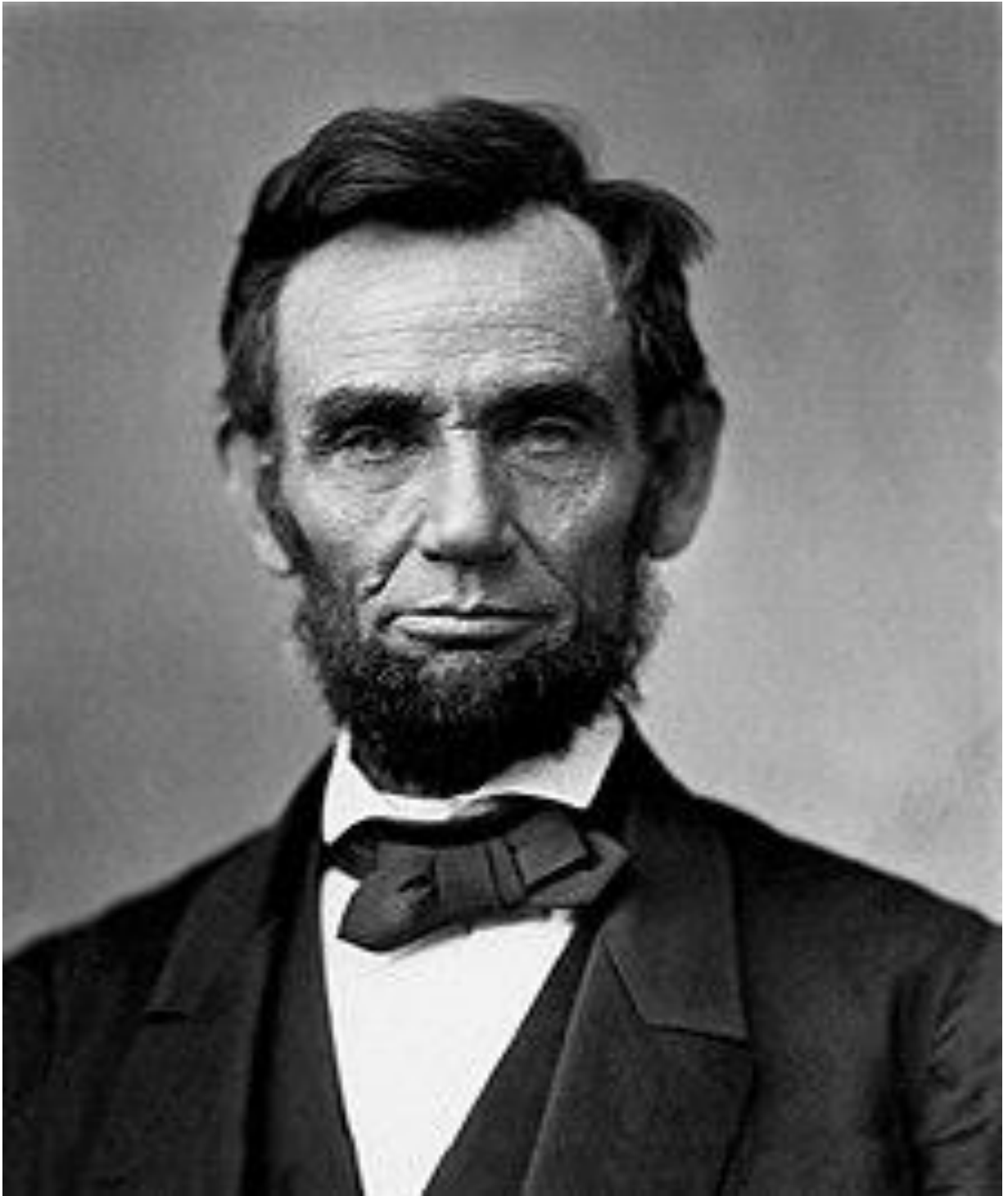


Título: Algoritmo de superresolución para la restauración de imágenes de opacidad de la cápsula posterior en pacientes operados de catarata.

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor(es): Robinson Machado Díaz

Tutor(es): Michel Álvarez Cancio



La mejor forma de predecir el futuro es crearlo. Abraham Lincoln

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Robinson Machado Díaz

Firma del Autor

Michel Álvarez Cancio

Firma del Tutor

DATOS DE CONTACTO

AGRADECIMIENTOS

Agradezco primeramente a toda mi familia por estar presente en todo momento de mi vida, por darme las fuerzas para cumplir todas mis metas y hoy ser ingeniero informático. Especialmente le agradezco a mi abuela Victoria por ser el pilar donde sostengo toda mi vida, por ser en todo momento mi mamá, mi papá y sobre todo mi gran abuela. Sin ella no hubiese llegado tan lejos ni tan bien. También agradezco enormemente a mi mamá Anay y a mi papa Javier por todo lo que han hecho por mí en estos 24 años, a mi tío Humberto por ser mi mayor ejemplo a seguir y ser parte de mi vida en todo momento, por apoyarme en todas mis decisiones y siempre aconsejarme a ser lo mejor en todos los casos. A toda mi familia en general.

Agradezco a todos los compañeros que de una forma u otra han tenido que ver en mi vida, en especial a los que conocí aquí en la UCI, en especial a Pablo, Cristófer, Ariandi, Nadir, Yudiel, Dasiel, Frank y muchos otros que no

alcanzarían las 80 hojas para mencionar. A los que fueron amigos de estudio, de fiestas, de buenos momentos y de malos. A todos los profesores que contribuyeron a mi desarrollo intelectual y personal.

A mi tutor Michel por haberme acogido desde 3er año y por siempre estar disponible para ayudarme en el transcurso de esta tesis. Al tribunal por todos los palos en cada corte que poco a poco hicieron de esta una tesis de gran calidad, a Reyder tanto por ser un excelente profesor, como por ser un gran oponente.

DEDICATORIA

A mi familia por todo su apoyo en estos 5 años, en especial al pilar de mi vida mi abuela querida. Sin ella nada hubiese podido alcanzar y ser hoy ingeniero seria solo un sueño.

RESUMEN

En la actualidad la enfermedad de cataratas es una de las principales causas de ceguera en el mundo, estimándose que cuenta con la mitad de los casos de ceguera en la población mundial. Un elemento importante en la cirugía de cataratas es la opacidad de la cápsula posterior, ya que sigue siendo la complicación postoperatoria más frecuente a largo y mediano plazo. Unas de las imágenes empleadas en la identificación de la misma son las imágenes en retroiluminación provenientes de la lámpara de hendidura. Estas presentan una desventaja como resultado de la reflexión de la luz proveniente del artefacto, por lo que se hace necesario realizar un procesamiento a dichas imágenes para obtener resultados más favorables en el diagnóstico del especialista. Para ello se diseñó un algoritmo de superresolución con el objetivo de mejorar la calidad de estas imágenes provenientes de la lámpara de hendidura, mejorando la resolución espacial de las mismas, en dicho procesamiento se utilizó la Transformada Discreta del Coseno para trabajar la superresolución en el dominio de la frecuencia. Como método de interpolación intermedia se utilizó la Interpolación Bicúbica y luego se le aplicaron algunos filtros para eliminar el ruido que puede producir la estimación de nuevos píxeles. Al resultado obtenido se le aplica el detector de bordes Canny para medir el porcentaje de bordes como indicador de calidad, evidenciando un aumento aproximado de 1,8% más de píxeles blancos en la imagen procesada en comparación con la imagen de baja resolución demostrando que las imágenes resultantes no solo son de mayores dimensiones que la original, sino que presentan mayor nivel de detalles.

PALABRAS CLAVE

Opacidad de la cápsula posterior, superresolución, imagen en retroiluminación, Transformada Discreta del Coseno, resolución.

ÍNDICE

INTRODUCCIÓN	10
CAPÍTULO 1: FUNDAMENTOS DE LA INVESTIGACIÓN, METODOLOGÍA Y TECNOLOGÍAS EMPLEADAS.	15
1.1 Procesamiento digital de imágenes: principales conceptos.	15
1.2 Superresolución.....	18
1.2.1. Técnicas de superresolución.	20
1.2.2 Superresolución basada en la interpolación.	20
1.2.3 SR vía deformación de imagen (Warping).	22
1.2.4 SR basada en la Teoría de Conjuntos Convexos.....	22
1.2.5 SR utilizando Máximo a Posteriori (MAP).....	23
1.2.6 SR en el dominio de la frecuencia.	24
1.2.6.1 SR utilizando la Transformada Coseno Discreta.	26
1.3 Métodos seleccionados.	28
1.4 Herramientas y metodologías utilizadas para el desarrollo.	29
1.4.1 Metodología de desarrollo de software.	29
1.5 Lenguajes utilizados.....	30
1.5.1 Lenguaje de programación Java.....	31
1.5.2 Entorno Integrado de desarrollo.....	32
1.5.3 Herramienta para el modelado.	32
1.6 Conclusiones del capítulo.	32
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN	34
2.1 Descripción de la solución.	34
2.2 Fase de planeación.	36
2.2.1 Especificación de los requisitos.	36
2.2.2 Descripción de historias de usuario.	37
2.2.3 Fase de planificación de la entrega.	38
2.2.4 Plan de iteraciones.	39
2.2.5 Plan de entrega.	41
2.3 Diseño de la solución.	41

2.3.1 Tarjetas CRC.....	41
2.3.2 Estándares de codificación.....	43
2.3.3 Patrones de diseño utilizados.....	43
2.4 Conclusiones del capítulo.....	44
CAPÍTULO 3: VALIDACIÓN Y RESULTADOS DEL SISTEMA.....	46
3.1 Fase de implementación.....	46
3.2 Interfaz del sistema.....	48
3.2.1 Resultados al aplicar el algoritmo.....	50
3.3 Validación de los resultados.....	50
3.4 Verificación del sistema.....	53
3.4.1 Estrategia de pruebas.....	53
3.4.2 Pruebas unitarias.....	54
3.4.3 Pruebas de aceptación.....	57
3.5 Conclusiones del capítulo.....	63
Conclusiones.....	64
RECOMENDACIONES.....	65
REFERENCIAS BIBLIOGRÁFICAS.....	66
ANEXOS.....	¡ERROR! MARCADOR NO DEFINIDO.
Anexos#1.....	¡Error! Marcador no definido.
Anexos#2.....	¡Error! Marcador no definido.

ÍNDICE DE FIGURAS

Figura 1. Procesamiento de imagen	16
Figura 2 Resultados del incremento de la resolución espacial de una imagen.....	17
Figura 3 Resolución espacial	17
Figura 4: Proceso simple de superresolución.	19
Figura 5: Interpolación Bicúbica. (Izquierda la imagen original y la derecha la resultante)	22
Figura 6: Resultados de la aplicación de la Transformada de Fourier	25
Figura 7 Proyecciones 1D de la imagen original y la de referencia.....	26
Figura 8 Matriz de 8x8 con los 64 valores base de la DCT.....	26
Figura 9 Descripción del algoritmo de SR propuesto.....	34
Figura 10: Clase Imagen.....	44
Figura 11: Creación de instancias.....	44
Figura 12: Interfaz principal del sistema.....	49
Figura 13: Resultado de aplicar el algoritmo.	50
Figura 14: Máscara binomial de 3x3	52
Figura 15: Resultados de la cuantificación de los bordes al aplicar Canny.	53
Figura 16: Algoritmo para convertir de RGB a escala de grises de la imagen de entrada.	55
Figura 17: Diagrama de flujo del algoritmo “escalaGrises”	55
Figura 18: Resultados de las pruebas de aceptación	62
Figura 19 Tratamiento de las no conformidades.	63

ÍNDICE DE TABLAS

Tabla 1. Historia de Usuario#1.....	37
Tabla 2: Estimación de tiempo por historias de usuario.....	38
Tabla 3: Plan de duración de las iteraciones.....	40
Tabla 4: Plan de duración de las iteraciones.....	41
Tabla 5: Tarjeta CRC de la clase NewJFrame.....	42
Tabla 6: Tarjeta CRC de la clase ProcesamientoImagen.....	42
Tabla 7: Tiempo de implementación de las HU de la iteración #1.....	46
Tabla 8: Tarea de ingeniería de la funcionalidad Cargar imagen de BR.....	47
Tabla 9: Tiempo de implementación de las HU de la iteración #2.....	47
Tabla 10: Tiempo de implementación de las HU de la iteración #3.....	48
Tabla 11: Caminos básicos.....	56
Tabla 12: Caso de prueba del camino básico aplicado al método escalaGrisés.....	57
Tabla 13: Caso de prueba de aceptación HU#1.....	58
Tabla 14: Caso de prueba de aceptación HU#2.....	58
Tabla 15: Caso de prueba de aceptación HU#3.....	59
Tabla 16: Caso de prueba de aceptación HU#6.....	59
Tabla 17: Caso de prueba de aceptación HU#9.....	60
Tabla 18: Caso de prueba de aceptación HU#10.....	60
Tabla 19: Caso de prueba de aceptación HU#11.....	61
Tabla 20: Resultados de las pruebas de aceptación.....	61

INTRODUCCIÓN

El lente, o cristalino del ojo, refracta los rayos de luz que ingresan en él para ayudarnos a ver. Este debe ser totalmente transparente. Una de las enfermedades que atenta contra el cristalino es la catarata.

La catarata corresponde a una opacificación que se forma en el lente del ojo (cristalino) y que causa la pérdida de la visión. El lente del ojo centra la luz, ajusta la visión para que pueda ser clara y la lleva hacia la retina, membrana que se encuentra en la parte posterior del ojo, lugar donde la imagen se graba y de ahí va al cerebro a través del nervio óptico. (1)

Existen muchas causas para la formación de cataratas como el envejecimiento (la proteína del lente cambia a causa del uso y se rasga, siendo la causa más común), hábito de fumar, diabetes mellitus, infecciones, lecciones traumáticas, exposición a la radiación y defectos congénitos. También hay factores de riesgo que aumentan la probabilidad que una persona sufra de cataratas como la edad, la exposición a la radiación de los rayos solares ultravioletas y padecer de diabetes. (2)

El tratamiento de la catarata es quirúrgico. La cirugía de catarata es una operación para extraer el cristalino del ojo cuando este está nublado. Una cirugía de catarata no restaura la visión perdida a causa de condiciones oculares como la degeneración macular, el glaucoma, o la retinopatía diabética. Durante la cirugía de catarata, se le extrae el cristalino nublado natural y se le coloca un cristalino transparente artificial, ese cristalino se denomina lente intraocular (IOL, por sus siglas en inglés). (2)

La opacidad de la cápsula posterior (OCP) es actualmente uno de los aspectos más importantes en la cirugía de catarata de los tiempos modernos. Sigue siendo la complicación posoperatoria tardía más frecuente tras la cirugía de catarata asociada con disminución de la agudeza visual, deterioro de la sensibilidad al contraste y problemas con deslumbramiento que conllevan importantes repercusiones sociales, médicas y económicas. (3). Después de un análisis de 90 estudios publicados por la Revista Cubana de Oftalmología en la actualidad mostró que su incidencia se encuentra entre el 0,7% y 47,6% en los primeros 5 años de la cirugía (3)(4), siendo en Cuba una cifra a considerar ya que asciende hasta el 50% de los casos (2). Esto, pudiera

atribuirse a los diferentes criterios de selección de la muestra, sistemas de evaluación de la opacidad, edad, tipo de lente intraocular y técnica quirúrgica (4).

El Instituto Cubano de Oftalmología Ramón Pando Ferrer constituye el centro de referencia tanto nacional como internacional de la especialidad Oftalmología y el centro rector de la docencia contando con prestigiosos especialistas. Ahí son atendidos todos los servicios oftalmológicos que una persona pueda presentar.

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el grupo de investigación Artificial Intelligence Research and Innovation (AIRI), donde se ha desarrollado en conjunto con el instituto cubano de oftalmología un software para la cuantificación de opacidad de la cápsula posterior (PANDOC).

