Universidad de las Ciencias Informáticas

Facultad 3

Algoritmo de superresolución para imágenes oblicuas provenientes de la lámpara de hendidura.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores:

Elizabet Peña Machado

Yoandy García Barrios

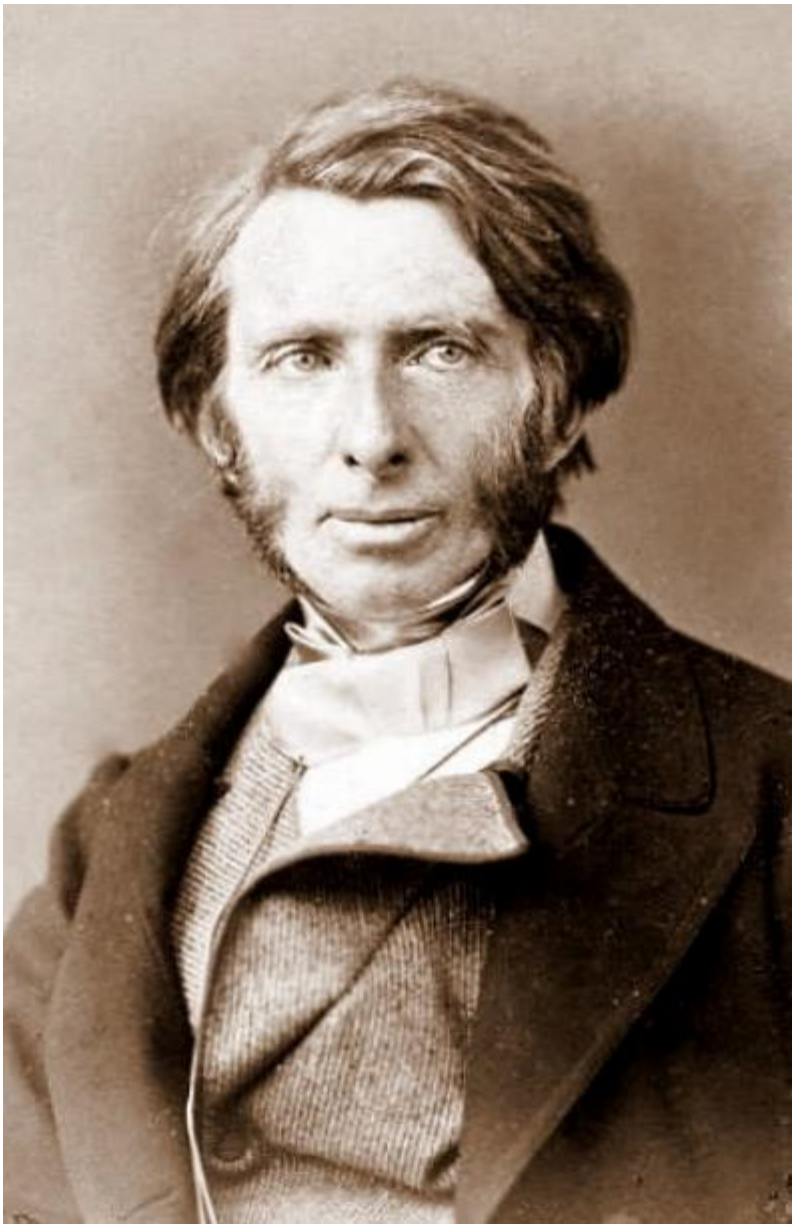
Tutores:

Ing. Michel Álvarez Cancio

Ciudad de La Habana, Junio de 2017

Frase

“La calidad nunca es un accidente, siempre es el resultado del esfuerzo de la inteligencia**”**



John Ruskin

Declaración de Autoría

Declaro ser el autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Elizabet Peña Machado

_____

Firma del autor

Yoandy García Barrios

_____

Firma del autor

Ing. Michel Álvarez Cancio

_____

Firma del tutor

Dedicatoria

Dedicado a mi familia, con todo el amor y cariño del mundo y principalmente a mí por el esfuerzo y sacrificio en el desarrollo de la presente investigación.

Elizabet Graciela Peña Machado

Dedicatoria

Dedicado a mi familia y seres queridos que aunque algunos ya no estén físicamente, están presente siempre en mí.

Yoandy García Barrios

Agradecimientos

Agradezco a:

Dios por su infinita misericordia y permitirme llegar a este punto de mi vida.

Mami y papi por su amor.

Mis hermanos por darme alegría.

Mis tías, abuelas, primos.

Amador por ayudarme muchos años y ser alguien especial.

Isimiyaki por su apoyo.

Rey por ayudarme.

A mí por alentarme siempre.

Elizabet Graciela Peña Machado

Agradecimientos

Agradezco a:

Mí mismo (debo decir), aunque otros dirían a Dios.

Mi madre y mi padre por haber sido divinos al moldearme como soy.

Mi compañera, por ese hermoso tiempo que compartimos y también los feos, yo me quedo con lo mejor.

Mis bellas hermanas, por tenerme tan mortificador a lo largo de los tiempos.

Mis abuelas y abuelos, tías y tíos, primas y primos.

Mis pocas amigas y pocos amigos.

Mis profesores.

Mis socias y socios.

Esos que se creen mis enemigos, en especial, pues sin ellos no podría haber llegado tan lejos ni tan alto.

Yoandy García Barrios

Resumen

En la actualidad existen enfermedades que afectan la visión como es el caso de las cataratas. Esta constituye la principal causa de ceguera tratable con cirugía. 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 o mediano plazo. Unas de las imágenes empleadas en la identificación de la misma, son las imágenes oblicuas 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 técnica de Estimación de Movimiento, mediante el método de Vandewalle, y la Reconstrucción, empleando el método de Interpolación Bicúbica para obtener una imagen resultante de mayores dimensiones y luego aplica una Convolución Normalizada Adaptativa. Al resultado obtenido se le aplica el detector de bordes Canny para medir el porciento de bordes como indicador de calidad. Demostrando que las imágenes resultantes no solo son de mayores dimensiones que la original, sino que presentan mayor nivel de detalles.

Palabras Claves: Convolución Normalizada Adaptativa, imagen oblicua, Opacidad de la Cápsula Posterior, superresolución, Vandewalle.

Índice

Índice de Figuras:

Índice de tabla

Introducción

Existen en todo el mundo un número significativo de enfermedades como es el caso de las cataratas, provocando pérdida total o parcial de la visión. Cataratas es una opacidad del cristalino del ojo, trayendo como consecuencia que la luz se disperse dentro del ojo y no sea posible enfocar en la retina, de esta forma crea una serie de imágenes difusas. Los primeros síntomas de cataratas son la sensación de niebla o deslumbramiento y la disminución de la visión [1]. Es la causa más común de ceguera tratable con cirugía, tiene diversas causas, pero se le atribuye mayormente a la edad, acelerando este proceso si el paciente padece de enfermedades como la diabetes o hipertensión. Con mayor frecuencia esta enfermedad tiende a aparecer en pacientes mayores de cincuenta años de edad [2].

Tras una intervención quirúrgica el paciente puede recuperar su visibilidad total o parcialmente, pero no en todos los casos la cirugía es un éxito a largo plazo, pues en muchos de ellos el paciente puede presentar complicaciones postoperatorias. La complicación postoperatoria más frecuente a largo o mediano plazo es la opacidad de la capsula posterior (OCP) [2], tras la cirugía el paciente puede presentarla, estando la misma asociada con el deterioro de la sensibilidad al contraste, problemas de deslumbramiento y disminución de la agudeza visual, las cuales conllevan importantes repercusiones sociales, médicas y económicas [3]. Según un análisis de 90 estudios publicados en la actualidad su incidencia se encuentra entre 0,7% y 47,6 % en los primeros cinco años de la cirugía [4], siendo en Cuba una cifra considerable 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].

En el mundo se han desarrollado en la actualidad varios softwares para la cuantificación de la opacidad en el ojo humano. Estos sistemas trabajan solamente con imágenes en retroiluminación obtenidas de las lámparas de hendidura, posteriormente estas imágenes son analizadas por los especialistas, quienes deben identificar las estructuras de opacidad dentro de las imágenes. Con el objetivo de identificar la OCP, en la Universidad de las Ciencias Informáticas se desarrolló el software PANDOC, el cual es capaz de cuantificar y detectar diferencias de opacidad, para lograr una evaluación objetiva del grado de la misma. Para ello es necesario el análisis de las imágenes en iluminación oblicua, proveniente de la lámpara de hendidura. En este proceso los especialistas tienen un papel esencial, debido a que, en correspondencia con su nivel de experticia, será el diagnóstico emitido, debiendo identificar manualmente los puntos que se consideran opacos [5].

En este orden, la calidad de las imágenes de OCP del paciente, obtenidas de la lámpara de hendidura, juega un papel fundamental. La ausencia de detalles y niveles altos de ruido o desenfoque, dificultan la realización de un diagnóstico adecuado por el especialista, por lo cual se hace necesaria la restauración o mejora de la calidad.

Dada la situación problemática planteada anteriormente se identifica el siguiente **problema a resolver**: ¿Cómo restaurar detalles en las imágenes oblicuas provenientes de la lámpara de hendidura?

Se identifica como **objeto de estudio**: la superresolución (SR) de imágenes oblicuas, enfocado en el **campo de acción**: los algoritmos de SR de imágenes digitales.

Para dar solución al problema planteado se define como **objetivo general**: diseñar un algoritmo para la restauración automática de detalles en imágenes oblicuas provenientes de una lámpara de hendidura, posibilitado de esta manera la mejor calidad de la imagen para el diagnóstico del paciente por el especialista.

Se plantea la siguiente **idea a defender**: con la aplicación del algoritmo de SR, se restaurarán detalles en las imágenes de OCP provenientes de la lámpara de hendidura, obteniendo una imagen con mayor resolución y definición, posibilitando así un análisis especializado más adecuado.

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

- Identificación de las diferentes técnicas de procesamiento de imágenes médicas.
- Revisión de las técnicas de filtrado y mejora en imágenes médicas.
- Estudio del uso de técnicas de Estimación de movimiento (EM) en imágenes digitales.
- Estudio del uso de la interpolación en imágenes digitales.
- Estudio del uso de técnicas de Reconstrucción en imágenes digitales.
- Descripción de los pasos a seguir para el uso del algoritmo propuesto.
- Diseño de la solución propuesta en función de los requisitos especificados.
- Implementación de las técnicas de filtrado y mejora propuestos.
- Implementación del algoritmo de SR propuesto.
- Validación del método de SR escogido a través de un caso de estudio.
- 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 varios **métodos científicos**, los cuales se mencionan a continuación.

Métodos empíricos:

- **Observación**, como método consiste en la percepción directa del objeto de investigación, es el instrumento universal del científico. La observación permite conocer la realidad mediante la percepción directa de los objetos y fenómenos [4]. Este método fue utilizado para la observación y estudio de los diferentes algoritmos en su aplicación a diferentes

muestras, identificando además cuáles eran los más adecuados para la obtención de los mejores resultados.

Métodos teóricos:

- **Analítico-Sintético**, es un procedimiento teórico mediante el cual un todo complejo se descompone en sus diversas partes y cualidades. Por su parte la síntesis establece mentalmente la unión entre las partes previamente analizadas y posibilita descubrir las relaciones esenciales y características generales entre ellas. Estas operaciones no existen independientemente una de otra, el análisis de un objeto se realiza a partir de la relación que existe entre los elementos que conforman dicho objeto como un todo; y a su vez, la síntesis se produce sobre la base de los resultados previos del análisis [6]. La aplicación de este método permite caracterizar los algoritmos de SR y comprender la problemática existente.

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

Capítulo 1. "Fundamentación teórica": en este capítulo se abordarán de manera sintetizada los elementos básicos teóricos. También se expondrán brevemente los algoritmos de SR presentes en la literatura consultada. Estará enfocando a dos elementos importantes en los algoritmos de SR, la EM y la R como técnicas, las cuales se utilizarán para dar solución a la problemática planteada. 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 las Historias de Usuarios (HU), fase de planificación, plan de iteraciones, tarjetas CRC, el estándar de codificación y el patrón de diseño utilizado.

Capítulo 3: "Resultados y validación": 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: Fundamentación teórica

El procesamiento digital de imágenes ha sido de gran utilidad en la sociedad, en la criminalística, medicina, biología, geología entre otras. Para el desarrollo del mismo en la actualidad se han realizado numerosos métodos que permiten mejores resultados en el trabajo con imágenes, entre ellos se destaca la SR. En el presente capítulo se realiza la descripción y análisis desde el punto de vista teórico del problema general en que se enmarca la investigación. Como puntos fundamentales se abordan los principales conceptos asociados al dominio del problema. También se expondrán

brevemente los algoritmos de SR presentes en la literatura consultada, se caracterizan las herramientas y lenguajes, así como la metodología de desarrollo.

1.1 Procesamiento digital de imágenes: principales conceptos

El procesamiento digital de imágenes 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 [7].

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 radica en que se reduce la variabilidad del observador y aumenta la exactitud, aunque en ocasiones no son los más favorables ya que estos sistemas trabajan con imágenes obtenidas de las lámparas de hendidura [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 más detallada, y dar cumplimiento a los objetivos propuestos. A continuación se exponen los principales conceptos relacionados con el procesamiento digital de imágenes, garantizando un mejor entendimiento.

- **Imagen digital:** es un arreglo bidimensional de píxeles, donde el valor de cada píxel se representa mediante una función f, donde f(x, y) representa el nivel de brillantez, color o intensidad de la imagen en tales coordenadas. Por lo tanto, una imagen en blanco y negro puede ser representada por una matriz de dimensión M×N. Además de la representación en blanco y negro, otras opciones son las imágenes en tonos de gris, para las cuales f(x, y) representa un nivel de intensidad típicamente asociado a un entero entre 0 y 255 [9]. Ver Fig 1.

Fig 1. Tratamiento de imágenes

- **Captura de imágenes:** la idea general de la captura o adquisición de una imagen, es llevar la imagen dentro de la computadora, donde pueda ser almacenada o visualizada para luego ser manipulada y mejorada. El principal elemento en la adquisición de una imagen es una cámara que captura las imágenes de un objeto. Entre las modalidades de adquisición de

imágenes médicas se encuentra el Pentacam, el cual es capaz de reconstruir imágenes tridimensionales de alta resolución (AR) del polo anterior del ojo, y la lámpara de hendidura, la cual es uno de los instrumentos de diagnóstico más comúnmente utilizados por un oftalmólogo.

- **Resolución de una imagen:** indica el número de píxeles que contiene la imagen por área. Se suele medir en píxeles por pulgadas o píxeles por centímetro. Es de gran importancia tener en cuenta, dado a que es directamente proporcional con la calidad de la imagen, por tanto, a mayor resolución, mayor será la calidad de presentación, pero se tendrá como inconveniente que el archivo gráfico ocupe más espacio en disco [9]. La resolución se refiere a la cantidad de detalle en una imagen. Por lo general cuanto más detalle tenga una imagen, mayor será la resolución. En la Fig 2 se muestran los resultados del aumento de la resolución espacial de una imagen, como la primera muestra es de 10 píxeles y luego evoluciona hasta 350 píxeles de ancho.

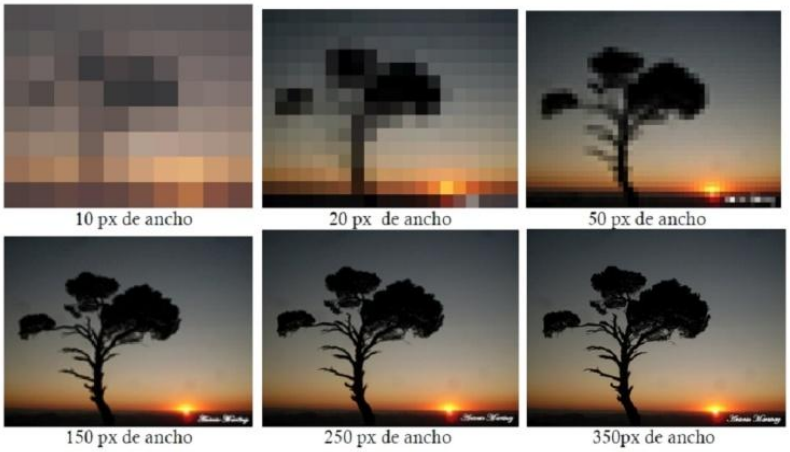


Fig 2. Resultados del incremento de la resolución espacial de una imagen

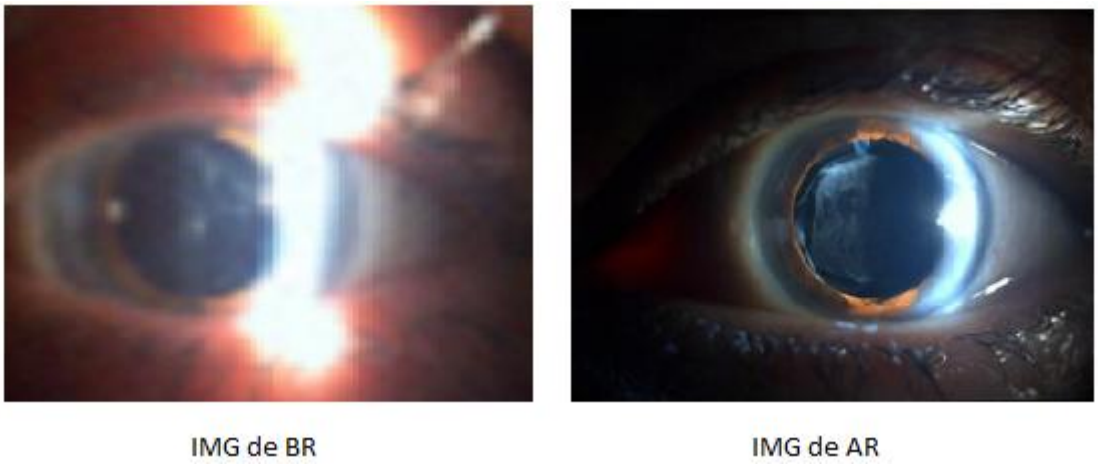


Fig 3. Resolución espacial

- **Lámpara de hendidura:** es el microscopio con el que se mira el ojo. Tiene un lugar donde el paciente apoya la cabeza (un hueco para el mentón y otro para la frente). Así queda fija la cabeza y los ojos están en una posición estable para enfocar el aparato. Unas lentes de aumento llevan la imagen aumentada a través de unos oculares hasta el oftalmólogo. Hay unos mandos para mover la lámpara hasta la posición exacta y centrar la imagen en un ojo u otro. También se puede modificar los aumentos, e incluso podemos desplazar todo el microscopio a los lados para obtener imágenes más oblicuas del ojo [11]. Proporciona iluminación y magnificación para examinar las varias partes del ojo. La luz se proyecta como una franja o hendidura brillante, lo que permite el examen detallado del ojo en pequeños segmentos. Se utiliza en el examen del segmento anterior del ojo, incluyendo el lente cristalino. Con lentes suplementarios la lámpara de hendidura es útil en el examen de la región posterior del ojo, el fondo del ojo y buena parte de la retina [12].
- **Imagen en iluminación oblicua:** la iluminación oblicua se puede usar para revelar estructuras internas y morfología de varios materiales que aparecen casi transparentes bajo iluminación de campo claro y que no se pueden tratar otros métodos de contraste. Estas pueden incluir células vivas, tejidos y organismos enteros, cristales químicos, vidrio o materiales sintéticos transparentes.