El software PANDOC provee al oftalmólogo de una herramienta por medio de la cual este es capaz de cuantificar numéricamente y detectar diferencias de opacidad (a veces imperceptibles para el ojo humano), logrando así una evaluación objetiva del grado de opacidad, minimizando el sesgo de observación entre un médico y otro (5). La precisión del resultado de la cuantificación está sujeta a los parámetros de calidad de la imagen, uno de estos parámetros que influye son las dimensiones de la imagen. La ausencia de detalles y niveles altos de ruidos o desenfoque dificultan la realización de un diagnóstico adecuado por el oftalmólogo, debido a que, en correspondencia con su nivel de experticia, será el diagnóstico emitido.

Por tal motivo la presente investigación trata acerca de la creación de un algoritmo de superresolución (SR) que realice una restauración de los detalles de las imágenes en retroiluminación provenientes de la lámpara de hendidura y que presente bajos tiempos de ejecución comparado con el algoritmo desarrollado dentro del propio grupo de investigación, que debido a su alto tiempo de cómputo no resultó llamativo para los oftalmólogos su uso en las consultas.

A partir de la problemática descrita anteriormente se deriva el siguiente **problema a resolver**.

¿Cómo restaurar detalles utilizando un algoritmo de superresolución en las imágenes en retroiluminación provenientes de la lámpara de hendidura?

De acuerdo al problema planteado se define como **objeto de estudio**: Restauración de imágenes digitales, delimitando el **campo de acción** hacia algoritmos de Superresolución en imágenes digitales. El **objetivo general** es desarrollar un algoritmo de superresolución para la restauración de detalles en imágenes de retroiluminación provenientes de la lámpara de hendidura.

Para dar cumplimiento al objetivo general se proponen los siguientes **Objetivos**:

- Elaborar el marco teórico de la investigación mediante el estudio de las técnicas actuales de superresolución en imágenes digitales.
- Identificar los requisitos funcionales y no funcionales, diseñar la propuesta de solución e implementar los elementos del diseño obtenidos para el desarrollo de la propuesta de solución.
- Validar la solución propuesta mediante pruebas de software y métricas de calidad de imagen.

Se plantea la siguiente **idea a defender**: Con el desarrollo de un algoritmo de superresolución es posible restaurar detalles en las imágenes de opacidad de la cápsula posterior provenientes de la lámpara de hendidura.

Con el propósito de dar cumplimiento al objetivo propuesto, se plantean las siguientes **tareas de investigación**:

1. Identificación de las diferentes técnicas de procesamiento de imágenes médicas.
2. Revisión de las técnicas de filtrado y mejora en imágenes médicas.
3. Estudio del uso de algoritmos de interpolación en imágenes digitales.
4. Estudio del uso de algoritmos de superresolución en imágenes digitales.
5. Descripción de los pasos a seguir para el uso del algoritmo propuesto.
6. Diseñar la solución propuesta en función de los requisitos especificados.
7. Implementación de las técnicas de filtrado y mejora de la imagen.
8. Integración al software PANDOC.
9. Validación de los resultados comparado con el algoritmo desarrollado por el grupo de investigación.

10. Validación de la implementación del sistema a partir de la aplicación de pruebas unitarias y funcionales.

Durante el desarrollo de la investigación es necesario utilizar métodos científicos, los cuales se mencionan a continuación.

Métodos empíricos:

- **Observación:** Permite realizar la observación y estudio de los diferentes algoritmos aplicados a diferentes muestras, identificando así cuáles eran los más adecuados para la obtención de los mejores resultados.
- **Experimento:** A partir de este método se pudo probar las distintas técnicas y métodos de superresolución en imágenes médicas.
- **Entrevista:** Proporciona datos importantes acerca de las necesidades de los especialistas en la identificación de las regiones con opacidad.

Métodos teóricos:

- **Analítico-Sintético:** La aplicación de este método permitió caracterizar los algoritmos de superresolución y a partir de esto se seleccionaron los elementos claves para el desarrollo de este trabajo y así comprender la problemática existente.
- **Análisis documental:** Permite realizar el estudio bibliográfico como aspecto esencial en la conformación del marco teórico referencial de la investigación, como sustento de las valoraciones realizadas, así como el estudio de los contenidos relacionados con la identificación de la OCP en pacientes operados de cataratas y principales técnicas de superresolución de imágenes médicas digitales.
- **Análisis histórico-lógico:** Posibilitó la aproximación a la evolución de los referentes teóricos del tema, analizar diferentes criterios relacionados con la trayectoria y evolución de software de análisis de OCP en pacientes operados de catarata, para poder comprender el nivel de desarrollo alcanzado por este tema en el mundo.

La presente investigación está estructurada por los siguientes capítulos; a continuación se describe el objetivo principal de cada uno de ellos:

Capítulo 1. “Fundamentos de la investigación, metodología y tecnologías empleadas ”: en este capítulo se abordaran de manera sintetizada los elementos básicos teóricos. También se expondrá brevemente los algoritmos de SR presentes en la literatura consultada. Se describen las herramientas y lenguajes, así como la metodología de desarrollo de software.

Capítulo 2. “Propuesta de solución”: en este capítulo se describe la solución de la problemática, se especifican los requisitos funcionales del software levantados durante el desarrollo de la solución. Así como los artefactos generados en esta etapa.

Capítulo 3. “Validación y resultados del sistema”: en este capítulo se exponen y se validan los resultados obtenidos, una vez procesadas las imágenes se utiliza el detector de bordes Canny, como métrica de calidad. Se realizan las pruebas de validación del sistema mediante las pruebas de caja negra y caja blanca.

CAPÍTULO 1: FUNDAMENTOS DE LA INVESTIGACIÓN, METODOLOGÍA Y TECNOLOGÍAS EMPLEADAS.

Son analizados en este capítulo los fundamentos teóricos que constituyen la base sobre la cual se realiza este trabajo investigativo, se definen conceptos que giran alrededor del objetivo general de la investigación y que permiten sintetizar y relacionar los temas vinculados con el problema a resolver. Además se analizan algunas de las soluciones existentes y posibles herramientas a utilizar para el desarrollo de la aplicación.

1.1 Procesamiento digital de imágenes: principales conceptos.

El procesamiento digital de imágenes (PDI) se refiere a los procesos cuyas entradas y salidas son imágenes, extrayendo atributos de las mismas, e incluyendo el reconocimiento de objetos individuales. Dichos procesos se llevan a cabo con la ayuda de una computadora. Se definirá como el conjunto de técnicas y métodos cuya finalidad será mejorar la calidad de la información contenida en las imágenes para su posterior visualización. (6)

En los últimos años han surgido varias herramientas de software capaces de expresar el grado de incidencia de OCP. La ventaja principal de estos sistemas es que reducen la variabilidad del observador y aumenta la exactitud, aunque hay momentos en que no son los más favorables ya que estos sistemas trabajan con imágenes obtenidas de las lámparas de hendidura, dichas imágenes poseen una desventaja muy peculiar al presentar una mancha pronunciada de luz o reflejo especular, y son denominadas imágenes de Purkinje (7) (8), a las cuales se hace necesario realizarles un proceso de mejora mediante un tratamiento para imágenes digitales, con el objetivo de obtener una imagen de alta resolución (AR), y dar cumplimiento a los objetivos propuestos.

Para garantizar un mejor entendimiento se exponen a continuación los principales conceptos relacionados con el procesamiento digital de imágenes.

- **Imagen digital:** una imagen se representa por una función en dos dimensiones $f(x,y)$, cuyo valor corresponde a la intensidad de luz en cada punto del espacio de las coordenadas (x,y) . En el caso de una imagen monocromática, al valor de $f(x,y)$ se le denominará nivel de gris en el punto de coordenadas (x,y) . Una imagen puede ser considerada como una señal bidimensional. Si el dominio (valores de (x,y)) y el rango

(valores de $f(x,y)$) son continuos, la imagen es continua o análoga; si el dominio y el rango son discretos, la imagen es digital. Por lo tanto una imagen en blanco y negro puede ser representada por una matriz de dimensión $M \times N$. Además de la representación en blanco y negro, otras opciones son las imágenes en tonos de grises, para las cuales $f(x, y)$ representa un nivel de intensidad típicamente asociado a un entero entre 0 y 255. (9) Ver Fig. (1).

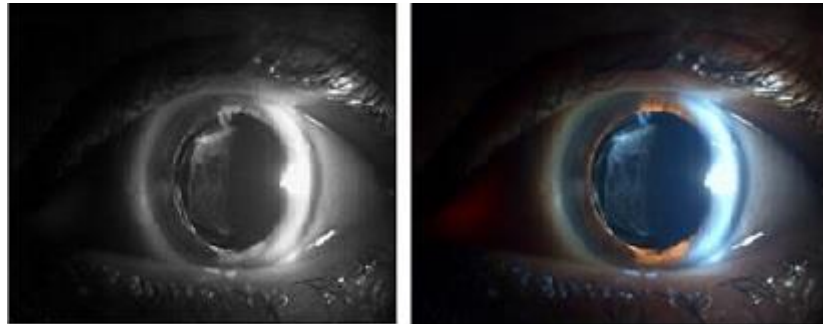


Figura 1. Procesamiento de imagen

- **Muestreo (sampling):** La conversión de las coordenadas a un dominio discreto está asociada al concepto de muestreo (píxeles). El muestreo es la conversión que sufren las dos dimensiones de la señal analógica (imagen continua), y la que genera la noción del pixel (10) (11).

Las dos principales causas de pérdida de información cuando se captura una imagen digital son la naturaleza discreta de los píxeles y el rango limitado de los valores de intensidad luminosa que se puede tener en cada pixel. De ahí surge el concepto de resolución de una imagen: una parte asociado al número de píxeles y otra parte a los niveles de gris.

- **Resolución de una imagen:** indica el número de píxeles que contiene una imagen por área. Se suele medir en píxeles por pulgadas o píxeles por centímetro. Es uno de los aspectos más importantes a tener en cuenta, dado que es directamente proporcional con la calidad de la imagen, por tanto a mayor resolución, mayor será la calidad de la imagen. Esto tiene como inconveniente el aumento del espacio en disco

del archivo gráfico (9) .Por lo general cuanto más detalles tenga una imagen, mayor será su resolución. En la Fig. 2 se muestran resultados del aumento de resolución espacial de una imagen, como la primera muestra es de 32 píxeles y luego evoluciona hasta 128 píxeles.



Figura 2 Resultados del incremento de la resolución espacial de una imagen.

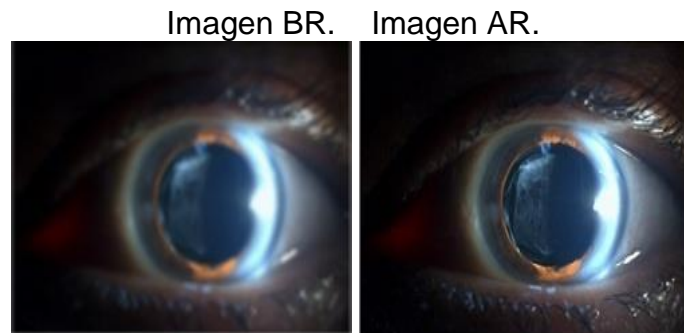


Figura 3 Resolución espacial

- **Captura de imágenes digitales:** La idea general de la captura de una imagen, es llevar la imagen dentro de la computadora, donde pueda ser almacenada o visualizada para luego ser manipulada y mejorada. El elemento principal en la adquisición de una imagen es una cámara que captura las imágenes de un objeto.
- **Lámpara de hendidura:** es un dispositivo óptico que se utiliza en oftalmología y optometría para examinar las varias partes del ojo. (12) Consiste en un microscopio binocular dotado de una fuente de luz que permite ver en tres dimensiones y con una amplificación que oscila entre 6 y 40 aumentos, las estructuras del polo anterior del

ojo. La luz se proyecta como una franja o hendidura brillante, lo que permite el examen detallado del ojo en pequeños segmentos. Se trata de una tecnología esencial en oftalmología con la que es posible detectar la posición, forma y profundidad de cualquier anomalía en la estructura del ojo. (13)

En el procesamiento digital de imágenes es importante el uso de la Superresolución, ya que con esta se logra resultados satisfactorios expuestos a continuación.

1.2 Superresolución.

En este trabajo de diploma usaremos el término superresolución para describir el proceso de obtención de una imagen o de una secuencia de imágenes observadas de AR a partir de una imagen observada de baja resolución (BR) (14).

La superresolución se basa en la teoría de información, la cual dice que “la información no se hurta” entonces, básicamente la información no redundante de las imágenes de BR es usada para construir una imagen de SR.

En el proceso de SR las imágenes de BR son registradas y almacenadas. El resultado es una imagen compuesta de muestra, estos pixeles son interpolados y vueltos a muestrear en la retícula de AR. Una forma sencilla del proceso es mostrada en la Figura 4:

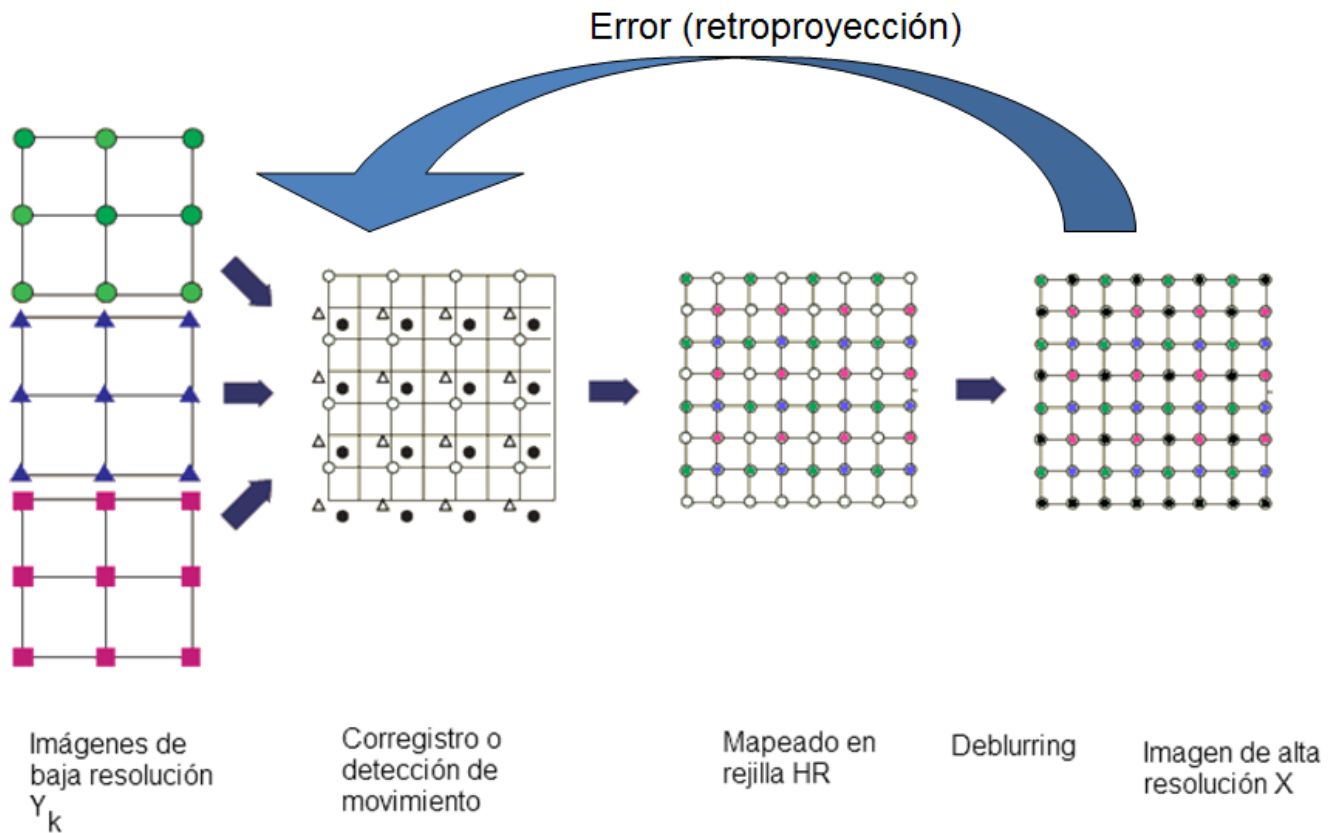


Figura 4: Proceso simple de superresolución.