En el procesamiento digital de imágenes es importante el uso de la SR, pues esta arroja resultados satisfactorios expuestos a continuación.

1.2 Superresolución

Se conoce como SR de una imagen a la fusión de la información de varias imágenes tomadas a partir de una misma escena, para poder representar detalles que en un principio no son apreciables en las imágenes originales, normalmente a partir de una secuencia de imágenes de más BR. Por tanto, un enfoque prometedor es el uso de técnicas de procesamiento de señal para obtener una imagen o secuencia de AR de múltiples imágenes de BR. Recientemente, la SR, se ha convertido en un área de investigación muy activa. Por lo tanto, a partir de un conjunto de imágenes de BR con un desplazamiento relativo entre ellas muy pequeño, se puede conseguir una imagen de mayor definición [13].

La mayoría de las técnicas de SR con cuadros múltiples se basan en una idea simple: utilizar información de diferentes imágenes con el fin de crear una nueva que aproveche las mejores características de las imágenes utilizadas como base. El primer trabajo de este tipo fue publicado en la década del 80, (Tsai y Huang, 1984) aunque el término SR aparece finalmente en los años 90, (Irani y Peleg, 1990). Hoy está extendido y es la base de las mejoras en dispositivos de captura digital como cámaras fotográficas y filmadoras [14].

1.2.1 Técnicas de superresolución

La idea subyacente de la SR se basa en la adición de nueva información a una trama objetivo mediante la detección de los desplazamientos de sub-píxeles entre dicha trama y sus adyacentes. Estos desplazamientos de sub-píxeles son producidos por la cámara y/o por el movimiento de objetos en la escena. De ahí la importancia de una buena EM. Independientemente del método SR utilizado (dominio de la frecuencia, interpolación uniforme o métodos estadísticos), la técnica subyacente de EM debe ser lo más precisa posible [15]. Mientras que la literatura se puede encontrar en la R de imagen de SR [13, 16], se le ha prestado poca atención a la realización de técnicas de EM cuando se utiliza en problemas de SR [17].

La premisa básica para poder incrementar la resolución espacial en técnicas de SR es la disponibilidad de múltiples imágenes de BR capturadas de una misma escena. Solamente conteniendo cambios en unidades enteras, las imágenes de BR no son útiles para reconstruir la escena con AR, porque no hay información nueva en cada imagen de BR que pueda ser utilizada con este fin. Solo cuando hay nueva información en cuanto a cambios a nivel de subpíxel y el aliasing está presente, es que las imágenes de BR pueden ser utilizadas para obtener una imagen de SR. Para que existan imágenes de BR con nueva información cada una, sin embargo, de una misma escena, es necesario que existan movimientos relativos de la escena de la cual se están obteniendo las imágenes [13, 18, 19]. Las superposiciones de grandes regiones que normalmente existen entre tramas sucesivas de la misma secuencia y el muestreo múltiple de estas regiones en varios marcos, permiten concluir que es posible combinar esta información para conseguir imágenes de mayor resolución espacial. Las técnicas de EM se utilizan para encontrar estas áreas solapadas de un marco a otro. Los vectores de movimiento resultantes deben ser de precisión sub-pixel para proporcionar información útil para la SR.

Las técnicas de SR se relacionan con problemas de interpolación y de restauración de imágenes. Ambas son áreas estudiadas extensivamente. Pero la SR no es ninguna de las dos por separadas, ni ellas son consideradas tampoco técnicas de SR. Las razones fundamentales son, en el caso de la interpolación de imágenes, esta se utiliza para incrementar el tamaño de una sola imagen, pero no aumenta la calidad de la misma en el caso de ser submuestreada (aliasing), es decir, no recupera el contenido de alta frecuencia perdido o degradado durante el proceso de muestreo de BR, por esta razón a los métodos de interpolación de imágenes no se les considera técnicas de SR y el término no significa solamente aumento del tamaño de la imagen, sino aumento también de la definición de las estructuras presentes en la imagen (mediante el aumento de componentes de alta frecuencia) así como de la calidad de la imagen [19]. Esto último es logrado mediante la restauración de imágenes digitales, la cual es un área también muy estudiada y establecida dentro de las aplicaciones de procesamiento digital de imágenes [20, 21].

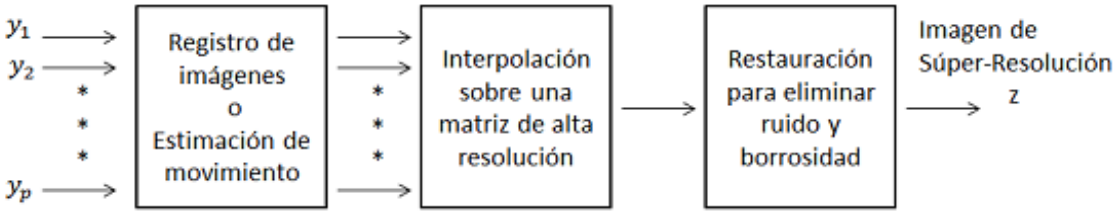


Fig 4. Esquema para SR partiendo de múltiples imágenes con cambios y rotaciones entre ellas

1.2.1.1 Estimación de movimiento

La EM es el proceso de obtención de información de movimiento percibida a través de la información de la imagen actual y una o varias imágenes de referencia.

I. Vandewalle et al. (EPFL): (VA)

Este método [22], utiliza la propiedad de que un cambio en el dominio espacial produce un cambio lineal en la fase de la Transformada de Fourier. También, una rotación en el dominio espacial es visible en la amplitud de la Transformada de Fourier (ver Fig. 5 y Fig. 6). Es decir, para una relación dada por:

2.1 $$f_2(x_1,x_2) = f_1(R(x_1 + \Delta x_1, x_2 + \Delta x_2))$$

con $f_1(x_1, x_2)$ la imagen de referencia, $f_2(x_1,x_2)$ la versión trasladada y rotada de ella, "R" la matriz que se multiplica por el sistema coordenado original provocando la rotación de la imagen, siendo Δx_1 y Δx_2 los factores de traslación de la imagen en cada una de las dimensiones.

$$R = \begin{pmatrix} cos(\Phi) & -sin(\Phi) \\ sin(\Phi) & cos(\Phi) \end{pmatrix}$$

En el dominio de la frecuencia se tiene que:

$$F_2(u,v) = \iint_x f_2(x_1x_2)\, e^{-j2\pi(ux_1 + vx_2)} dx_1 dx_2$$

2.2 $$F_2(u,v) = \iint_x f_1(R(x_1 + \Delta x_1, x_2 + \Delta x_2))\, e^{-j2\pi(ux_1 + vx_2)} dx_1 dx_2$$

$$|F_2(u,v)| = |F_1(R(u,v))|$$

Donde $|F_2(u,v)|$ es una versión rotada de $|F_1(u,v)|$ el mismo ángulo Φ que la rotación en el dominio espacial $|F_1(u,v)|$ y $|F_2(u,v)|$ no dependen de los cambios Δx, dado que los cambios espaciales solo afectan los valores de fase de la Transformada de Fourier [22]. De esta forma, el algoritmo de EM de (Vandewalle et al.) [22] multiplica las imágenes de BR por una ventana Tukey para hacerlas circularmente simétricas, luego calcula la Transformada de Fourier de las mismas y determina los cambios 1D en sus amplitudes y fases para obtener los ángulos de rotación y los cambios espaciales

∆x respectivamente. Una ventaja de este método es que se descartan las componentes de alta frecuencia, donde el aliasing podría haber ocurrido y esto provee mayor robustez.

El procedimiento para la estimación de la rotación en [22], requiere transformar el espectro a coordenadas polares, para luego computar el contenido de frecuencias h como una función del ángulo de rotación mediante la integración sobre líneas radiales, es decir:

2.3 $$h(\alpha) = \int_{\alpha - \Delta\alpha/2}^{\alpha + \Delta\alpha/2} \int_0^{\infty} |F(r,\theta)|\, dr\, d\theta$$

El cálculo de *h*(α) en la práctica (ver [26]), resulta en una función h(α) de 1D para cada espectro de frecuencias $|F_1(u,v)|$ y $|F_2(u,v)|$ (ver Fig.7). El ángulo de rotación exacto puede entonces ser computado como el valor para el cual la correlación entre dichas funciones 1D alcanza el máximo. La estimación del cambio espacial entre las imágenes es obtenida, a partir de que un cambio paralelo al plano de la imagen puede ser expresado en el dominio de la frecuencia como un cambio lineal de fase [22], es decir:

$$F_2(u,v) = \iint_x f_2\,(x_1 x_2)\, e^{-j2\pi(ux_1 + vx_2)} dx_1 dx_2$$

2.4 $$F_2(u,v) = \iint_x f_1\,(R(x_1 + \Delta x_1, x_2 + \Delta x_2))\, e^{-j2\pi(ux_1 + vx_2)} dx_1 dx_2$$

$$F_2(u,v) = e^{-j2\pi(ux_1 + vx_2)} \iint_x f_1\,(x_1 + \Delta x_1, x_2 + \Delta x_2)\, e^{-j2\pi(ux_1 + vx_2)} dx_1 dx_2$$

$$|F_2(u,v)| = e^{-j2\pi(u\Delta x_1 + v\Delta x_2)} |F_1(u,v)|$$

Los cambios ∆x pueden ser computados como la pendiente de la diferencia de fase de las transformadas de Fourier: $2\pi(ux_1 + vx_2) = \angle(F_2(u,v)/F_1(u,v))$.