Como se mencionó, dado una o un conjunto de imágenes de BR su camino hacia la superresolución involucra mínimo 3 pasos (15):

- Adquisición
- Registro
- Proyección de los valores de imagen de baja resolución en valores de alta resolución.

Por lo general la mayoría de las aplicaciones que obtienen imágenes de AR suelen estar basadas en el análisis y procesamiento posterior de las imágenes adquiridas. La mayor calidad de las imágenes de AR facilita la interpretación humana (en este caso al oftalmólogo) y ayuda a su percepción automática.

Dentro de las técnicas de SR podemos distinguir entre las técnicas de SR basadas en múltiples observaciones desplazadas entre sí, que suelen denominarse como SR basada en el movimiento, y las técnicas de SR libre de movimiento (16) donde la SR está basada en los cambios del emborronamiento presente en las observaciones o en los cambios de posición de las fuentes lumínicas.

La SR basada en el movimiento aporta técnicas para reconstruir las imágenes de AR a partir de un conjunto de imágenes observadas de BR, incrementando el ancho de banda de la imagen y eliminando la degradación provocada por el proceso de captura de las imágenes observadas de BR. La idea básica de la SR consiste en combinar la información no redundante en las múltiples imágenes observadas de BR, para generar una imagen de AR. La presencia de la información no redundante en las múltiples observaciones de BR se debe a los desplazamientos a nivel de subpixel entre las imágenes captadas. (17) (18) (19)

1.2.1. Técnicas de superresolución.

Desde el trabajo seminal de Tsai y Huang (20) la SR ha sido uno de los campos de investigación más activos en las dos últimas décadas (véase (21) (17) (22)). La SR ha sido abordada desde el dominio de la frecuencia, el dominio espacial y la perspectiva del procesamiento de señales hasta la del aprendizaje automático.

Básicamente todas las técnicas se basan en la idea de que disponiendo de la información de una señal, podemos recuperar toda la información que se pierde en el momento de su captura o adquisición (19). El grado de esta pérdida depende de la resolución o capacidad de muestreo del dispositivo (19) (23) que hace la adquisición de la señal, en este caso una imagen.

1.2.2 Superresolución basada en la interpolación.

Estos algoritmos se basan en el proceso de calcular valores numéricos desconocidos (píxeles) a partir de otros conocidos mediante el uso de un polinomio interpolador. Dicho polinomio asigna pesos a la vecindad de cada píxel cuyo valor se desconoce. Los algoritmos más utilizados son los siguientes: vecinos más cercanos, interpolación bilineal e interpolación bicúbica (24).

El algoritmo del vecino más cercano es el método de interpolación más básico. Este método consiste en cambiar el valor de los píxeles por el valor más cercano posible de acuerdo con el vecino próximo a interpolar (23) (25).

El error de posición de este método es a lo sumo medio píxel. Este error es perceptible en objetos con fronteras rectas en las que aparece un efecto de salto después de la transformación, se crea la forma característica de “escalera” en líneas y bordes.

La técnica de interpolación Bilineal (23) (26), consiste en implementar un tipo de filtro media, el cual se aplica tanto para el eje x como el eje y. Esto se logra promediando los cuatro píxeles adyacentes al pixel $p(x, y)$ interpolado. Este tipo de interpolación es mucho mejor que el vecino más cercano, para cambiar el tamaño de una imagen, pero causa cierto suavizado indeseable.

La técnica de interpolación Bicúbica es similar a la Bilineal, en cuanto a que se realiza tanto en el eje x como en el eje y (27). A continuación se describe el procedimiento.

En la interpolación bicúbica (23) (26) , la interpolación de los píxeles, se hace mediante la ecuación:

$$F_{hr}(i', j') = \sum_{m=-1}^2 \sum_{n=-1}^2 F(i + m, j + n) R(x) \quad (1.1)$$

$$R(x) = \frac{1}{6} [P(x + 2)^3 - 4P(x + 1)^3 + 6Px^3 - 4P(x - 1)^3] \quad (1.2)$$

$$P = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (1.3)$$

Este método, preserva mejor los detalles que la interpolación Bilineal, aunque aún mantiene un efecto de escalera o dentado, pero en mucho menor grado (28). El algoritmo completo utilizado se describe en (27). Ver fig. (5).

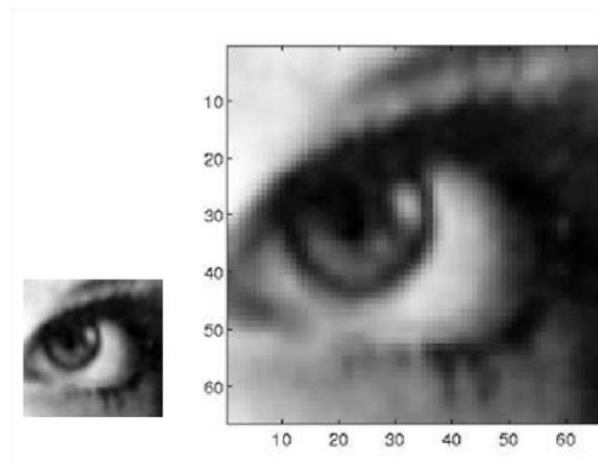


Figura 5: Interpolación Bicúbica. (Izquierda la imagen original y la derecha la resultante)

1.2.3 SR vía deformación de imagen (Warping).

Esta técnica, es propuesta por Chiang y Boulton en 1996 (29), basada en un integrador de remuestreo (re-sampling), la cual deforma la imagen, mientras se fuerza la imagen reconstruida, para esto se requiere de función de respuesta a impulso (PSF – Point Spread Function) de forma rectangular (30) (31).

El propósito fundamental del Warping es permitir que la imagen tome una nueva forma (reshaping) de la geometría de la imagen. Inherente a esta aplicación es la alineación o “matching” de los datos de las imágenes.

Este método crea manchas en zonas, en los bordes cuando estos son pequeños y están muy cercanos, además de suavizar la imagen.

1.2.4 SR basada en la Teoría de Conjuntos Convexos.

Los métodos de SR basada en la proyección sobre conjuntos convexos: Projection onto Convex Sets (POCS) describe un método iterativo alternativo incorporando conocimiento previo de la imagen sobre la solución en el proceso de reconstrucción (17).

Esta técnica consiste en formular el problema de SR mediante la definición de múltiples conjuntos convexos de restricción a los que debe pertenecer la imagen de AR x . La técnica POCS propone

un algoritmo iterativo para encontrar la imagen de AR dentro de la intersección de conjuntos convexos desde un valor inicial X_0 , de la forma:

$$X_{k+1} = P_M \dots P_1 X_k, \quad (1.4)$$

donde M representa el número de conjuntos convexos definidos y P_k para $k=1,2,\dots, M$ representan los operadores de proyección sobre los distintos conjuntos convexos. En cada iteración se modifica la estimación de la imagen AR, a partir del error de las observaciones generadas a partir de la imagen de AR estimada en la iteración anterior, y esto hasta que se cumplan todas las restricciones.

La gran ventaja de los métodos de SR según el enfoque POCS reside en la facilidad de su implementación, flexibilidad para utilizar modelos de observación generales y en la facilidad de incluir información a priori. Pero el coste computacional es alto porque la convergencia suele ser lenta. Por último, la solución no es única, y depende de la solución inicial escogida. (32) (33)

1.2.5 SR utilizando Máximo a Posteriori (MAP).

Como ya se conoce el problema de la superresolución en la mayoría de las situaciones está mal condicionado debido a que la información planteada en las imágenes de baja resolución no es suficiente para obtener una imagen de superresolución, entonces el problema mal condicionado debe ser estabilizado. El método MAP, el cual puede fácilmente incluir una imagen a priori o de regularización, es un marco eficiente para describir el problema de superresolución. (34)

La maximización de la distribución de probabilidad a posteriori es equivalente a este problema:

$$\hat{z} = \arg \min [\sum_k \|Y_k - A_k Z\|^2 + \lambda \Gamma(z)] \quad (1.5)$$

Donde el primer término es fidelidad de los datos, y $\Gamma(z)$ es el término de regularización.

Aquí, la optimización utiliza la dirección conjugada en el lugar del gradiente local para buscar el mínimo para una rápida convergencia. El gradiente es presentado a continuación, es denotado como:

$$r(\hat{z}^n) = \sum_k A_k^t (A_k \hat{z}^n - Y_k) + \lambda \nabla \Gamma(\hat{z}^n) \quad (1.6)$$

Donde el término de la derecha del gradiente $\nabla \Gamma(\hat{z}^n)$ es el término de la derivada de regularización con z y puede ser aproximado para la estimación de la imagen de SR (35).

1.2.6 SR en el dominio de la frecuencia.

SR en el dominio de la frecuencia se basa en tres premisas para lograr la reconstrucción de la imagen deseada: la propiedad de desplazamiento de la Transformada de Fourier, la relación de aliasing existente entre la Transformada Continua de Fourier (CFT) de la imagen de alta resolución y la Transformada Discreta de Fourier (DFT) de las imágenes de baja resolución y la asunción de que la imagen deseada se encuentra limitada en banda (20) (14) (36).

El problema de ir de BR a AR se puede resolver también en el dominio de la frecuencia. Aplicando el teorema de DFT tenemos:

$$F'(u, v) = \exp\{2j\pi(\delta_x^1 u + \delta_y^1 v)\}F(u, v) \quad (1.7)$$

donde F' es la imagen estimada, y F es la imagen de baja resolución, la imagen $f'(x,y)$ es muestreada ahora con periodos de tiempo T_x^1 y T_y^1 para generar la imagen de baja resolución de tamaño $N \times N$.

$$'F'_{IR}(i, k) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} f'_{ir}(m, n) \exp\left[-2j\frac{\pi}{M}(mi + nk)\right], i, k = 0, 1, \dots, N - 1 \quad (1.8)$$

Debido al teorema del muestreo, la Transformada de Fourier Continua y la Transformada de Fourier Discreta están relacionadas. (23)

$$'F'_{IR}(i, k) = \frac{1}{T_x^l T_y^l} \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} F' \left[\frac{2\pi}{T_x^l} \left(\frac{i}{N} + m \right), \frac{2\pi}{T_y^l} \left(\frac{k}{N} + n \right) \right] \quad (1.9)$$

Si la transformada de Fourier de la imagen original $F(u, v)$ está limitada en banda, dado que satisface $|F(u, v)| = 0$ para $|u| \geq \frac{\pi}{T_x}$ y $|v| \geq \frac{\pi}{T_y}$, la transformada de Fourier de las señales de BR tiene efecto de aliasing (17).

Cada imagen discreta de BR es la suma de $2L_x * 2L_y = 4L_x L_y$ muestras discretas de la imagen original $F(u, v)$ de acuerdo con la ecuación:

$$F'_{IR}(i, k) = \frac{1}{T_x^l T_y^l} \sum_{m=-L_x}^{L_x} \sum_{n=-L_y}^{L_y} \exp \left[j2\pi \left\{ \frac{\delta_x^i}{T_x^l} \left(\frac{i}{N} + m \right) + \frac{\delta_y^i}{T_y^l} \left(\frac{k}{N} + n \right) \right\} \right] * F \left[\frac{2\pi}{T_x^l} \left(\frac{i}{N} + m \right), \frac{2\pi}{T_y^l} \left(\frac{k}{N} + n \right) \right] \quad (1.10)$$

Y usando el orden lexicográfico una relación de orden total que se utiliza para ordenar caracteres de los índices m , n en el lado derecho y l en el izquierdo obtenemos la matriz.

$$F^{(l,k)} = \theta(\delta)^{(i,k)} * F_{IR}^{(i,k)} \quad (1.11)$$

Donde $F_{IR}^{(i,k)}$ es la imagen de baja resolución ordenada, y $F^{(i,k)}$ es la imagen de SR, y $\theta(\delta)^{(i,k)}$ es la función de estimación. De este análisis podemos decir que este método se puede resumir en la aplicación de una transformada de Fourier para $M \times N$ valores, y aplicar la inversa para $2M \times 2N$ valores. Ver fig. (6) donde se muestra la aplicación de la transformada de Fourier.

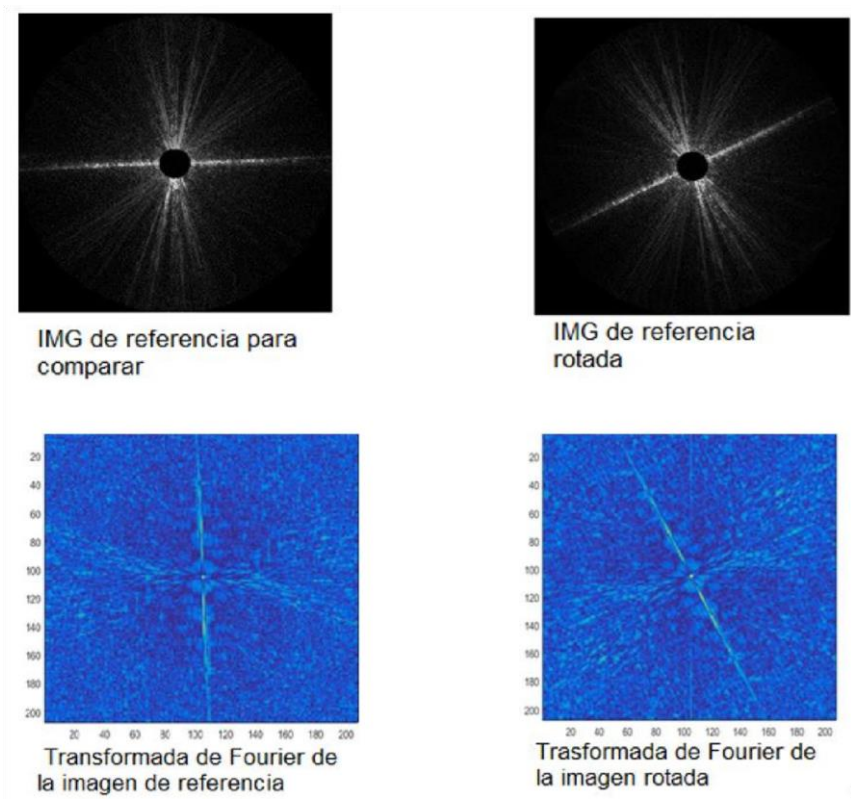


Figura 6: Resultados de la aplicación de la Transformada de Fourier

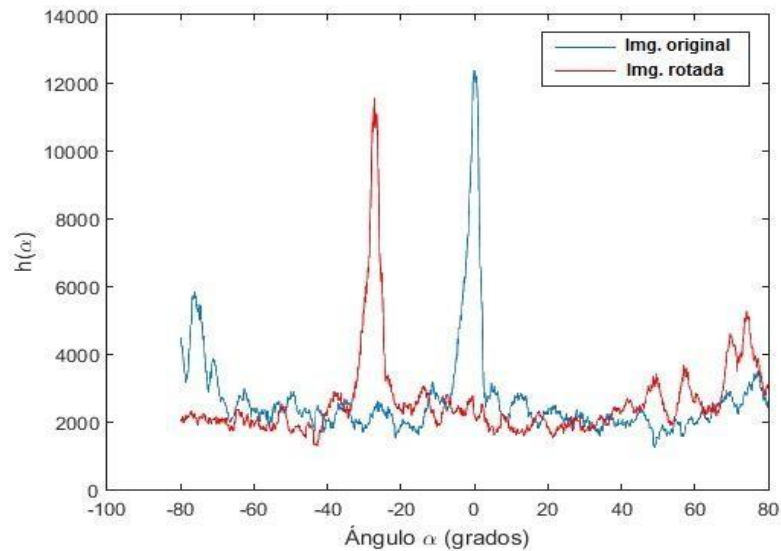


Figura 7 Proyecciones 1D de la imagen original y la de referencia.