Finalmente, para todas las frecuencias se escribe la ecuación lineal describiendo un plano a través de la diferencia de fase computada con pendientes ∆x desconocidas, para luego hallar los parámetros de cambio deseados ∆x como la solución mediante mínimos cuadrados de las ecuaciones [26].

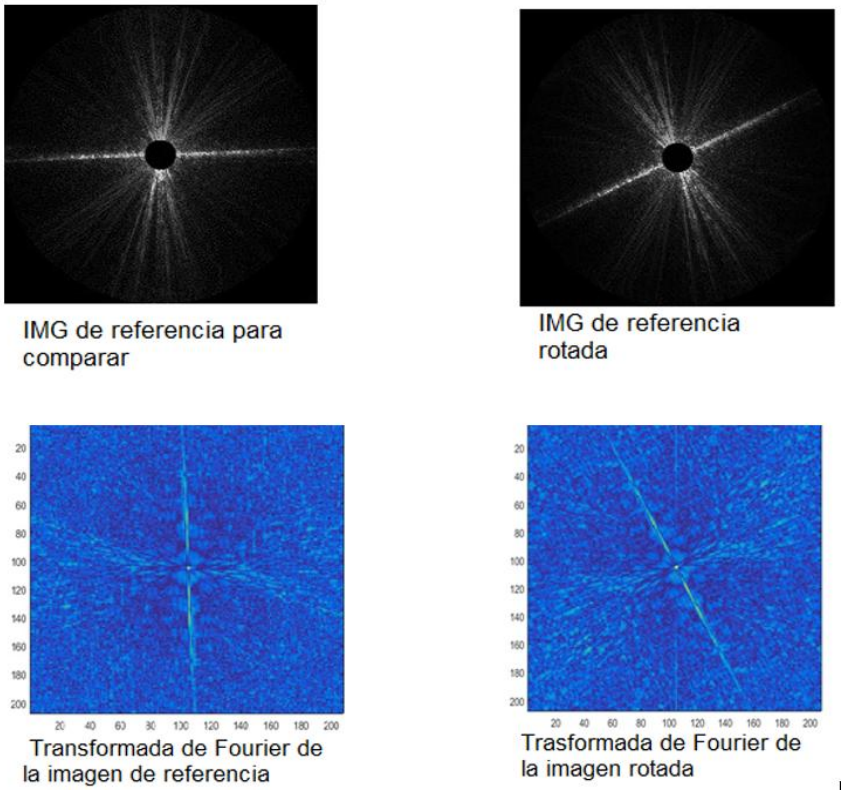


Fig 5. Resultados de la aplicación de la Trasformada de Fourier

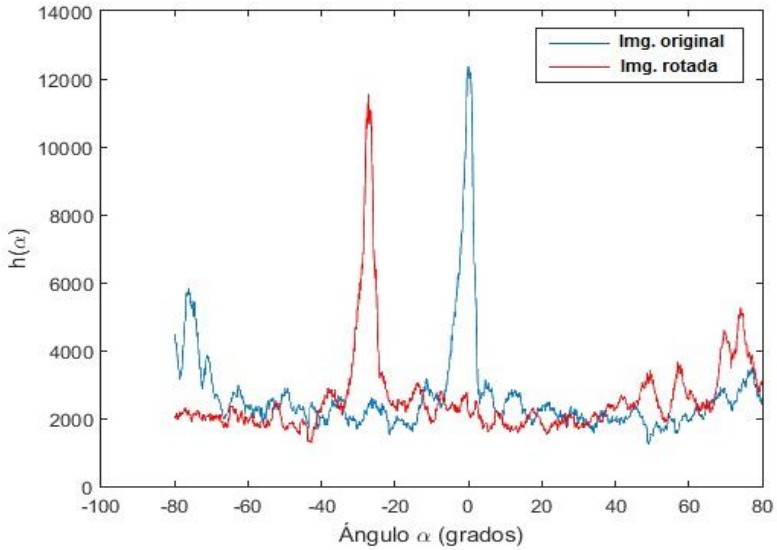


Fig 6. Proyecciones 1D de la imagen original y la de referencia

II. Keren et al.

El algoritmo de EM de (Keren et al.) [23]. Utiliza diferentes versiones submuestreadas de las imágenes a ser analizadas. Primero una versión submuestreada 4x, es utilizada para estimar el cambio y la rotación usando la series de Taylor. Lo mismo es hecho con un submuestreo de 2x, pero después de corregir los cambios y rotaciones detectados antes. Finalmente, se hace lo mismo con la resolución total de las imágenes, para tener estimados más precisos. Así de [23], dados a, b

cambios horizontales y verticales respectivamente y θ el ángulo de rotación alrededor del origen, entre las imágenes g(x,y) y f(x,y) se tiene que:

$$2.5 \qquad \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) = f(xcos(\theta) - ysin(\theta) + a, ycos(\theta) + xsin(\theta) + b)$$

Si se expanden el sin(θ) y cos(θ) a los primeros dos términos en su serie de Taylor se obtiene:

$$2.6 \qquad \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) \approx f(x + a - y\theta - x\frac{\theta^2}{2}, y + b + x\theta - y\frac{\theta^2}{2})$$

Luego, expandiendo f al primer término de su propia serie de Taylor se obtiene la siguiente ecuación de primer orden:

$$2.7 \qquad \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) \approx \mathrm{f(x,y)} + (\mathrm{a} - \mathrm{y}\theta - \mathrm{x}\frac{\theta^2}{2})\frac{\partial \mathrm{f}}{\partial \mathrm{x}} + (\mathrm{b} + \mathrm{x}\theta - \mathrm{y}\frac{\theta^2}{2})\frac{\partial \mathrm{f}}{\partial \mathrm{y}}$$

La función de error entre g y f después de la rotación por θ y la traslación por a y b puede entonces ser aproximada por:

$$2.8 \; E(a,b,\theta) = \sum [f(x,y) + (a - y\theta - x\frac{\theta^2}{2})\frac{\partial f}{\partial x} + (b + x\theta - y\frac{\theta^2}{2})\frac{\partial f}{\partial y} - g(x,y)]^2$$

donde la suma es sobre las porciones solapadas de f y g, pero incluso es necesario una región aún más pequeña que esa [23]. El mínimo de E(a,b,θ) es hallado mediante el computo de sus derivadas y comparando con cero, produciendo así un sistema de tres ecuaciones lineales con las tres variables desconocidas y de interés (a,b,θ), donde R es una abreviatura para $x\frac{\partial f}{\partial y} - y\frac{\partial f}{\partial x}$.

$$2.9 \qquad \begin{aligned} \left[\sum\left(\frac{\partial f}{\partial f}\right)^2\right]a + \left[\sum\frac{\partial f}{\partial x}\frac{\partial f}{\partial y}\right]b + \left[\sum R\,\frac{\partial f}{\partial x}\right]\theta &= \sum\frac{\partial f}{\partial x}(f-g) \\ \left[\sum\frac{\partial f}{\partial x}\frac{\partial f}{\partial y}\right]a + \left[\sum\left(\frac{\partial f}{\partial y}\right)^2\right]b + \left[\sum R\frac{\partial f}{\partial y}\right]\theta &= \sum\frac{\partial f}{\partial y}(f-g) \\ \left[\sum R\,\frac{\partial f}{\partial x}\right]\theta + \left[\sum R\,\frac{\partial f}{\partial y}\right]b + [\sum R^2]\theta &= [\sum R(f-g)] \end{aligned}$$

Como las imágenes son capturadas en intervalos de tiempo discretos, los desplazamientos entre ellas pueden no ser lo suficientemente pequeños para el método de recuperación de movimiento a través de las ecuaciones (2.7) [24]. Luego, debido a las aproximaciones hechas para obtener (2.5), la expresión es correcta solo para pequeños valores de (a,b,θ). Por ello se desarrolla el siguiente proceso iterativo para tener precisión de subpíxel (en este proceso iterativo se hace interpolación a nivel de subpíxel), luego se resuelven las ecuaciones (2.9), utilizando estas soluciones obtenidas, se aproxima g (a través de la ecuación 2.5), a continuación se efectúa la próxima iteración con la nueva g obtenida, hasta un número fijo de iteraciones o hasta que las soluciones sean muy pequeñas (a,b,θ se aproximan a cero). En [23] recomiendan siempre forzar la g original mediante los valores acumulados de a, b, y θ, esto se hace para mantener la precisión.

Para incrementar la velocidad y mejorar la precisión, se utiliza una estructura de datos de pirámide Gaussiana. Primero, se computan los parámetros del movimiento para una imagen de resolución reducida en la pirámide, en la cual incluso grandes desplazamientos devienen pequeños en este nivel de reducción (que es por lo cual se hace esto). Los parámetros del movimiento computados son entonces interpolados hacia una imagen más grande, el movimiento detectado (a,b,θ) es corregido a través de algunas iteraciones, y otra vez es interpolado hacia el próximo nivel de resolución. Este proceso es continuado hasta que la imagen original (de tamaño completo) es obtenida [23, 24].

1.2.1.2 Reconstrucción

Cuando las imágenes de baja resolución se registran con precisión, las muestras de las diferentes imágenes se pueden combinar para reconstruir una imagen de alta resolución.

I. Interpolación

Los algoritmos de interpolación (ampliación) de imágenes se basan en el proceso de calcular valores numéricos desconocidos (píxeles) a partir de otros conocidos mediante la aplicación de algoritmos finitos. La idea central es producir una imagen de tamaño superior a la original, completando la información faltante con datos interpolados a partir de un algoritmo específico. [25]

Interpolación bicúbica (IB): actualmente es uno de los algoritmos de interpolación más utilizado, lo que hace es considerar los 16 píxeles más cercano al que se desea interpolar. O sea, se aproxima localmente el nivel de gris en la imagen original mediante la generación de una superficie polinómica bicúbica [25]. Equivale a aplicar, en primer lugar, cuatro interpolaciones unidimensionales horizontales entre cuatro píxeles (dos a un lado y dos a otro) y posteriormente otra interpolación unidimensional, en este caso vertical, entre los cuatro valores obtenidos anteriormente. El algoritmo completo utilizado se describe en [26].