La ventaja de la SR en el dominio de la frecuencia es que no presenta mucha complejidad teórica en comparación con otros métodos de SR, además que presenta bajos costes computacionales porque su resolución es altamente paralelizable. Teniendo como limitante el hecho de que solo se admita movimiento de traslación global entre imágenes.

1.2.6.1 SR utilizando la Transformada Coseno Discreta.

La transformada coseno discreta (DCT del inglés Discrete Cosine Transform) está bastante relacionada con la DFT, con la diferencia de que es una transformada real, debido a que los vectores base se comportan exclusivamente de funciones coseno muestreadas.

La DCT representa una imagen como una suma de sinusoides de distintas amplitudes y frecuencias. (23) (37) Ver fig. (8).

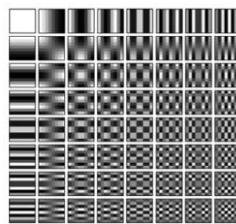


Figura 8 Matriz de 8x8 con los 64 valores base de la DCT

El concepto de la DCT se basa en tomar cada píxel de un bloque de 8 por 8. Ese píxel es una “Muestra” (sample) de una señal variable en el tiempo, proporcional a la luminancia y de otra señal variable en el tiempo, proporcional a la crominancia.

Estas dos señales son las que se pasarán, separadamente al dominio de las frecuencias, obteniéndose los coeficientes en frecuencia y transmitiéndolos, en lugar de transmitir las funciones de tiempo.

La DCT de una imagen se define de dimensiones MxN se define como:

$$B_{(p,q)} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A(m,n) \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad \begin{array}{l} 0 \leq p \leq M-1 \\ 0 \leq q \leq N-1 \end{array} \quad (1.12)$$

$$\alpha_p \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 0 \leq p \leq M-1 \end{cases} \quad (1.13)$$

$$\alpha_q \begin{cases} 1/\sqrt{N}, & q = 0 \\ \sqrt{2/N}, & 0 \leq q \leq N-1 \end{cases} \quad (1.14)$$

donde los valores de B son los coeficientes de la DCT de A.

Ya que hemos realizado la DCT ahora para decodificación utilizamos la transformada inversa (IDCT) que se utiliza para la obtención de un píxel aplicando la antitransformada DCT a un bloque de coeficientes. La IDCT no es más que un proceso de mezcla de distintas configuraciones de píxel donde las respectivas amplitudes y polaridades están dados por los coeficientes.

Análogamente, la transformada inversa se define como:

$$A_{(m,n)} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q B(p,q) \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad \begin{array}{l} 0 \leq p \leq M-1 \\ 0 \leq q \leq N-1 \end{array} \quad (1.15)$$

$$\alpha_p \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 0 \leq p \leq M - 1 \end{cases} \quad (1.16)$$

$$\alpha_q \begin{cases} 1/\sqrt{N}, & q = 0 \\ \sqrt{2/N}, & 0 \leq q \leq N - 1 \end{cases} \quad (1.17)$$

La técnica de la DCT es generalmente utilizada para compresión de imágenes, aunque se propone que también pueda ser utilizada para estimar los valores perdidos durante la adquisición de una imagen.

1.3 Métodos seleccionados.

Para darle solución a la problemática planteada una vez analizadas todas las técnicas y métodos existentes antes mencionados, se comprobó que en el vecino más cercano el error de posición es de medio píxel, este error es perceptible en objetos con fronteras rectas en las que aparece un efecto de salto después de la transformación. En la interpolación bilineal se produce una ligera disminución en la resolución a consecuencia del emborronado propio del promedio empleado. Mientras que, al utilizar la interpolación Bicúbica, no se sufre del problema del efecto de salto de color entre los píxeles y proporciona menor emborronamiento. Cuando usamos Warping se crean manchas en los bordes cuando estos están son pequeños y están muy cercanos, además de suavizar la imagen. En los métodos probabilísticas básicamente es necesario conocer información a priori y en muchas ocasiones no contamos con ella, además pueden llegar a crear efectos de fusionado de algunos objetos, además que en algunos casos necesitamos más de una imagen. Los resultados que proveen los métodos que usan transformadas son buenos, pero hay que realizar un ajuste de intensidad.

Con base de este análisis se escogió la SR basada en la frecuencia, ya que los resultados que proveen son buenos, especialmente prestándole mucha atención al método de la DFT ampliamente utilizado en la compresión de imágenes, pero en este trabajo de tesis será utilizado para aumentar la resolución espacial de una imagen. También se escogió aplicar distintamente los métodos de

interpolación expuestos, ya que el siguiente trabajo se basa en la obtención de una imagen de AR basados en una sola imagen BR.

1.4 Herramientas y metodologías utilizadas para el desarrollo.

Con el desarrollo de un sistema informático, la elección adecuada de los lenguajes y herramientas a utilizar es un proceso sumamente importante para la obtención de resultados satisfactorios. Un error provocado por una toma de decisión incorrecta en esta etapa, puede dilatar el proceso y generar datos adicionales no planificados.

1.4.1 Metodología de desarrollo de software.

Las metodologías de desarrollo de software constituyen el conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar un nuevo software. Representan un marco de trabajo que tiene entre sus principales funciones: guiar, planificar, estructurar, controlar, manipular y dirigir el proceso de desarrollo de sistemas de información. (38)

Las metodologías se pueden clasificar en tradicionales, que se basan en una fuerte planificación durante el desarrollo, y las metodologías ágiles, en las que el desarrollo de software es incremental, cooperativo, sencillo y adaptativo.

Luego de la evaluación de las distintas metodologías de desarrollo y el estudio de sus características, etapas de desarrollo y ventajas que posibilitan, se determina utilizar una metodología ágil, dado que la prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software. (39)

El equipo de desarrollo es de una sola persona, el cliente es parte del equipo de desarrollo, además de la dificultad que representa para un equipo de desarrollo pequeño el adoptar una metodología robusta a causa de la cantidad de documentación generada y la alta resistencia a los cambios durante el desarrollo. Es por ello que se identifica la Programación Extrema (XP) (40) como una alternativa acertada, además de ser la metodología de desarrollo del grupo de investigación AIRI. Una de las prácticas más significativas que posee esta metodología es que con en ella es posible simplificar el diseño para agilizar el desarrollo, así como facilitar el mantenimiento del sistema. Es

también recomendable utilizar esta metodología debido a que los requisitos tienen alta probabilidad de cambiar con el tiempo. (38)

Ventajas de XP (40)

- Comienza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- El cliente o el usuario se convierte en parte del equipo.

XP consta de 4 fases (40) (41)

1. **Planificación:** es la fase donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada iteración.
2. **Diseño:** la metodología XP hace énfasis en los diseños simples y claros. Por ello XP propone implementar el diseño más simple posible que funcione. Para ello XP propone utilizar soluciones Spike, recodificación, metáforas y tarjetas CRC.
3. **Codificación:** en esta fase se desarrolla en función de cada historia de usuario, además de ser la fase donde se definen las tareas de la ingeniería y los tiempos reales en que se realizaron cada una de las funcionalidades especificadas. La implementación debe realizarse de acuerdo a los estándares de codificación y debe llevarse un ritmo sostenido de trabajo.
4. **Pruebas:** estas pruebas se realizan al final del ciclo en el que se desarrollan, para verificar que las iteraciones no han afectado a las anteriores. Las pruebas que se llevan a cabo son pruebas unitarias, detección y corrección de errores, y pruebas de aceptación. (41)

1.5 Lenguajes utilizados.

Un lenguaje de programación es un conjunto de instrucciones que permiten la creación de procesos que pueden ser llevados a cabo por máquinas como las computadoras. (42) Para la realización del sistema se utilizaron los siguientes lenguajes.

1.5.1 Lenguaje de programación Java.

Java es un lenguaje de programación creado por Sun Microsystems Inc en un proceso por etapas que arranca en 1990 basado en la programación orientada a objetos. El nombre “Java” surgió en una de las secciones de “brainstorming” celebradas por el equipo de desarrollo del lenguaje, no es un acrónimo, sino únicamente algo humeante, caliente y que a muchos programadores les gusta beber en grandes cantidades: una taza de café. Una de las características más importantes de Java es que los programas ejecutables, creados por el compilador de Java, son independientes de la arquitectura. Se ejecutan indistintamente en una gran variedad de equipos con diferentes microprocesadores y sistemas operativos. Java aprovecha características de la mayoría de los lenguajes modernos evitando sus inconvenientes en particular los de C++. Tiene una gran funcionalidad gracias a sus librerías (clases). El manejo de la memoria no es un problema, la gestiona el propio lenguaje y no el programador. Incorpora Multi-Threading para permitir la ejecución de tareas concurrentes dentro de un mismo programa. El lenguaje Java es robusto. Las aplicaciones creadas en este lenguaje son susceptibles de contener pocos errores. Existen varios IDE de desarrollo que lo soportan como: NetBeans, Eclipse, IntelliJ IDEA y Jcreator, entre otros.

(43)

Debido a las necesidades existentes, se decide seleccionar un lenguaje de programación rápido, eficiente y robusto con el cual el equipo de desarrollo este familiarizado y que brinde las condiciones óptimas para la construcción exitosa del software. También debido a que el software PANDOC está desarrollado en el lenguaje de programación Java, se ha llegado a la conclusión de que el lenguaje a utilizar es Java para desarrollar la solución propuesta.

Características fundamentales de Java: (43)

- Simple: elimina la complejidad de otros lenguajes como C y se enfoca en el contexto de los lenguajes orientados a objetos.
- Robusto: maneja la memoria de la computadora para que el programador no se tenga que preocupar por ello, además de realizar verificaciones en busca de errores lo mismo en tiempo de compilación que en tiempo de ejecución.

- Portable: un programa compilado en Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java, ya que su código compilado es interpretado.
- Multiproceso: puede ejecutar diferentes líneas de código al mismo tiempo.

1.5.2 Entorno Integrado de desarrollo.

Para el desarrollo de la solución, como Entorno de Desarrollo Integrado (IDE según sus siglas en inglés) se seleccionó el NetBeans 8.1. Es libre, de código abierto y contiene todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java. Es mundialmente conocido por su integración con el lenguaje Java, facilita el desarrollo utilizando funcionalidades como completamiento de código, permite la utilización y edición de los componentes visuales de una forma sencilla. Es un entorno de desarrollo, disponible para varios sistemas operativos como Windows, Linux, Mac y Solaris. Además incluye el control de versiones, lo cual representa una ventaja debido a que permite administrar las diferentes versiones del código fuente (44).

1.5.3 Herramienta para el modelado.

Como herramienta para el modelado se seleccionó Visual Paradigm for UML (), una de las líderes del mercado de las llamadas herramientas de ingeniería de software asistida por computadora (CASE, por sus siglas en inglés) (45). Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Soporta multitud de estándares de modelado como UML, SysML, ERD, DFD, BPMN. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Es multiplataforma y cuenta con una versión libre para la comunidad. Soporta la revisión ortográfica, brindando sugerencias para varios idiomas como: español, inglés y alemán. (46)

1.6 Conclusiones del capítulo.

En el presente capítulo se realizó un estudio de varios algoritmos de superresolución el cual permitió conocer que están básicamente formados por métodos de estimación o interpolación de

píxeles. Después de realizada una clasificación básica de ellos este análisis permitió entender que la complejidad o sencillez de estos métodos no inciden directamente en los resultados. Después del estudio de estos algoritmos se expusieron algunas deficiencias encontradas lo que llevó como resultado a la selección del algoritmo de superresolución basado en la frecuencia utilizando la DFT para la estimación de valores de los píxeles. Además se identificaron y describieron las tecnologías y herramientas empleadas para desarrollar la propuesta planteada, destacándose la metodología XP, el lenguaje de programación Java y el Visual Paradigm como herramienta Case.

Capítulo 2: Propuesta de Solución.

En este capítulo se presenta la propuesta de solución para la implementación del algoritmo de superresolución a través de la DCT. Se describe la fase inicial de la metodología de desarrollo utilizada para implementar la solución propuesta, se presentan los artefactos generados propios de la metodología de desarrollo utilizada que se propone tales como: historia de usuarios y tarjetas CRC. Se propone un estándar de codificación y el patrón de diseño empleado. Además se realiza una descripción detallada de la solución propuesta con las herramientas y tecnologías a utilizar.

2.1 Descripción de la solución.

Para establecer la propuesta de solución se seleccionó llevar a cabo la SR en el dominio de la frecuencia a través de la DCT, el cual se menciona en las técnicas de SR expuestas anteriormente, con el objetivo de mejorar la resolución espacial de las imágenes en retroiluminación provenientes de la lámpara de hendidura. Permitiéndole al especialista una visión más amplia y con más calidad para los pacientes operados de cataratas. A continuación se muestra el diagrama de actividades Fig. 9 que modela la secuencia de pasos del algoritmo.

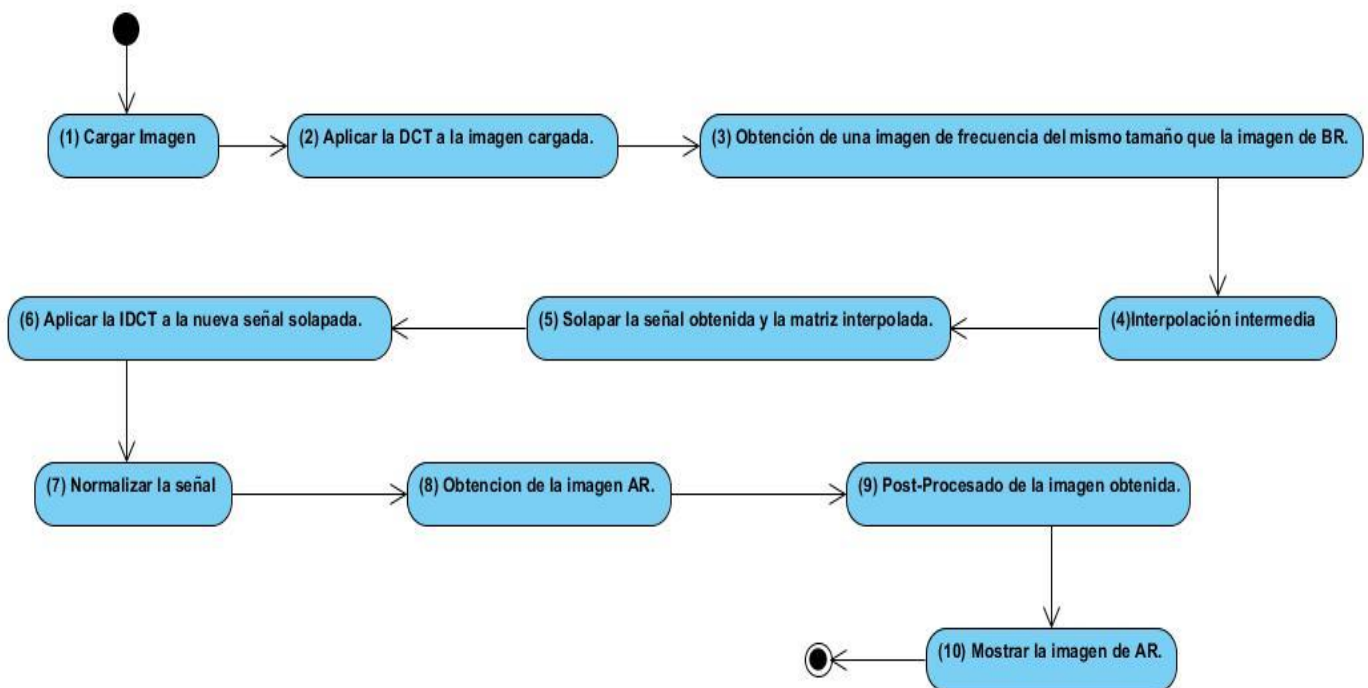


Figura 9 Descripción del algoritmo de SR propuesto.

Paso 1. Cargar imagen.

En este paso el usuario selecciona la imagen de BR que desea procesar y convertir en una imagen de AR.

Paso 2. Aplicar DCT a la imagen cargada.

Para aplicar la DCT se divide la secuencia muestreada en grupos de ocho muestras y se calcula la DCT independientemente para cada grupo. Debido a que las funciones base cosinusoidales siempre tienen el mismo conjunto de valores en cada uno de los puntos de muestreo discreto, solamente cambian los valores de los coeficientes. Las señales variables en el tiempo proporcional a la luminancia y la crominancia son las que se pasaran, separadamente al dominio de la frecuencia.

Paso 3 Obtención de una imagen de frecuencia del mismo tamaño que la imagen de BR.

Se obtiene la imagen de señales que devuelve el software al aplicar la DCT, mostrándose en un JLabel para su visualización y posterior procesamiento.

Paso 4 Interpolación intermedia.

En este paso se interpolan los valores de la señal obtenida en el paso anterior a través de una interpolación Bicúbica obteniéndose una señal del doble del tamaño que la original.

Paso 5. Solapar la señal obtenida y la matriz creada.

En esta etapa se posiciona la señal en la matriz de ceros creándose una imagen del doble de tamaño pero con bordes negros.

Paso 6 Aplicar la IDCT a la nueva señal solapada.

En este paso se le aplica a la nueva señal solapada la IDCT en bloques de 8x8. Esto lleva la imagen del dominio de la frecuencia a la representación en el dominio espacial pero con el doble del tamaño original del que fue cargada.

Paso 7. Normalizar la señal.

Se le aplica un filtro de paso alto para atenuar las frecuencias bajas manteniendo invariable las frecuencias altas. Esto mejora la detección de bordes y refuerza los contrastes que existen en la imagen.

Paso 8 Imagen de AR.

En este paso se obtiene una imagen de AR con mayor valor espacial que la imagen de BR cargada por el cliente.

Paso 9 Post-procesado de la imagen de AR.

En este paso se le aplica a la imagen un filtrado de realce para eliminar el ruido y la borrosidad que pudo provocar la estimación de píxeles llevado a cabo por el algoritmo.

Paso 10 Mostrar resultados.

Por último se muestra la imagen de AR reconstruida y se guarda en la dirección especificada por el usuario, y en caso de que no especifique ninguna se guarda en el mismo directorio de donde se cargó la imagen en BR.

2.2 Fase de planeación.

La fase de planeación es la etapa inicial del desarrollo de software propuesta en la metodología XP. En este punto se comienza a interactuar con el cliente para identificar cuáles son las historias de usuarios (HU). Se definen los ajustes necesarios a la metodología según las características del proyecto y el cliente definiéndose el nivel de prioridad de las HU, como el tiempo, el esfuerzo que conllevaran su desarrollo y el encargado del mismo. El resultado de la presente fase es un plan de entregas donde se realiza una estimación de las versiones que tendrá el producto en su elaboración, de forma tal que sea una guía durante el desarrollo. (47)

2.2.1 Especificación de los requisitos.

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema de software es llamado Ingeniería de Requerimientos. El objetivo es entregar una especificación de requerimientos de software correcta y completa. Esto mejora la forma en que se comprenden y definen sistemas de software complejos, trata los principios, métodos, técnicas y herramientas que permiten descubrir, documentar y mantener los requisitos para sistemas basados en computadoras en forma sistemática y repetible. (48)

Requisitos funcionales (RF): son declaraciones de las funcionalidades que debe proporcionar el sistema. Los requisitos funcionales establecen los comportamientos del sistema definiendo como debe comportarse el sistema en situaciones particulares (48). Los requisitos funcionales de la aplicación son mencionados a continuación:

- RF.1** Cargar imagen de BR.
- RF.2** Extraer los datos de la imagen.
- RF.3** Aplicar la DCT a la imagen.
- RF.4** Llevar la luminancia al dominio de la frecuencia.
- RF.5** Llevar la crominancia al dominio de la frecuencia
- RF.6** Aplicar la IDCT.
- RF.7** Llevar del dominio de la frecuencia a formato RGB los valores de la luminancia.
- RF.8.** Llevar del dominio de la frecuencia a formato RGB los valores de la crominancia.
- RF.9** Filtrado de la imagen.
- RF.10** Post-procesado de la imagen.
- RF.11** Mostrar imagen de AR.

2.2.2 Descripción de historias de usuario.

Uno de los artefactos definidos por la metodología XP son las HU las cuales son utilizadas para especificar las funcionalidades que brindará el sistema. Una HU es una representación de un requisito de software escrito en una o dos frases al utilizar el lenguaje común del usuario. Representan una forma rápida de administrar los requerimientos de los usuarios sin tener que elaborar gran cantidad de documentos formales, permitiendo responder rápidamente a los requisitos cambiantes. (47) Como parte de la presente investigación, el empleo de este artefacto fue clave, permitiendo de una manera simple describir una tarea concisa que en correspondencia con la prioridad por el cliente a cada HU y teniendo en cuenta la complejidad y riesgo determinado por el programador, se realiza la estimación de cada una de las identificadas.

A continuación se muestran la primera HU identificadas en esta fase:

Tabla 1. Historia de Usuario#1.

Historia de Usuario

Número: 1	Nombre de la Historia de Usuario: Cargar imagen de BR
Modificación a la Historia de Usuario: 0	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1 semanas
Riesgo en desarrollo: medio	Puntos reales: 1 semanas
Programador responsable: Robinson Machado Díaz	
Descripción: la HU comienza cuando el usuario carga la imagen de baja resolución, esta operación podrá realizarla al hacer <i>clic</i> a través de su selección en el menú de la aplicación.	
Observaciones: las imagen debe ser de BR.	

2.2.3 Fase de planificación de la entrega.

En la fase de planificación de la entrega se definen las prioridades de cada HU y consecuentemente se realiza una estimación del esfuerzo necesario de cada una de ellas. Se definen las entregas con el cliente teniendo como norma que una entrega debe adecuarse a tres meses máximo. Para el desarrollo satisfactorio de la solución propuesta, se realizó una estimación del esfuerzo para cada una de las HU.

En la tabla 2 se muestran cada una de las HU, así como la estimación del tiempo en que se cumplirá.

Tabla 2: Estimación de tiempo por historias de usuario.

No	Historias de usuario	Puntos de estimación (semanas)
1.	Cargar imágenes de BR.	1

2.	Extraer los datos de la imagen.	1
3.	Aplicar la DCT a la imagen	5
4.	Llevar la luminancia al dominio de la frecuencia	1
5.	Llevar la crominancia al dominio de la frecuencia	2
6.	Aplicar la IDCT.	5
7.	Llevar del dominio de la frecuencia a formato RGB los valores de la luminancia.	1
8.	Llevar del dominio de la frecuencia a formato RGB los valores de la crominancia	1
9.	Filtrado de la imagen.	2
10.	Post-procesado de la imagen.	1
11.	Mostrar imagen de AR.	1

2.2.4 Plan de iteraciones.

Una vez definidas las historias de usuario y estimar el tiempo y el esfuerzo para el desarrollo de cada una de ellas, se diseña un plan de iteraciones donde las historias de usuario están contenidas. Es decir, una iteración tiene como finalidad implementar las HU que se consideraron más necesarias atendiendo a su relevancia e impacto. El plan se propone, regulando el rango establecido para cada una de ellas, por tanto se pretende realizar el desarrollo en 3 iteraciones, referidas a continuación.

Iteración 1

La iteración tiene como fin aplicar la DCT, donde se carga la imagen de BR. Posteriormente se aplica un filtrado para llevar a la frecuencia la luminancia y la crominancia de los píxeles, y finalmente se aplica la DCT obteniéndose una imagen de frecuencias.

Iteración 2

Esta iteración se encarga de obtener la señal que produce la DCT y crear una matriz de ceros del doble del tamaño de esta señal, para luego posicionar la señal en la matriz de ceros y aplicar la IDCT a la nueva señal solapada.

Iteración 3

En esta iteración se realizan los filtrados de la imagen y el post-procesamiento de la imagen de AR para que luego sea visualizada y conservada por el cliente.

Tabla 3: Plan de duración de las iteraciones.

Iteraciones	Historias de usuario	Duración
Iteración 1	Cargar imagen de BR.	10
	Extraer los datos de la imagen.	
	Aplicar la DCT a la imagen.	
	Llevar la luminancia al dominio de la frecuencia.	
	Llevar la crominancia al dominio de la frecuencia.	
Iteración 2	Aplicar la IDCT.	7
	Llevar del dominio de la frecuencia a formato RGB los valores de la luminancia	
	Llevar del dominio de la frecuencia a formato RGB los valores de la crominancia	
	Filtrado de la imagen.	

Iteración 3	Post-procesado de la imagen.	4
	Mostrar imagen de AR.	

2.2.5 Plan de entrega.

El plan de entrega es el cronograma que define cuáles HU serán agrupadas para conformar una entrega y el orden en que se realizarán estas. Este plan se encarga de definir el número de liberaciones que se realizarán en el transcurso del proyecto y las iteraciones que se requieren para desarrollar cada una de ellas.

Tabla 4: Plan de duración de las iteraciones.

	Iteración 1	Iteración 2	Iteración 3
Cantidad de HU	5	4	2
Fecha de entrega	16/2/2018	1/5/2018	1/6/2018

2.3 Diseño de la solución.

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que hacerlo todo lo menos complicado posible para conseguir un diseño fácil de entender y de implementar, que a la larga reducirá el costo de tiempo y esfuerzo para desarrollar.

2.3.1 Tarjetas CRC

La metodología XP en lugar de utilizar diagramas para desarrollar modelos representa las clases mediante tarjetas. Las tarjetas Clase, Responsabilidad y Colaboración (CRC) son utilizadas para representar las responsabilidades de las clases y sus interacciones.

Las tarjetas CRC representan una entidad del sistema, a la cual asignar responsabilidades y colaboraciones. El formato físico de las mismas facilita la interacción entre los participantes del proyecto, en sesiones en las que se aplica técnicas de grupos como tormenta de ideas o juego de roles, y se ejecutan escenarios a partir de la especificación de requisitos, historias de usuario. De

esta forma surgen las entidades del sistema junto con sus responsabilidades y colaboraciones.
(49)

Tabla 5: Tarjeta CRC de la clase NewJFrame

Clase: NewJFrame	
Responsabilidades	Colaboradores
Permite mostrar y editar la imagen proveniente del Pentacam o de la Lámpara de Hendidura.	LabellImage
Permite realizar el proceso de SR	ProcesamientoImagen
Permite cargar la imagen de un fichero externo y tomar todos sus atributos.	Image

Tabla 6: Tarjeta CRC de la clase ProcesamientoImagen

Clase: ProcesamientoImagen	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Controla los procesos que se realizan en el sistema, mostrando los resultados visuales de aplicar cada uno de ellos. • Permite realizar diferentes procesos que contribuyen a facilitar el trabajo del usuario en el sistema. 	<ul style="list-style-type: none"> • DCT • Filtro • Contraste • LabellImage

2.3.2 Estándares de codificación.

Un estándar de codificación completo comprende todos los aspectos de la generación de código, debiendo reflejar un estilo armonioso en su confección (como si solo un programador hubiese escrito todo el código). Al comenzar un proyecto de software, se debe establecer un estilo de codificación para asegurarse de que se trabaje de forma coordinada, permitiendo realizar el mantenimiento a un sistema de software o incorporar código fuente previo, al establecer como operar con la base del código existente (50).

- Los comentarios irán encima del método correspondiente.
- Se utiliza nombres significativos para las clases, métodos, propiedades y variables.
- Los nombres de las clases serán con mayúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán de igual forma.
- Se define una declaración por línea.
- Los identificadores de las variables se escribirán en minúscula y si el nombre es compuesto las siguientes palabras empezaran con mayúscula.
- Se evita asignar el mismo valor a varias variables en la misma sentencia, pues dificulta su lectura.

2.3.3 Patrones de diseño utilizados.

Los patrones de diseño constituyen una descripción de la interacción entre las clases, con el objetivo de resolver un problema de diseño general en un determinado contexto. El uso de patrones permite la estandarización de la forma en que se realiza el diseño de manera que el código pueda ser reutilizable, y se le pueda dar mantenimiento cuando se requiera. Algunos ejemplos en este proyecto de la utilización de patrones de diseño se muestran a continuación:

Experto: el uso de este patrón se evidencia en la clase Imagen. Esta es la encargada de cargar la imagen desde un fichero externo.

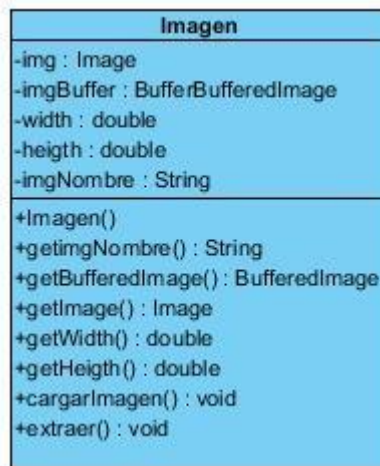


Figura 10: Clase Imagen

Creador: el uso de este patrón se evidencia en la clase ProcesamientoImagen

```
FastBitmap fb = new FastBitmap(imageActual);
Grayscale g = new Grayscale();
g.applyInPlace(fb);
CosineTransform Ct = new CosineTransform();
Ct.Forward(fb);
fb = Ct.toFastBitmap();
```

Figura 11: Creación de instancias.

2.4 Conclusiones del capítulo.

En este capítulo se abordaron los aspectos fundamentales del análisis y diseño de la propuesta de solución. El levantamiento de los requisitos del sistema permitió determinar las funcionalidades básicas a desarrollar durante el proceso. Se describió la propuesta de solución, enmarcada en el procesamiento digital de imágenes médicas según la metodología XP, generándose como artefactos un total de 11 HU, que describen las funcionalidades que se ejecutan. Se realizó también la estimación del esfuerzo necesario para la implementación de las UH, llevándose a cabo el plan de entregas en el cual se define cuál de ellas formara parte de cada entrega, determinándose como fecha de entrega final el 1/6/2018. Se lograron definir las actividades a implementar durante la fase de diseño mediante las tarjetas CRC. Se describió el funcionamiento del sistema, y cada una de

sus funcionalidades, así como el empleo de estándares de codificación y patrones de diseño, los cuales, permitieron realizar un diseño organizado y de fácil comprensión para otros desarrolladores.

CAPÍTULO 3: VALIDACIÓN Y RESULTADOS DEL SISTEMA.

En este capítulo se desarrollaran las fases de implementación y las pruebas al sistema propuestas por la metodología de desarrollo escogida. Las pruebas según la metodología XP se dividirán en dos grupos: pruebas unitarias, encargadas de verificar el código, la cual es diseñada por los programadores, y pruebas de aceptación o pruebas funcionales, destinadas a evaluar si al terminar una iteración se consiguió la funcionalidad requerida, estas últimas son diseñadas por el cliente final (51). Los resultados de la validación de un método se utilizan para juzgar la calidad, la fiabilidad y la constancia de los resultados, se trata de una parte integrante de cualquier buena práctica analítica.

3.1 Fase de implementación.

En esta fase se especifica la implementación de las HU en su correspondiente iteración, obteniéndose en cada una de ellas una versión funcional del producto. El primer paso es hacer un reconocimiento de cada HU a la par con el plan de iteraciones, pudiendo ser modificados si fuera necesario; una vez identificadas las HU, los programadores procederán a descomponer cada una de las tareas específicas, para esto se crearan las tareas de ingeniería, para así poder organizar la implementación de cada una de las HU. Estas tareas, a diferencia de las HU las cuales son escritas en el lenguaje del cliente, son realizadas por el programador en un lenguaje técnico. Como parte de la planificación elaborada en el capítulo anterior se definen a continuación las iteraciones del sistema.

Iteración 1

En esta iteración se implementaron las funcionalidades que permiten reconocer las imágenes provenientes de la lámpara de hendidura y realizar el proceso de DCT.