- Como están situados a distancias distintas del píxel de valor desconocido, se da mayor peso en el cálculo a los más cercanos.
- Produce imágenes muy nítidas.
- Es un buen compromiso entre tiempo de procesado y calidad de resultado.
- Es un procedimiento estándar en programas de edición de imágenes e interpolación en cámaras.

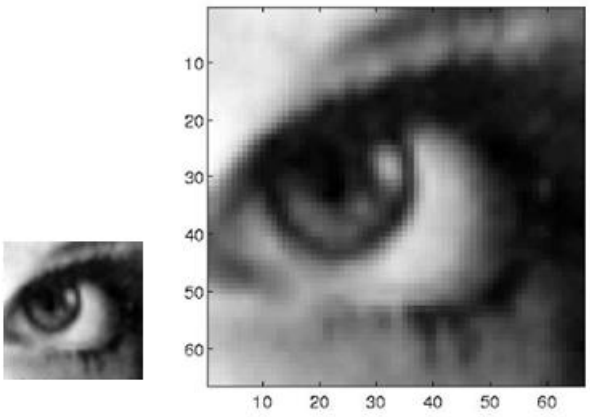


Fig 7. Interpolación bicúbica (Izquierda la imagen original y a la derecha la resultante

II. Papoulis-Gerchberg (PG)

El mismo crea una matriz de AR utilizando los valores de los píxeles conocidos a partir de las imágenes de baja resolución. Luego, va al dominio de la frecuencia para extraer las altas frecuencias (es decir, hace cero las componentes de alta frecuencia en el espectro de frecuencia), forzando entonces los píxeles conocidos a aumentar su amplitud, logrando así predecir algunos de los valores de alta frecuencia, y repite este proceso hasta la convergencia [27].

El mismo se basa en el trabajo realizado independientemente por [27]. Mientras que Gerchberg propuso un método para realizar la reconstrucción de la señal dado el límite de difracción de la señal y una parte del espectro, la motivación para el trabajo de Papoulis fue la extrapolación de una señal de banda limitada de sólo una parte de la señal original, es decir, la determinación de la transformada

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t} dt \qquad 2.10$$

de una señal f(t) dado un segmento finito donde:

$$p_T(t) = \begin{cases} 1, |t| \leq T \\ 0, |t| > T \end{cases} \qquad 2.11$$

La extrapolación de la señal se lleva a cabo por el método de alternancia de proyecciones, [28] iterando alternamente entre dominios de tiempo y espectrales. La señal g(t) es primero filtrada con paso bajo con una frecuencia de corte σ, asumiendo a σ como el ancho de banda de señal de f(t). Esto se ilustra en la formación de $F_1(\omega)$ en la Fig.8 (f) de $G(\omega) = G_0(\omega)$ que se muestra en la Fig.8 (d). En la enésima iteración esto se puede expresar como:

$$F_n(\omega) = G_{n-1}(\omega) p_\sigma(\omega), p_\sigma(\omega) = \begin{cases} 1, |\omega| \leq \sigma \\ 0, |\omega| > \sigma \end{cases} \qquad 2.12$$

Para la ilustración se muestra el filtro $p_\sigma(\omega)$ para ser uno ideal, pero uno es libre de seleccionar un filtro de paso bajo apropiado$p_\sigma(\omega)$. La función inversa de F_1(ω) se calcula entonces como f_1(t) (Fig.8 (e)). Esto resulta en una reducción de la señal de error $|f(t) - f_1(t)|^2$ fuera del segmento conocido de la señal. Esto se deduce del teorema de Parseval. Sin embargo, la señal f_1(t) no coincide con la señal observada g(t) en la región [-T, T]. Esta parte de la señal se restablece entonces al segmento original conocido que forma la función g_1(t) para la siguiente iteración como se muestra en la Fig.8 (g). El cambio resultante en el dominio espectral debido a la introducción de componentes de mayor frecuencia puede verse en la Fig.8 (h). Generalizando esto para la enésima iteración obtenemos:

$$g_n(t) = f_n(t) + [f(t) - f_n(t)]p_T = \begin{cases} g(t), |t| \leq T \\ f_n, |t| > T \end{cases} \quad 2.13$$

Este proceso se itera con el nuevo g_1(t) así formado. En cada iteración se reduce el error cuadrático medio de la señal extrapolada [24]. Por tanto, con iteraciones sucesivas, la señal extrapolada generada se aproxima a la señal deseada f(t). La convergencia del método está garantizada y se muestra en [24]. Sin embargo, el proceso requiere un número infinito de iteraciones. Si nos detenemos después de r iteraciones, la señal reconstruida viene dada por$f_r(t)$ en lugar de f(t). También, en la práctica, los datos medidos g(t) = g_0(t) contendrán error. La propagación de este error de medición puede ser controlada por la terminación temprana del proceso iterativo [29]. El proceso también asume que la señal f(t) es limitada en bandas, pero se encuentra que el método funciona razonablemente bien para señales con energía suficientemente baja en sus componentes de frecuencia más alta. A continuación, se muestra la extrapolación iterativa de g(t) utilizando el método de Papoulis-Gerchberg. (a) la señal que se desea recuperar, (b) espectro de la señal, (c) la señal disponible temporalmente g(t) = g_0(t) utilizada como señal estimada inicial, (d)espectro de (c), (e) el tiempo de señal de (f),(f) espectro filtrado con paso bajo de (d), (g) porción de (c) reintegrada en (e).

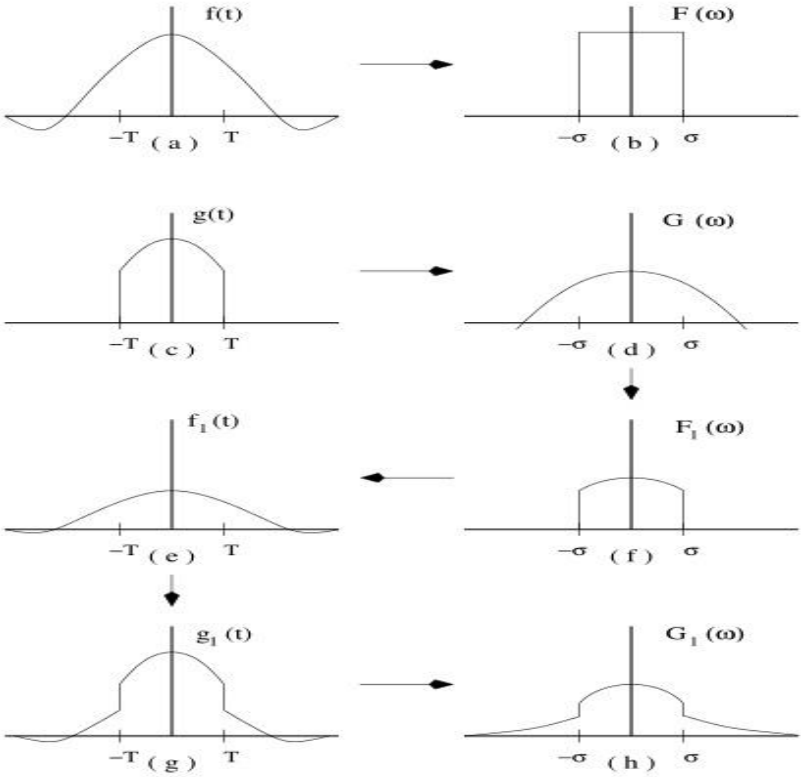


Fig 8. Ilustración de la extrapolación iterativa

III. Iterated Back Projection (IBP)

Comienza con una estimación de la imagen de AR, luego el proceso de obtención de la imagen es simulado para obtener un conjunto de imágenes de BR. Seguidamente se calcula la diferencia entre la imagen observada y las de BR creadas, e iterativamente se le añade esta imagen gradiente a la imagen supuesta inicialmente de AR, mediante un proceso conocido como *back projecting* [24], para mejorar la estimación supuesta inicialmente con la imagen AR. Esta imagen gradiente es la suma de los errores entre cada imagen de BR y la imagen de AR estimada. Este proceso se repite iterativamente hasta que el error sea mínimo. De los pasos críticos considerados para el método IBP, el primero es construir el modelo para el proceso de imagen que se puede describir como:

$$g_k(y) = DH^{psf} \times f(x) + n_k \qquad 2.14$$

Donde la función g_k representa las k-ésimas imágenes de BR observadas, y denota el píxel de las imágenes de BR influenciadas por el área de x de la imagen de SR f,H^{psf} es la Función de Dispersión de Punto del núcleo de desenfoque, D es el operador de decimado, yn_k es un término de ruido aditivo. El segundo paso es el registro donde en primer lugar se asume una imagen de SR verdadera. Basado en el modelo de imagen dado en la ecuación 2.14, se evalúan diferentes imágenes de BR. Dadas las imágenes de BR calculadas, se obtiene una nueva imagen de SR. Posteriormente, esta nueva imagen de SR se utiliza para generar el nuevo conjunto de imágenes de BR. Si este nuevo conjunto de imágenes de BR es el mismo que el conjunto anterior, entonces

la imagen de SR adoptada es la imagen de SR verdadera, de lo contrario la imagen de error obtenida a partir de la diferencia entre las imágenes de BR se proyecta de nuevo a la imagen de SR sujeta. Este proceso se repite hasta que no queda ninguna imagen de error IBP se puede representar matemáticamente como:

$$f^{(n-1)}(x) = f^{(n)}(x) + \sum_y (g_k(y) - g_k^n(y)) \times H^{BP} \quad 2.15$$

Donde $f^{(n)}$es la imagen de SR estimada después de la iteración n, $g_k(n)$representa las imágenes de BR calculadas del modelo de imagen de $f^{(n)}$ después de la enésima iteración y H^{BP} es el núcleo de retroproyección. La técnica de SR se beneficia de diferentes métodos de interpolación y técnica de registro IBP. La técnica de SR también se compara con la interpolación bicúbica convencional y el relleno de onda nula (WZP por sus siglas en inglés), así como con la técnica del estado del arte propuesta [24]. Primero la imagen de BR es interpolada usando diferentes modelos de interpolación y entonces es diezmada a cuatro imágenes de BR. Las cuatro imágenes de BR obtenidas mediante la toma de algunas imágenes secuenciales o generadas artificialmente mediante el uso del filtro de desenfoque, son interpoladas. Las mismas no son nítidas y debido al suavizado causado por la interpolación, se requiere una nitidez. Por lo tanto, estas cuatro imágenes interpoladas se utilizan como entrada para la técnica de registro de IBP. Finalmente, la imagen de salida AR vuelve al primer paso. Este proceso se repite hasta que el valor de la función de error se haga más pequeño. En el método propuesto después de cada iteración, la imagen de salida tiene mejor resolución y en la próxima iteración la imagen de entrada es mejor que la primera. A continuación, se muestra el proceso estándar de IBP de súper resolución de una imagen de nxn a una de αn x αn.