Tabla 7: Tiempo de implementación de las HU de la iteración #1

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Cargar imagen de BR.	1	1

Extraer los datos de la imagen.	1	1
Aplicar la DCT a la imagen	5	5
Llevar la luminancia al dominio de la frecuencia.	1	1
Llevar la crominancia al dominio de la frecuencia.	2	2

A continuación, se muestran las tareas de ingeniería efectuadas para las funcionalidades implementadas en esta iteración.

Tabla 8: Tarea de ingeniería de la funcionalidad Cargar imagen de BR.

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre: Cargar imagen de BR.	
Tipo de tarea: desarrollo	Puntos de estimación: 2
Programador responsable: Robinson Machado Díaz	
Descripción: Carga de un fichero seleccionado por el usuario la imagen de BR.	

Las demás tareas de ingeniería generadas por la iteración 1 se encuentran en el apartado “Anexo #2”.

Iteración 2

Tabla 9: Tiempo de implementación de las HU de la iteración #2

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Aplicar la IDCT.	5	5

Llevar del dominio de la frecuencia a formato RGB los valores de la luminancia	1	1
Llevar del dominio de la frecuencia a formato RGB los valores de la crominancia	1	1
Filtrado de la imagen.	2	2

Las tareas de ingeniería generadas por la iteración 2 se encuentran en el apartado “Anexo # 2”.

Iteración 3

Tabla 10: Tiempo de implementación de las HU de la iteración #3

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Post-procesado de la imagen.	1	1
Mostrar imagen de AR.	1	1

Las tareas de ingeniería generadas por la iteración 3 se encuentran en el apartado “Anexo # 2”.

3.2 Interfaz del sistema.

A continuación, se muestra la interfaz del sistema que permite visualizar los resultados del algoritmo diseñado.

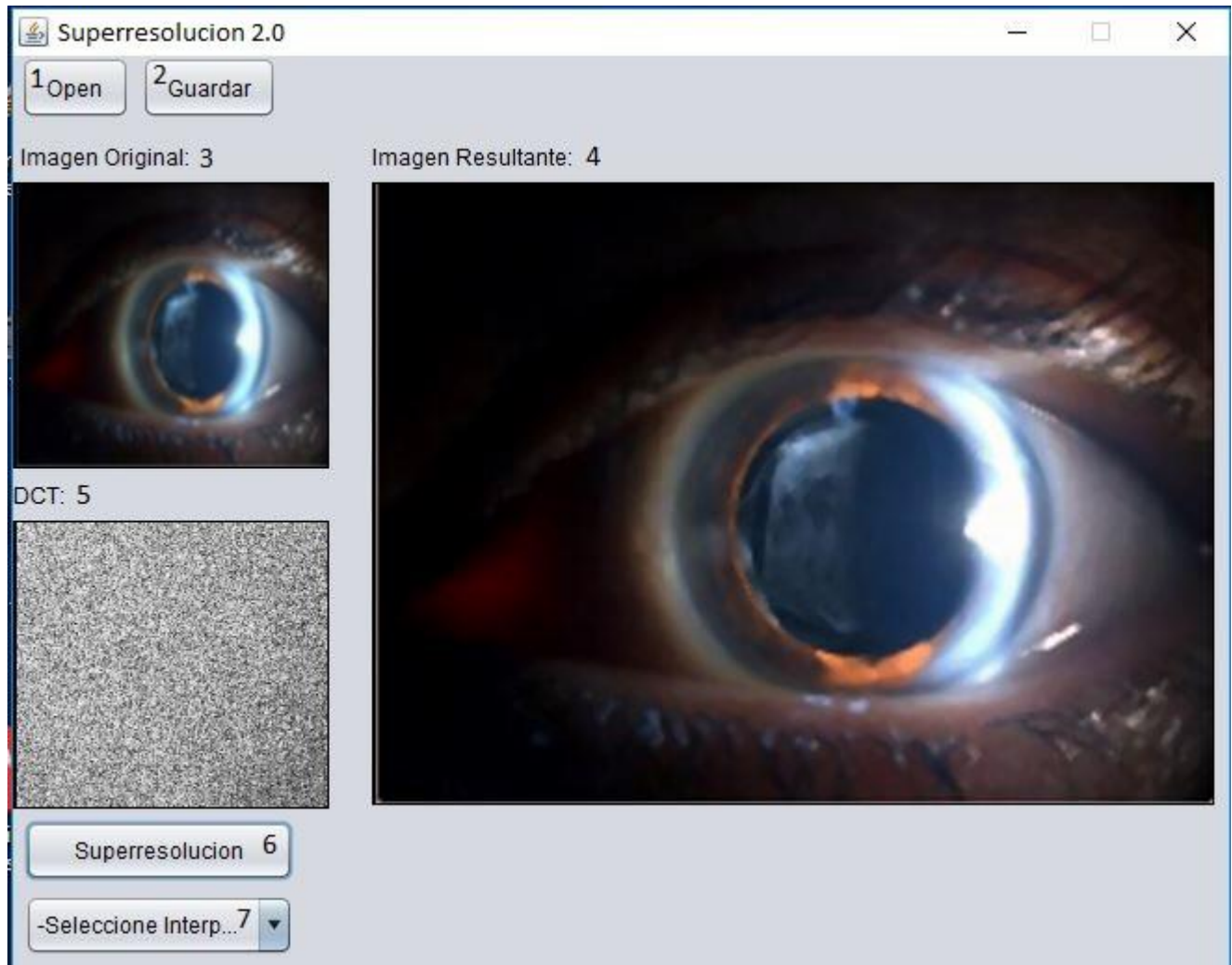


Figura 12: Interfaz principal del sistema.

- 1: Permite cargar la imagen de BR que se desea procesar.
- 2: Permite guardar en formato JPG la imagen procesada.
- 3: Muestra la imagen de BR.
- 4: Muestra la imagen de AR terminado el algoritmo.
- 5: Muestra el mapa DCT de la imagen de BR en el dominio frecuencial.
- 6: Permite aplicar el algoritmo de superresolución.
- 7: Permite aplicar interpolaciones Bicúbica, bilineal y vecino más cercano a la imagen de AR obtenida.

3.2.1 Resultados al aplicar el algoritmo.

El algoritmo diseñado se le aplicó a una muestra de 5 imágenes de BR provenientes de la lámpara de hendidura. A continuación se muestra un ejemplo luego de aplicar el algoritmo propuesto, en la Fig.13 derecha (resultado) e izquierda (original) respectivamente.

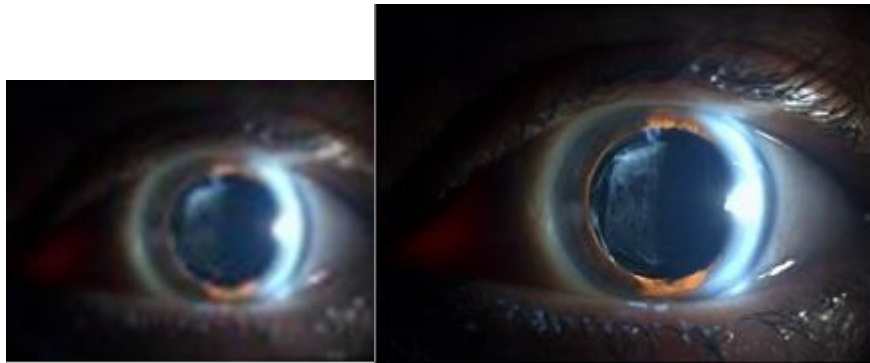


Figura 13: Resultado de aplicar el algoritmo.

3.3 Validación de los resultados.

El procesamiento digital de imágenes se basa en técnicas para mejorar la calidad de la imagen, partiendo de una muestra y obteniendo una mejorada. En el presente trabajo se utilizan los detectores de bordes para determinar la calidad de una imagen como indicador de comparación. Un borde es la frontera entre un objeto y el fondo. Identificando este borde, se puede localizar todo el objeto, así como medir sus propiedades básicas (52). Su detección resulta útil en diversas disciplinas, como la medicina, cuando se pretende preservar importantes propiedades estructurales o cuando se plantea la necesidad de distinguir formas o reconocer figuras dentro de una imagen (53). Uno de los métodos más importantes para realizar una detección global de bordes sobre una imagen es el conocido como método de Canny (54). Esta técnica, que se caracteriza por estar optimizada para la detección de bordes diferenciales. Consta de tres pasos:

- 1. Cálculo del gradiente:** lo primero que se realiza es la aplicación de un filtro gaussiano a la imagen original con el objetivo de suavizar la imagen y tratar de eliminar el posible ruido existente. Para la obtención del gradiente se aplica el filtro Sobel a la imagen suavizada. Se debe tener cuidado de no realizar un suavizado excesivo, pues se pondrían perder detalles de la imagen y provocar un pésimo resultado. Una vez que se suaviza la imagen, para cada

pixel se obtiene la magnitud y módulo (orientación) del gradiente, obteniendo así dos imágenes.

- 2. Supresión de No-Máximos:** una vez calculada la dirección del gradiente en el paso anterior, se debe encontrar la dirección que mejor se aproxima a la norma al borde. En este caso se eliminan los puntos que no tienen un valor máximo de gradiente según la dirección perpendicular al contorno. Las dos imágenes generadas en el paso anterior sirven de entrada para generar una imagen con bordes adelgazados. El procedimiento es el siguiente: se consideran cuatro direcciones identificadas por las orientaciones de 0° , 45° , 90° y 135° con respecto al eje horizontal. Para cada pixel se encuentra la dirección que mejor se aproxime a la dirección del ángulo del gradiente.
- 3. Histéresis:** en este paso se definen los bordes definitivos, seleccionando solo aquellos píxeles cuyo gradiente se encuentra entre dos umbrales establecidos. Una vez se ha completado este proceso se obtiene como resultado una imagen binaria en la que cada píxel se define como píxel de borde o como píxel no perteneciente al borde.

Al disminuir la desviación típica de la función gaussiana de suavizado se reduce el efecto de suavizado, por lo que en la imagen de bordes se incrementará el número de aristas. Por el contrario, al aumentar dicho parámetro se obtendrán aquellos contornos que correspondan con límites más contrastados, si bien conlleva un aumento del error en su posición (53).

Operador Sobel

Este operador detecta los bordes horizontales y verticales por separado sobre una imagen en escala de grises. El mismo convierte las imágenes RGB a imágenes a escala de grises. Calcula el gradiente de la intensidad de una imagen en cada punto. Así, para cada punto, este operador da la magnitud del mayor cambio posible, la dirección de éste y el sentido desde oscuro a claro (59). El resultado muestra como abrupta o suavemente cambia una imagen en cada punto analizado y en consecuencia, cuán probable es que éste represente un borde en la imagen, como la orientación a la que tiende ese borde (56).

Se trata de una técnica basada en el gradiente para realizar la detección de bordes, de forma que las variaciones de intensidad prevalecen frente a las zonas de intensidad constante (55).

Las máscaras que realizan esta operación de detección de bordes, en función de cual sea la dirección en la que se aplica el gradiente, son G_x en la dirección de las x que corresponde a las columnas y G_y en la dirección de las y que corresponde a las filas:

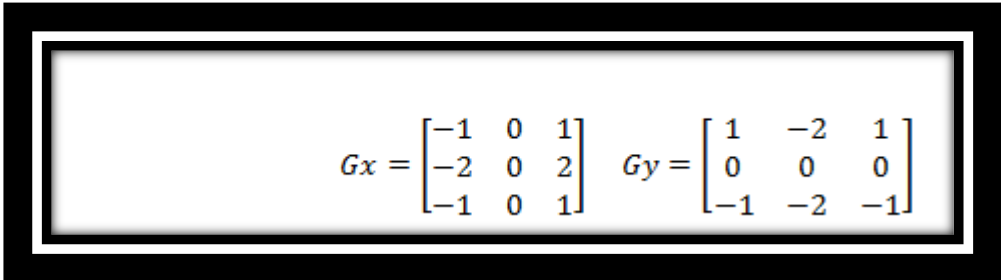

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Figura 14: Máscara binomial de 3x3

Para la validación mediante el detector de bordes Canny se tomó una muestra de 2 imágenes de BR y su versión restaurada mediante el algoritmo propuesto. Se procede a aplicar el detector de bordes Canny y contabilizar la cantidad de bordes en cada una de las imágenes (la de BR y la de AR).

A continuación, se ofrece una muestra de dos imágenes luego de aplicar Canny y cuantificar el número de bordes. La primera imagen es de baja resolución con una cantidad de bordes de 3457, y la segunda es el resultado de aplicar el algoritmo de superresolución con una cantidad de bordes de 11587. Este cuenta con una diferencia de 8130. Por tanto, teniendo en cuenta en que se estimó como un indicador de calidad la cantidad de bordes, al realizar la comparación estadística se observa que hay una mejoría, pues la cantidad en la imagen resultante (derecha) es mayor que la cantidad de la tomada como muestra (izquierda). Esto demuestra que se obtienen resultados satisfactorios.

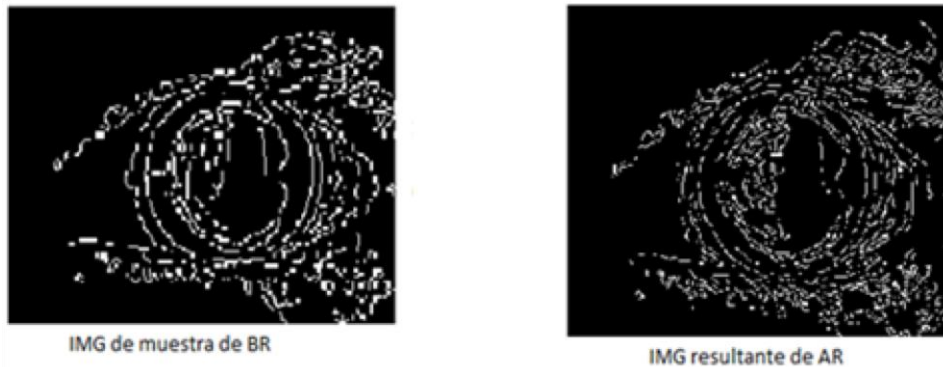


Figura 15: Resultados de la cuantificación de los bordes al aplicar Canny.

3.4 Verificación del sistema.

Los errores sencillos de un sistema que suelen estar ocultos, con el transcurso del desarrollo del mismo son más difíciles de detectar que los grandes errores que están a simple vista. Por esto es de suma importancia tener cuenta la aplicación de las pruebas de software de las distintas etapas de desarrollo del producto, pues las correctas aplicaciones de estas garantizan su calidad, provocando satisfacción al cliente y conformidad con lo realizado. Teniendo dominio sobre todos los procesos involucrados en el mismo.

3.4.1 Estrategia de pruebas.

Uno de los pilares de la metodología XP es el proceso de pruebas (57). Las pruebas de software son un elemento crítico para garantizar la calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación (58). XP exhorta constantemente a hacer pruebas tanto como sea posible, permitiendo aumentar la calidad de los sistemas, reduciendo el número de errores no detectados (51). La metodología XP divide las pruebas del software en dos grupos: pruebas unitarias, encargadas de verificar el código de manera automática y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales, destinadas a evaluar si al fin de una iteración se consiguió la funcionalidad requerida en las historias de usuarios diseñadas por el cliente final (59).

Objetivos de las pruebas.

Las pruebas presentan una interesante anomalía para el ingeniero del software. Durante las fases anteriores de definición y desarrollo, el ingeniero intenta construir el software partiendo de un concepto abstracto y llegando a una implementación tangible. El ingeniero crea una serie de casos de pruebas que intentan demoler el software construido. (60) Con el objetivo de:

- Detectar cualquier tipo de errores que presente el sistema.
- Identificar la calidad y eficiencia del software.
- Verificar que las funcionalidades del sistema estén en correspondencia con los requerimientos funcionales realizados por el cliente.

Una vez comenzada la construcción del sistema, se desplegó un conjunto de pruebas con el fin de comprobar su correcto funcionamiento y detectar la presencia o no de posibles errores que produjeran fallos en el producto.

3.4.2 Pruebas unitarias.

Una prueba unitaria es la verificación de una unidad de código determinado dentro de un sistema. Estas pruebas nos aseguran que un módulo determinado cumpla con un comportamiento esperado de forma aislada antes de ser integrado al sistema, es decir, debe aprobar satisfactoriamente todos los casos de pruebas definidos. El uso de estas pruebas evita tener que invertir una gran cantidad de horas de sesiones en “debugging” al momento de integrar el código con el sistema existente. En la metodología XP los programadores deben definir las pruebas unitarias para cada módulo antes de escribir el código, y solo para aquellas unidades de código donde exista la posibilidad que puedan fallar, es necesario escribir casos de prueba (61).

A continuación, se describirá la confección de las pruebas unitarias realizadas, desglosándolas en sus elementos fundamentales: confección del grafo de flujo, cálculo de la complejidad ciclomática, extracción de los caminos independientes, realización de los casos de prueba y análisis de los resultados obtenidos.

```

public BufferedImage escalaGrises() {
    //Variables que almacenarán los píxeles
    int mediaPixel, colorSRGB;//1
    Color colorAux;//1

    //Recorremos la imagen píxel a píxel
    for (int i = 0; i < imageActual.getWidth(); i++) {//2
        for (int j = 0; j < imageActual.getHeight(); j++) {//3
            //Almacenamos el color del píxel
            colorAux = new Color(this.imageActual.getRGB(i, j));//4
            //Calculamos la media de los tres canales (rojo, verde, azul)
            mediaPixel = (int) ((colorAux.getRed() + colorAux.getGreen() + colorAux.getBlue()) / 3);//4
            //Cambiamos a formato sRGB
            colorSRGB = (mediaPixel << 16) | (mediaPixel << 8) | mediaPixel;//5
            //Asignamos el nuevo valor al BufferedImage
            imageActual.setRGB(i, j, colorSRGB);//5
        }//6
    }//7

    //Retornamos la imagen
    return imageActual;//8
}

```

Figura 16: Algoritmo para convertir de RGB a escala de grises de la imagen de entrada.

Partiendo del siguiente fragmento del código tomado se obtuvo el siguiente grafo de flujo:

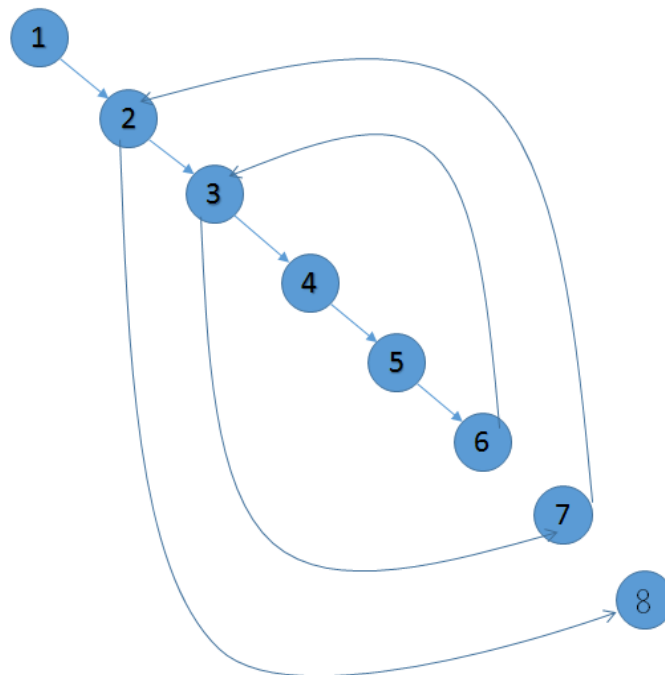


Figura 17: Diagrama de flujo del algoritmo “escalaGrises”.

Tras definir el grafo de flujo se procede al cálculo de la complejidad ciclomática siendo esta una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa (62). Para esta operación existen tres vías de solución, las cuales se enuncian a continuación:

- $V(G)=(A-N)+2$
- $V(G)=P+1$
- $V(G)=R$

La complejidad ciclomática, $V(G)$, de un grafo de flujo G , se define como:

$V(G) = (A-N)+2$, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.

La complejidad ciclomática, $V(G)$, de un grafo de flujo G , también se define como:

$V(G) = P+1$, donde P es el número de nodos predicados contenidos en el grafo de flujo G .

$V(G) = R$, donde R es la cantidad total de regiones.

Del grafo del flujo G se obtiene que:

$$V(G) = (A-N) + 2 = (9-8) + 2 = 3$$

$$V(G) = P+1 = 2+1 = 3$$

$$V(G) = R = 3$$

Luego de aplicar las tres formas para afirmar un resultado seguro y confiable, se obtiene una complejidad ciclomática $V(G)=3$ la cual representa la cantidad de caminos independientes para el grafo de flujo construido para el método “escalaGrises”, por lo que se definen los 3 caminos siguientes:

Tabla 11: Caminos básicos.

Número	Caminos básicos
1	1-2-3-4-5-6-3-7-2-8
2	1-2-8
3	1-2-3-7-2-8

Luego de tener elaborado el grafo de flujo e identificado los caminos a recorrer, se preparan los casos de prueba que forzarán la ejecución de cada uno de esos caminos. Con el fin de comprobar cada camino, se escogen los datos de manera que las condiciones de los nodos predicados estén adecuadamente establecidos. A continuación se especifican estos casos de pruebas.

Tabla 12: Caso de prueba del camino básico aplicado al método escalaGrisés.

Entrada	Imagen de BR.
Resultados esperados	La imagen en escala de grises.
Condiciones	La imagen debe estar previamente cargada.

Después de aplicar el mismo procedimiento a todas las funcionalidades de la aplicación no se detectó ninguna inconformidad en las mismas.

3.4.3 Pruebas de aceptación.

Cada prueba de aceptación o prueba funcional como también se le conoce está asociada a una historia de usuario, estas pruebas se realizan en esta etapa del proyecto describiéndose en ellas las posibles formas de utilización del software. Las pruebas funcionales no solo validan la transformación de una entrada en una salida, sino que validan una característica completa (48). En estos documentos de pruebas se indican las posibles respuestas que da el software tras la utilización de cada funcionalidad, así como los posibles mensajes de error, información o de aceptación que emite el software cuando se utiliza dicha funcionalidad (63).

Las pruebas de aceptación son creadas en base a las Historias de Usuario, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia ha sido correctamente implementada. Las pruebas de aceptación son de gran importancia, dado que miden el grado de satisfacción del cliente con el producto desarrollado (60). Por lo tanto, es responsabilidad de los clientes verificar que los datos de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, son ellos los encargados de indicar el orden de resolución de fallos. Una HU no se puede considerar terminada hasta tanto pase

correctamente todas las pruebas de aceptación. Es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información (60).

Para las pruebas de aceptación se diseñaron 7 casos de pruebas de los cuales para una primera iteración se obtuvieron 4 de forma satisfactoria y 3 de forma no satisfactoria. Para una segunda iteración se obtuvieron 6 casos de prueba de forma satisfactoria y 1 de forma no satisfactoria. Y para una tercera iteración se obtuvieron los 7 casos de prueba de forma satisfactoria. Ver Fig.19, y las no conformidades en cada una de estas iteraciones. Ver Tabla.20

Tabla 13: Caso de prueba de aceptación HU#1

Caso de prueba de aceptación
Código: HU_I1 Historia de Usuario:1
Nombre: Cargar imagen de BR
Descripción: registra la imagen de BR que el cliente desea procesar.
Condición de ejecución: La imagen debe estar en formato tif.
Entrada/ Pasos de ejecución <ul style="list-style-type: none"> El usuario selecciona la imagen. El sistema comprueba que este en formato jpg, bmp o gif. Muestra en un panel la imagen de BR cargada.
Resultado esperado: Se registra la imagen de BR y se muestra en un panel para ser posteriormente procesada.
Evaluación de la prueba: Satisfactoria.

Tabla 14: Caso de prueba de aceptación HU#2

Caso de prueba de aceptación
Código: HU_I2 Historia de Usuario:2
Nombre: Extraer los datos de la imagen.
Descripción: registra y almacena los datos de la imagen.
Condición de ejecución: La imagen debe ser previamente cargada.

<p>Entrada/ Pasos de ejecución</p> <ul style="list-style-type: none"> El sistema analiza la imagen cargada y almacena sus datos, como el nombre, el ancho, el largo y la matriz de pixeles.
<p>Resultado esperado:</p> <p>Se almacena los datos de la imagen para su posterior procesamiento.</p>
<p>Evaluación de la prueba: Satisfactoria.</p>

Tabla 15: Caso de prueba de aceptación HU#3

Caso de prueba de aceptación
Código: HU_I3 Historia de Usuario:3
Nombre: Aplicar la DCT
Descripción: aplica la DCT a la imagen de BR cargada por el cliente convirtiéndola del dominio espacial al de frecuencia.
Condición de ejecución:
Se debe tener los datos de la imagen almacenados.
<p>Entrada/ Pasos de ejecución</p> <ul style="list-style-type: none"> El sistema aplica la DCT en bloques de 8x8 a la imagen de BR cargada previamente convirtiéndola en una señal.
<p>Resultado esperado:</p> <p>Una señal del mismo tamaño que la imagen de BR cargada.</p>
<p>Evaluación de la prueba: Satisfactoria.</p>

Tabla 16: Caso de prueba de aceptación HU#6

Caso de prueba de aceptación
Código: HU_I6 Historia de Usuario:6
Nombre: Aplicar IDCT
Descripción: le aplica IDCT a la señal solapada.

Condición de ejecución: Se debe tener almacenada la señal solapada con el doble del tamaño de la imagen original.
Entrada/ Pasos de ejecución <ul style="list-style-type: none"> El sistema aplica la IDCT a la señal almacenada calculando los nuevos valores de pixeles para obtener la imagen de AR.
Resultado esperado: Se obtiene una imagen de AR y se almacena para su posterior procesado.
Evaluación de la prueba: Satisfactoria.

Tabla 17: Caso de prueba de aceptación HU#9

Caso de prueba de aceptación
Código: HU_I9 Historia de Usuario:9
Nombre: Filtrado de la imagen.
Descripción: se realiza para para atenuar las frecuencias bajas manteniendo invariable las frecuencias altas.
Condición de ejecución:
Entrada/ Pasos de ejecución <ul style="list-style-type: none"> El sistema aplica un filtro de paso alto a la imagen de AR obtenida al aplicar IDCT a la señal solapada.
Resultado esperado: Mejorar la detección de bordes y reforzar los contrastes existentes en la imagen de AR obtenida.
Evaluación de la prueba: Satisfactoria.

Tabla 18: Caso de prueba de aceptación HU#10

Caso de prueba de aceptación
Código: HU_I10 Historia de Usuario:10
Nombre: Post-procesamiento de la imagen.
Descripción: se le aplica a la imagen un filtrado de realce para eliminar el ruido y la borrosidad.
Condición de ejecución:

Se debe tener almacenada la imagen de AR obtenida.
<p>Entrada/ Pasos de ejecución</p> <ul style="list-style-type: none"> El sistema aplica un filtro a la imagen para realzar los contrastes de la imagen y así eliminar el ruido y la borrosidad que pudo provocar la estimación de pixeles llevado a cabo a la hora de aplicar el algoritmo.
<p>Resultado esperado:</p> <p>Una imagen de AR con mejor calidad espacial que la imagen de BR cargada por el cliente.</p>
Evaluación de la prueba: Satisfactoria.

Tabla 19: Caso de prueba de aceptación HU#11

Caso de prueba de aceptación
Código: HU_I11 Historia de Usuario:11
Nombre: Mostrar resultado
Descripción: Muestra la imagen de AR.
<p>Condición de ejecución:</p> <p>Debe haber finalizado el algoritmo de SR.</p>
<p>Entrada/ Pasos de ejecución</p> <ul style="list-style-type: none"> Se muestra la imagen de AR al cliente para que pueda visualizarla y guardarla si así lo desea.
<p>Resultado esperado:</p> <p>Mostrar en un JLabel la imagen de AR que se obtiene después de aplicar el algoritmo de SR.</p>
Evaluación de la prueba: Satisfactoria.

En la siguiente tabla se muestra cómo se comportan las no conformidades identificadas, su respectiva iteración y si se le dio solución a cada una de ellas.

Tabla 20: Resultados de las pruebas de aceptación.

No de iteración	No conformidades	Recomendaciones	No conformidades tratadas.
1	8	6	8
2	5	3	5

3	0	0	2
---	---	---	---

Las pruebas se realizaron de forma iterativa e incremental, y se comprobó en cada iteración la corrección de los errores detectados en la iteración anterior, lo que contribuyó a mejorar la calidad y funcionalidad del sistema, quedando aprobado por parte del cliente, quien evaluó el 100% de los casos de prueba de forma satisfactoria.