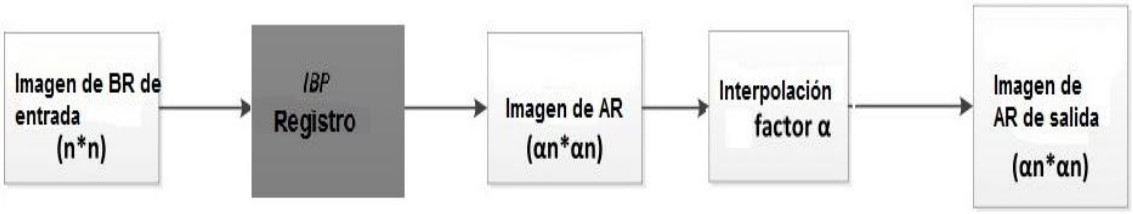


Fig.10 Proceso estándar de IBP de superresolución.

IV. Convolución Normalizada Adaptativa (CNA)

Structure-Adaptive Normalized Convolution como aparece en la literatura. Este algoritmo [30], utiliza la convolución normalizada para reconstruir una imagen de AR, a partir de varias imágenes de BR. El mismo posee robustez ante ruido; lo cual analiza las imágenes y determina cuáles píxeles son ruidosos para no utilizarlos en la reconstrucción; y en una segunda iteración se realiza una segunda corrección, la cual adaptaría el tamaño y la orientación de los filtros Gaussianos utilizados en la convolución normalizada, resultando en una imagen de alta resolución más definida. Los pasos son los siguientes: después que se tienen las imágenes registradas se detecta el movimiento entre ellas, en un marco común con precisión de subpíxel, se aplica una fusión robusta [30], utilizando la

convolución normalizada adaptativa a las imágenes de BR a las que se le corrigió el movimiento. La deconvolución finalmente reduce el ruido y los efectos borrosos causados por el proceso de captura de la imagen. Este proceso de fusión consta de tres pasos fundamentales:

El primero es basado en una operación local de mediana con peso para obtener un primer estimado de la imagen de AR (AR 0). Entonces se utiliza AR 0 como una estimación inicial para una convolución normalizada de primer orden en el segundo paso, que produce un mejor estimado de la imagen de AR (AR 1) y dos imágenes derivadas en las direcciones x, y (ARx y ARy) respectivamente. Estas imágenes derivadas resultantes son utilizadas entonces como entrada en el tercer paso para construir funciones de aplicabilidad anisotrópicas y la convolución normalizada adaptativa final. Cada paso mejora la imagen de AR estimada.

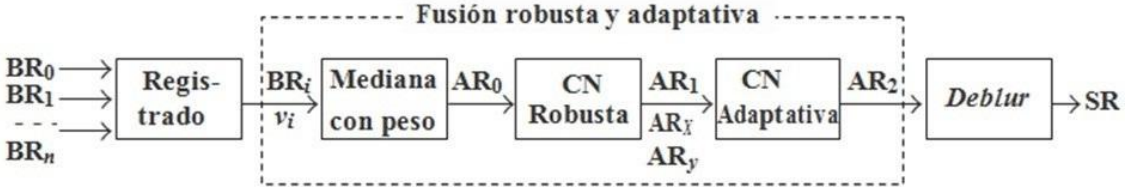


Fig.11 Convolución Normalizada Robusta y Adaptativa

La función de aplicabilidad adaptativa, es una función Gaussiana anisotrópica, cuyo eje principal es rotado para alinear con la orientación local dominante, es decir, esta función adapta su forma y su orientación a lo largo de la estructura subyacente de la imagen [30]:

$$2.16 \quad a(s,s_0) = \rho(s-s_0)exp\left\{-\left[\frac{(xcos(\varphi)+ysin(\varphi)}{\sigma_u(s_0)}\right]^2-\left[\frac{-xsin(\varphi)+ycos(\varphi)}{\sigma_v(s_0)}\right]^2\right\}$$

con s0 = (x0, y0) el centro del análisis, s − s0 = (x, y) son las coordenadas locales de las muestras de entrada con respecto a s0, ρ es una función centrada en el origen que limita el soporte del núcleo (kernel) a un radio determinado, σu y σv son las escalas direccionales del núcleo Gaussiano anisotrópico, específicamente, σv es la escala a lo largo de la orientación alargada y es mayor o igual que σu , siendo:

$$2.17 \qquad \sigma_u = \frac{\alpha}{\alpha+A}\sigma_c\sigma_v = \frac{\alpha+A}{\alpha}\sigma_c$$

Donde α es un parámetro de ajuste del límite superior de la función de aplicabilidad Donde σc es una escala local y es quien ajusta las dos escalas direccionales [30]. Es decir, σc permite que la función de aplicabilidad se achique o se agrande en dependencia de cuan densamente poblado es el vecindario.

1.2.2 Métodos seleccionados

Para darle solución a la problemática planteada una vez analizadas las técnicas y métodos correspondientes antes mencionados, se escogió Vandewalle para la EM, ya que cuenta con la ventaja de descartar los componentes de alta frecuencia, brindando mayor robustez y evitando tratar este elemento y así optimizar tiempo. En la etapa de reconstrucción se decidió emplear la IB con la intención de agrandar las dimensiones de la imagen de AR obtenidas, sin afectar la resolución, siendo uno de métodos más utilizados en el procesamiento digital de imágenes con este propio objetivo, ya que por sí sola, no aumenta la resolución. Finalmente se aplica el algoritmo CNA, ya que ofrece la posibilidad de trabajar con robustez ante ruido y a la vez efectuar una segunda iteración de corrección de la imagen, adaptando el tamaño y la orientación de los filtros gaussianos utilizados en la convolución normalizada, resultando finalmente en una imagen de AR mejor definida. Los mencionados métodos además fueron empleados en la experimentación con imágenes confocales de la córnea en [31], las cuales son muy similares a las que se abordan en la presente investigación, arrojando los mejores resultados.

1.3 Herramientas y metodología utilizadas para el desarrollo

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

1.3.1 Lenguajes utilizados

Un lenguaje de programación es un conjunto de instrucciones que permiten la creación de un programa, que luego será ejecutado por un ordenador. Para la realización del sistema se utilizaron los siguientes lenguajes.

1.3.1.1 Lenguaje de programación Matlab

Matlab permite a la hora de programar una serie de elementos típicos para la modificación del flujo de una secuencia de instrucciones. La sintaxis es muy parecida a la de cualquier lenguaje de programación. Todos estos operadores se pueden usar en la ventana de comandos, en línea, o en un fichero “.m”. La programación se lleva a cabo mediante un lenguaje que es muy parecido a lenguajes de alto nivel como BASIC o C. Esto permite que el usuario pueda agrupar sentencias que utiliza frecuentemente dentro de un programa que puede ser invocado posteriormente. De este modo se ahorra tiempo y esfuerzo en sucesivas sesiones pues no es necesario escribir todas las sentencias de nuevo [32].

1.3.1.2 Lenguaje unificado de modelado

UML es el estándar industrial de la notación de modelado para sistemas orientados a objetos, constituye la plataforma inicial para el desarrollo rápido de aplicaciones. Es un lenguaje gráfico para

visualizar, especificar, construir y documentar artefactos de un sistema. Se utilizan para crear diagramas que representan alguna parte o punto de vista del sistema. Además, sirve de apoyo a la mayoría de los procesos de desarrollo orientados a objetos, capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Permite la división de grandes sistemas en piezas de trabajo más sencillas cuando se labora en un entorno complejo [33].

1.3.2 Herramientas de desarrollo

Las herramientas de desarrollo de software son un programa utilizado por un programador para crear, estas se clasifican según los servicios que ofrece y/o la tarea a la que da soporte.

1.3.2.1 Herramientas CASE

Las herramientas CASE ("Ingeniería de Software Asistida por Computadora"), constituyen diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de aplicaciones, reduciendo el coste de las mismas en función de tiempo y dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de un producto informático, en tareas como el proceso de diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación y detección de errores. Abarcan todos los pasos del proceso de software y también aquellas actividades generales que se aplican a lo largo del mismo [34].

Visual paradigm: permite modelar con UML, sincronizar modelos y código, realizar ingeniería inversa, generar código automáticamente, así como diseñar bases de datos y generar sus esquemas con el lenguaje de definición de datos del sistema gestor de base de datos seleccionado. Es multiplataforma y cuenta con una versión libre para la comunidad. Soporta la revisión ortográfica, brindando sugerencias para los idiomas: inglés, español, francés, alemán y portugués [35]. Por lo que se utilizó por sus prestaciones, en la realización de los diagramas correspondientes al sistema.