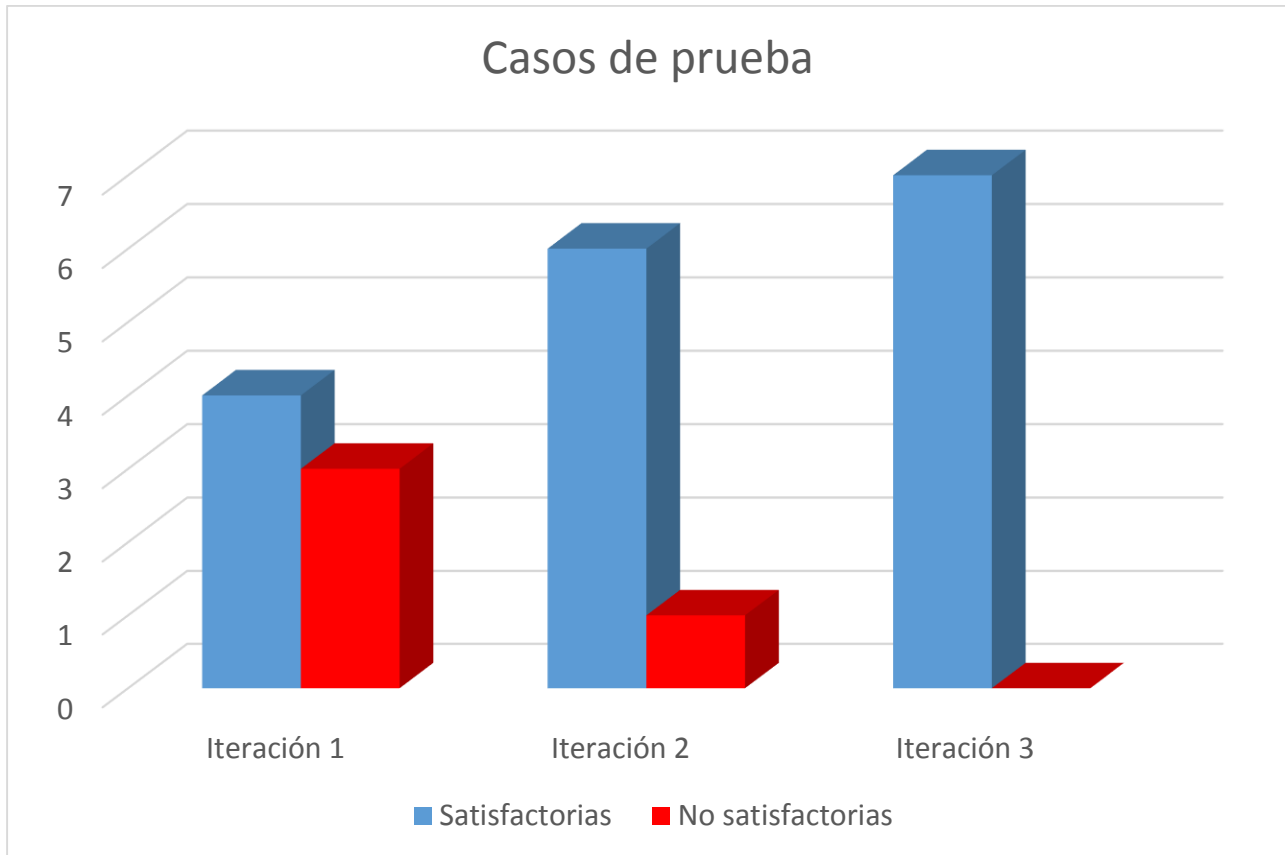


Figura 18: Resultados de las pruebas de aceptación

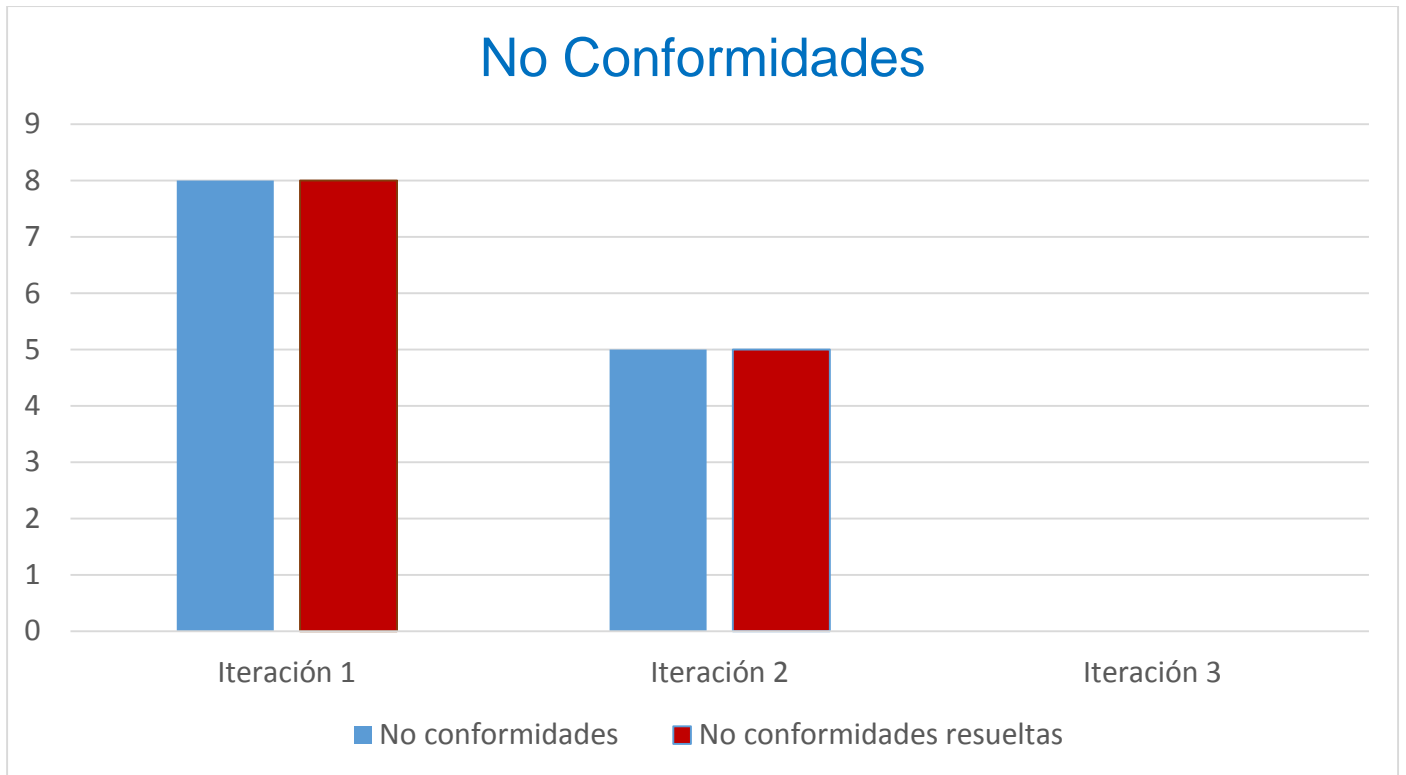


Figura 19 Tratamiento de las no conformidades.

3.5 Conclusiones del capítulo.

Durante el desarrollo del capítulo se realizaron pruebas para validar las funcionalidades del algoritmo propuesto, tales como las pruebas unitarias en las cuales no se detectaron errores y las pruebas de aceptación las cuales luego de 3 iteraciones se corrigieron las no conformidades detectadas, significando que las historias de usuario implementadas cumplen con los requisitos y necesidades del cliente. También se aplicó el detector de bordes Canny el cual evidenció como resultado que la imagen de AR obtenida tiene cerca de 2% más de píxeles blancos que la imagen de BR original, esto demostró que la imagen no solo aumentó su tamaño sino que además mejoró su calidad visual.

Conclusiones

Con el desarrollo del presente trabajo de diploma se cumplieron todas las tareas propuestas. De esta forma se arribó a las siguientes conclusiones:

- Luego de consultar diferentes literaturas se aprecia que la opacidad de la cápsula posterior o catarata secundaria representa la principal causa de disminución en la visión de los pacientes operados de catarata, por lo que se hace necesario prevenir a tiempo dicha patología.
- Se realizó la superresolución utilizando la Transformada Discreta del Coseno para mejorar la calidad de las imágenes provenientes de la lámpara de hendidura obteniéndose resultados satisfactorios.
- Se validaron los resultados mediante el detector de bordes Canny, para comprobar la calidad de la imagen obtenida. Para medir dicho indicador, la imagen resultante debe tener mayor número de bordes que la imagen de referencia. Se obtuvieron resultados satisfactorios mostrándose un mayor número de bordes en la imagen de alta resolución obtenida en comparación con la imagen de baja resolución.
- El algoritmo propuesto cumple con los requisitos especificados, ya que permite obtener una imagen de alta resolución, de mayores dimensiones que la original.
- Las pruebas de software permitieron verificar si el sistema cumple con las funcionalidades especificadas utilizando los métodos de caja negra y caja blanca, por lo que el algoritmo podrá ser usado en aras de mejorar la precisión al emitir un diagnóstico de la opacidad de la cápsula posterior.

RECOMENDACIONES

Para futuras investigaciones:

- Se recomienda la implementación y prueba de otros algoritmos de interpolación que puedan ser aplicados en el dominio de la frecuencia para aumentar el tamaño de la señal, sin que la misma pierda información o se distorsione. Así sería posible aumentar el tamaño de la imagen de AR más allá que solo el doble del tamaño que la imagen de BR original.
- Se recomienda estudiar las formas de validación de la interpolación en cuanto a la mejora de la calidad de las imágenes para la utilización del método que mejor resultado brinde a la hora de trabajar con imágenes en el dominio de la frecuencia.

REFERENCIAS BIBLIOGRÁFICAS

1. Boyd, K., *What Are Cataracts?*. American Academic of Ophthalmology, 2014.
2. Acosta R, H.L., Román R, Comas M, Castilla M, Castells X, Revisión sistemática de estudios.
3. Iván Hernández López, J.R.H.S., Yadira Castro González, Ailén Garcés Fernández, Zucell Veitía Rovirosa, Eneida Pérez Candelaria, *Estrategias de prevención de la opacidad de la cápsula posterior*. Revista Cubana de Oftalmología, 2010. **23**.
4. Kevin W. Bowyer, K.H., Patrick J. Flynn., *mage understandin for iris biometrics: A Survey*. Computer Vision and Image Understanding, 2008. **110**: p. 281-307. 1077-3142.
5. Michel Álvarez Cancio, E.P.V., Rafael Rodríguez Puentes, SilenaHerold García, *Algorithm for the identification of the posterior capsule opacity in images from PENTACAM*. Revista Cubana de Ciencias Informáticas 2017. **11**: p. 153-167.
6. Estrada, D.E.S., *Procesamiento digital de imagen para la Plataforma de Televisión Informativa*. TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.
7. ALVAREZ CANCIO, Michel; HERNÁNDEZ BARRIOS, Adrián; RODRÍGUEZ PUENTES, Rafael y HERNÁNDEZ LÓPEZ, Iván. PANDOC: software para la cuantificación objetiva de la opacidad de la cápsula posterior mediante tomogramas scheidtflug del pentacam. 2013.
8. BARMAN, Sarah A; HOLLICK, Emma J; BOYCE, James F et al., Quantification of posterior capsular opacification in digital images after cataract surgery. *Investigative ophthalmology & visual science*. 2000, vol. 41, n.o 12, págs. 3882-3892.
9. Raciél Cepero Ruz, C.C.R., Implementación de algoritmos de reducción de ruido en las imágenes, 2010.
10. Mcclellan Schafer Yoder, Signal Processing First.
11. FRIES, Bruce y FRIES, Marty. Audio digital práctico. Ed. Anaya Multimedia. 2005.
12. Demetrio Pita Salorio, *Diccionario terminológico de oftalmología*, Informe técnico ().
13. Jack J. Kanski, *Oftalmología clínica, 5ta edición*. Informe técnico, ISBN 978-84-8174-758-4.
14. Katsaggelos, A. K., Molina, R. y Mateos, J. *Super Resolution of Images and Video*. Morgan and Claypool, 2007.

15. Chaudhuri S., Super-Resolution Imagin, Kluwer Academic Publ. ISBN: 0-7923-7471-1. MA, USA.
16. Chaudhuri, S. y Manjunath, J. *Motion-free super-resolution*. Springer, 2005.
17. S. C. Park, M.K. Park y M.G. Kang, "Super-resolution image reconstruction:A technical overview", IEEE Signal Processing Magazine, vol. 20, no. 3, pgs.21-36, 2003.
18. C. A. Segall, R. Molina y A.K. Katsaggelos, "High-Reesolution images from low-resolution compressed video", IEEE Signal Processing Magazine, vol. 20, pgs. 37-48, 2003.
19. S. Chaudhuri, "Super-resolution Imaging", editor, Kluwer Academic Publishers, 2002.
20. R. Y. Tsai and T. S. Huang, "Multiframe image restoration and registration", in Advances in Computer Vision and Image Processing, R. Y. Tsai and T. S. Huang, Eds., vol. 1, pp. 317–339. JAI Press Inc., 1984.
21. Borman, S. y Stevenson, R. Spatial resolution enhancement of low-resolution image sequences. A comprehensive review with directions for future research. Informe técnico, Laboratory for Image and Signal Analysis (LISA), University of Notre Dame, Notre Dame, IN 46556, USA, 1998.
22. Milanfar, P. *Super-Resolution Imaging*. Digital Imaging and Computer Vision. Taylor&Francis/CRC Press (Ed.), 2010.
23. Chaudhuri, Subbasis, "Super-Resolution Imaging.Hingman", MA, Kluwer Academic Publishers, 2001, USA.
24. Flusser, B.Z.a.a.J., *Image registration methods: a survey*. Image and Vision Computing Journal, 2003: p. 977-1000.
25. T.M Lehmann, C. Gonner, K. Spitzer, "Survey: Interpolation methods in medical image processing", IEEE Trans. on Medical Imag.18, (11), 1999, p: 1049-1075.
26. H.S Hou and H. S Andrews, "Cubic splines for image interpolation and digital filtering", IEEE Trans. On Accoust. , Speech and Signal processing, vol.26, no. 6, 1978, p: 508-517.
27. Richards, J.A., Remote Sensing Digital Image Analysis. 1999.

28. M.Q. Phu, P.E. Tischer, H.R. Whu, "A median based interpolation algorithm for deinterlacing", Proc. of Int Symposium on Intelligent Signal Processing and Communication Systems, 2004, Seul South Korea, p: 390-397
29. Chiang y Boulton, The integrating resampler and efficient image warping, 1996.
30. E. Boulton, Ming-Chao Chiang and Ross J. Micehals SUPER-RESOLUTION WARPING VIA IMAGE Theory, Implementation and Evaluation Terrance, Image and Vision Computing, Elsevier House, July, 2000
31. Sanchez-Beato A. and Pajares G. Noniterative Interpolation-Based Super-Resolution Minimizing Aliasing in the Reconstructed Image. IEEE Trans. On Image Proc., Vol.17, No.10, 2008, p: 1817-1826, ISSN: 1057-7149.
32. Hafner, Liedlgruber, "POCS-based super-resolution for HD endoscopy video frames" in Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on, pp.185-190, 20-22 Junio 2013.
33. Xie Wei, Zhang Feiyan, Chen Hao, Qin Qianqing, "Blind Super-resolution Image Reconstruction Based on POCS Model", Measuring Technology and Mechatronics Automatic, 2009, ICMTMA '09. International Conference, vol 1, 12 Abril 2009.
34. Zhang Liangpei, Zhang Hongyan, Shen Huanfeng, Li Pingxiang. A super-resolution reconstruction algorithm for surveillance images. Signal Processing.
35. Francisco Gomeztagle Sepulveda, Algoritmos para superresolución en imágenes y video, 2010.
36. Katsaggelos, A.K y Molina, editors, Super resolution (special issue). The Computer Journal, 52, 395-396, 2009.
37. Bovik A, Handbook of Image and Video Process. Academic. ISBN: 0-12-119790-5.
38. Sánchez Mendoza, M.A, Metodologías de desarrollo de software., 2004.
39. Canos José H, Letelier Patricio, Panadés M Carmen. Metodologías Agiles en el Desarrollo de Software. Valencia, 2003.
40. Bustamante, D y Rodríguez, J. Metodología Actual: Metodología XP. 2014.
41. Joscowicz, J. Reglas y prácticas en eXtreme Programming., 2008.

42. Suarez, M.L, Cuaderno 1: Introducción a la programación y sus lenguajes. Competencias en TIC: Colección de Fascículos Digitales. 2015.
43. Oracle. Java [en línea] [citado el: 21-2 de 2018.] [https:// www.java.com/es/about/..](https://www.java.com/es/about/)
44. Netbeans.org. [En línea] [Citado el 3-3 de 2018.] [http:// netbeans.org](http://netbeans.org).
45. Visconti, M.a.A, Hernán, Fundamentos de la ingeniería de software.
46. Visual Paradigm. UML tool. Available from: [http:// www.visual-paradigm.com](http://www.visual-paradigm.com).
47. Kent, Beck, Jesús García Molina & Luis Joyanes Aguilar. Una explicación de la programación extrema: aceptar el cambio., Addison Wesley 2002.
48. Sommerville, I., Ingeniería. Del Software. 2005: España: Pearson Educación.
49. Casas, S., & Reinaga, H., Aspectos tempranos: un enfoque basado en Tarjetas CRC. 2009.
50. García, O. Nomenclatura estándar del código Java. [En línea] [http:// www.elclubdelprogramador.com/java-nomenclatura-estandar-del-codigo-java/](http://www.elclubdelprogramador.com/java-nomenclatura-estandar-del-codigo-java/).
51. J.J Gutiérrez, M.J. Escalona, M. Mejías, J. Torres. Pruebas del sistema en Programación Extrema. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. 2010.
52. Miguel A. Jaramillo. Implementación del Detector de Bordes Canny sobre Redes Neuronales Celulares.
53. T. Hermosilla, E.B., A. Balaguer and L.A Ruiz, Detección de bordes con precisión subpixel en imágenes digitales: Interpolación lineal frente a esquemas de tipo no lineal. 2006.
54. Canny, J, F. A Computational Approach to Edge Detection. 1986: p. 679-698.
55. Coto, E. Métodos de segmentación de imágenes Médicas. 2005
56. Saleh Alamri, M.S., Kalyankar, D.M., y Khamitkar, S. Image segmentation by using edge detection. 2010, p: 804-807.
57. Extreme Programming Explained. Beck, Kent. 1999.
58. Yagüey, A.G. J., Las pruebas en metodologías ágiles y convencionales: papeles diferentes., 2009.
59. Sommerville, Ian. Ingeniería del software. Séptima edición. Madrid, España: Pearson Educación. S.A, 2005. 84-7829-074-5.

60. Malfará, Dayvis y otros. Testing en eXtreme Programming. 2006
61. Pressman, R. Ingeniería del Software un enfoque práctico. 2005
62. Canos José H, Letelier Patricio, Panadés M Carmen. Metodologías Ágiles en el Desarrollo de Software. Universidad Politécnica de Valencia, 2003
63. Roger, P. Ingeniería de Software. 6ta edición ed. 2011