1.3.2.2 Entorno integrado de desarrollo

Un entorno integrado de desarrollo IDE, es un editor de código fuente, es una herramienta diseñada para la construcción automática. Estos cuentan con un autocompletado inteligente. Algunos IDE contienen un compilador, un intérprete o ambos.

Matlab (abreviatura de MatrixLaboratory, laboratorio de matrices), es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X. Es una poderosa herramienta para la resolución numérica de problemas. Matlab ofrece un entorno interactivo sencillo mediante una ventana en la que podemos introducir órdenes en modo texto y en la que aparecen los resultados [36]. Los gráficos se muestran en ventanas independientes. Cada ventana dispone de una barra de menú que controla su funcionalidad. Lo que distingue a MATLAB de otros sistemas de cálculo es su facilidad para trabajar con vectores y matrices [36]. Las operaciones ordinarias,

suma, producto, potencia, operan por defecto sobre matrices, sin más restricción que la compatibilidad de tamaños en cada caso.

1.3.3 Metodología de desarrollo de software

El proceso de desarrollo de software, es definido como el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software, tiene como finalidad la obtención de un producto que cumpla con las expectativas del cliente [37].

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

Luego de la evaluación de las 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 [38].

El cliente es parte del equipo de desarrollo, el equipo de desarrollo es de solo una persona, además de la dificultad 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 [39] [37], lo cual permitió identificar la Programación Extrema (XP) como una alternativa acertada, además de ser la metodología de desarrollo del grupo de investigación AIRI(Artificial Intelligence: Research and Innovation).

Ventajas de XP [39]:

- 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]:

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.

Diseño: La metodología XP hace especial énfasis en los diseños simples y claros. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.

Implementación: En la fase de codificación de desarrolla en función de cada historia de usuario, además de ser fase donde se definen las tareas de la ingeniería y los tiempos reales en se realizaron cada una de las funcionalidades especificadas, en la cual la implementación, debe realizarse de acuerdo los estándares de codificación.

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 de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para prever que no vuelvan a ocurrir.

1.4 Conclusiones del capítulo

Después de analizar la SR como parte del procesamiento digital de imágenes, se comprobó que garantiza mejorar la resolución espacial de estas. La SR trabaja en base 2 o más tomas de un objeto para lograr una mejor definición de los detalles aplicando determinadas técnicas y métodos. Para esto se analizaron dos técnicas: EM y R, donde se escoge el método de Vandewalle y CNA, respectivamente pues permiten obtener mejores resultados, teniendo como paso intermedio el empleo de la IB para aumentar solamente las dimensiones de la imagen de AR resultante. Fueron aplicadas a las imágenes oblicuas obtenidas, de la lámpara de hendidura, en el Instituto de oftalmología. Se estudiaron las herramientas, lenguajes y la metodología de desarrollo de software, destacándose la metodología XP y la herramienta Matlab.

Capítulo 2: Propuesta de solución

Obtener un producto informático con calidad, que satisfaga las necesidades del cliente, implica un dominio previo de todos los procesos involucrados en este, donde se aplican diferentes métodos y técnicas. En este capítulo se expone la solución propuesta, se levantan los requisitos del sistema. Se proponen un estándar de codificación y el patrón de diseño empleado.

2.1 Descripción de la solución

Para establecer la propuesta de solución se seleccionaron diferentes métodos de SR, los cuales se mencionan en las técnicas de SR expuestas anteriormente, con el objetivo mejorar la resolución espacial de las imágenes oblicuas provenientes de la lámpara de hendidura, de tal forma que permita al especialista una visión más amplia y con más calidad para los pacientes operados de cataratas. Por tanto, se expone una secuencia de pasos donde se describe la solución propuesta.

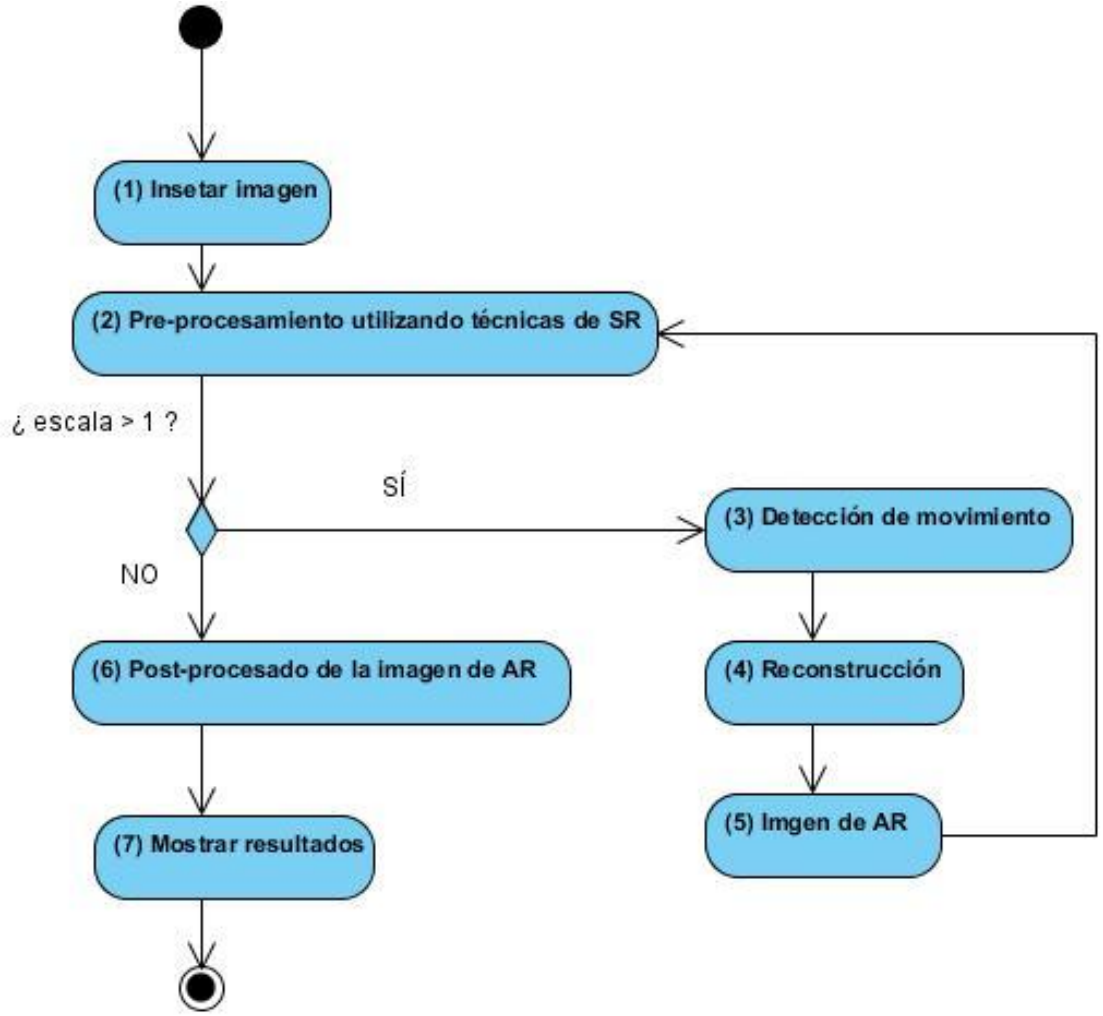


Fig 9. Descripción del algoritmo de SR propuesto

Paso 1. INSERTAR IMÁGENES

El usuario debe insertar una secuencia de imágenes de BR de una misma escena, donde los objetos presentes pueden estar trasladados o rotados (estas imágenes deben estar en el mismo directorio). Una vez cargadas, se define el factor de interpolación e inicia el procesamiento. Este factor de interpolación (escala) denota el tamaño final de la imagen de AR estimada con respecto al de la secuancia de BR insertada. La primera imagen que se inserte en la secuencia es la que será tomada como imagen de referencia, tomando al resto como versiones trasladadas y/o rotadas de la misma.

Paso 2. PRE-PROCESAMIENTO UTILIZANDO TÉCNICAS DE SR

Para realizar el pre-procesamiento se utilizan las técnicas de SR, se obtienen la cantidad de muestras por pixel que presenta la imagen, información que se puede obtener de las imágenes en formato **tif** y se transforma la imagen a una imagen de doble precisión y se obtiene una matriz de intensidades.

Paso 3. ESTIMACIÓN DE MOVIMIENTO

Antes de iniciar el método de estimación de movimiento por el método Vandewalle, se multiplican las imágenes por una ventana de tukey, para hacerlas simétricas circularmente. Después se realiza un cálculo del centro de las imágenes y del centro de su Transformada de Fourier, se calcula un promedio de la versión centrada de la Transformada de Fourier de cada una para estimar rotación y se buscan diferencias de fase para detectar traslación. Luego multiplica una matriz por el sistema de coordenadas original provocando una rotación, tomando en cuenta los factores de rotación y traslación de esta.

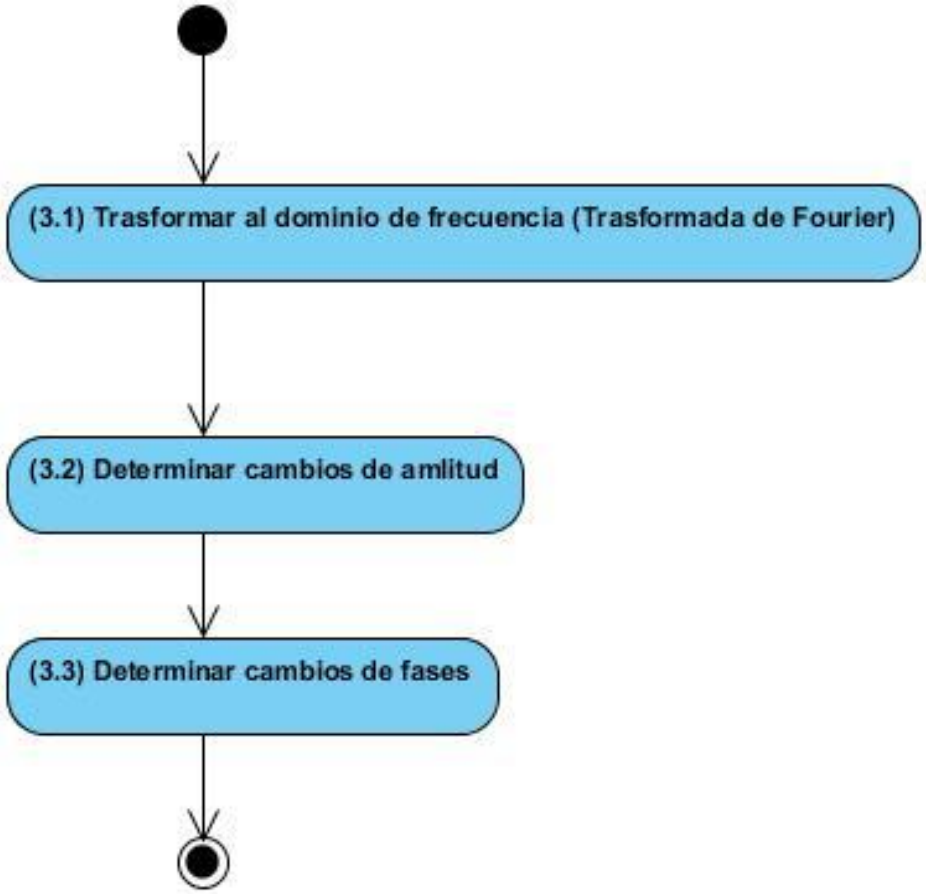


Fig 10. Descripción del paso estimación de movimiento

Paso 4. RECONSTRUCCIÓN

Para la reconstrucción se hace un registro de la imágenes con la corrección de desplazamiento y rotación de sus elementos. A estas se les aplica una operación de mediana con peso para obtener una primera aproximación de la matriz de AR (AR0). Esta es usada en el tercer paso para la realización de una convolución normalizada de primer orden, que produce una mejor estimación de la imgen de AR (AR1) y produce dos imágenes derivadas en las direcciones [x,y] (ARx y ARy). Estas imágenes derivadas son utilizadas en el cuarto paso para construir funciones de aplicabilidad

anisotrópicas para la convolución normalizada adaptativa final. El realce finalmente reduce el ruido y los efectos borrosos causados por el proceso de captura de la imagen.

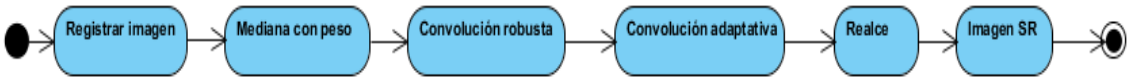


Fig 11. Convolución Normalizada Adaptativa

Paso 5. IMAGEN DE AR

Después de aplicado el método de reconstrucción mediante una CNA se interpolan los canales de color cb y cr del espacio de color ycbcr en función del factor de interpolación deseado, quedando las matrices de color del mismo tamaño que la matriz de intensidades.

Paso 6. POST-PROCESADO DE LA IMAGEN DE AR

En este paso se reconstruye la imagen a color a partir de los vectores de color cb y cr del espacio de color ycbcr, y la matriz de intensidades resultante, también se aplica una ecualización de histograma adaptativa para obtener un mejor contraste.

Paso 7. MOSTRAR RESULTADOS

Por último se muestra la imagen de AR reconstruida y sus dimensiones y se guarda en la dirección especificada por el usuario, y en caso de que este no halla especificado ninguna se guarda en el mismo directorio de donde se cargó la secuencia de imágenes.

2.2 Fase de planeación

La fase de planeación es la etapa inicial del desarrollo de software de la metodología XP. En este punto comienza a interactuar con el cliente para identificar cuáles son las historias de usuario (HU). Donde se definen el número y tamaño de las historias de usuario, en donde se plantean los ajustes necesarios a la metodología según las características del proyecto y el cliente definen el nivel de prioridad de las historias de usuario, como el tiempo, el esfuerzo que conllevarán su desarrollo.

2.2.1 Especificación de los requisitos

La ingeniería de requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente desea y cómo interactuarán los usuarios finales.

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema de software es llamado Ingeniería de Requerimientos. La meta de esta es entregar una especificación de requerimientos de software correcta y completa. La misma apunta a mejorar la forma en que se comprenden y definen sistemas de software complejos [41], trata los principios, métodos, técnicas

y herramientas que permiten descubrir, documentar y mantener los requisitos para sistemas basados en computadora de forma sistemática y repetible [42].

Requisitos funcionales (RF): son declaraciones de las funcionalidades que debe proporcionar el sistema. Definen la manera en que el software debe reaccionar a determinadas entradas. Especifican cómo debe comportarse el sistema en situaciones particulares. Pueden declarar explícitamente lo que el sistema no debe hacer [43]. Los requisitos funcionales de la aplicación son los siguientes:

- **RF.1** Cargar imágenes de baja resolución
- **RF.2** Filtrado
- **RF.3** Detección de rotación
- **RF.4** Detección de desplazamiento
- **RF.5** Convolución robusta
- **RF.6** Convolución adaptativa
- **RF.7** Realce de bordes
- **RF.8** Mostrar imagen de AR

2.2.2 Descripción de historias de usuarios

Los requisitos funcionales describen lo que debe cumplir el sistema en un lenguaje técnico. Una HU es una representación de un requisito de software escrito en una o dos frases al utilizar el lenguaje común del usuario, son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos, permiten responder rápidamente a los requisitos cambiantes, es una manera simple de describir una tarea concisa que aporta valor al usuario o al negocio.

Historia de Usuario	
Número: 1	**Nombre de la Historia de Usuario:** Cargar imágenes de BR
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	**Iteración asignada:** 1
Prioridad en negocio: Alta	**Puntos estimados:** 1 semanas
Riesgo en desarrollo: medio	**Puntos reales:** 1 semanas
Programador responsable: Elizabet Peña Machado	

Descripción: la HU comienza cuando el usuario carga la imagen de baja resolución, esta operación podrá realizarla al hacer *clic* a través de su selección en el menú de la aplicación.
Observaciones: las imágenes deben ser de BR.

Historia de Usuario	
Número: 2	**Nombre de la Historia de Usuario:** Filtrado
Modificación a la Historia de Usuario: 0	
Usuario: Elizabet Peña Machado	**Iteración asignada:** 1
Prioridad en negocio: Alta	**Puntos estimados:** 3 semanas
Riesgo en desarrollo: medio	**Puntos reales:** 3 semanas
Programador responsable: Yoandy García Barrios	
Descripción: la HU comienza cuando el usuario presiona el botón “Generar imagen de AR”, estando disponibles 2 imágenes de BR cargadas al menos.	
Observaciones: el usuario debe haber introducido correctamente el valor de interpolación.	

Historia de Usuario	
Número: 3	**Nombre de la Historia de Usuario:** Detección de rotación
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	**Iteración asignada:** 1
Prioridad en negocio: Alta	**Puntos estimados:** 3 semanas
Riesgo en desarrollo: medio	**Puntos reales:** 3 semanas

Programador responsable: Yoandy García Barrios
Descripción: se calcula un promedio de la versión centrada de la Transformada de Fourier de cada una para estimar rotación de la imagen tomada como referencia con respecto a las versiones rotadas de la misma.
Observaciones: la imagen analizada debe ser la imagen de referencia

Historia de Usuario	
Número: 4	**Nombre de la Historia de Usuario:** Detección de desplazamiento
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	**Iteración asignada:** 1
Prioridad en negocio: Alta	**Puntos estimados:** 3 semanas
Riesgo en desarrollo: medio	**Puntos reales:** 3 semanas
Programador responsable: Elizabet Graciela Peña Machado	
Descripción: se buscan diferencias de fase para detectar traslación luego de estimar la rotación de la imagen de referencia.	
Observaciones: se almacena la información para ser la entrada	

Historia de Usuario	
Número: 5	**Nombre de la Historia de Usuario:** Convolución robusta
Modificación a la Historia de Usuario: 0	
Usuario: Yoandy García Barrios	**Iteración asignada:** 2
Prioridad en negocio: Alta	**Puntos estimados:** 3 semanas
Riesgo en desarrollo: medio	**Puntos reales:** 3 semanas

Programador responsable: Elizabet Peña Machado
Descripción: después de haberse estimado el movimiento y haberse guardado la información, se pasa a la convolución robusta en la cual se va reconstruyendo una matriz de AR con los pixeles y esta información guardada.
Observaciones: el valor de interpolación debe ser entrado por el usuario, por defecto es 2.

Historia de Usuario	
Número: 6	**Nombre de la Historia de Usuario:** Convolución adaptativa
Modificación a la Historia de Usuario: 0	
Usuario: Elizabet Peña Machado	**Iteración asignada:** 2
Prioridad en negocio: Alta	**Puntos estimados:** 3 semanas
Riesgo en desarrollo: medio	**Puntos reales:** 3 semanas
Programador responsable: Yoandy García Barrios	
Descripción: después de haberse efectuado la convolución robusta en una primera iteración, se pasa a una segunda para una mejor reconstrucción, se reconstruye la matriz utilizando los píxeles con un alto valor de anisotropía.	
Observaciones: se mejora la imagen de AR estimada.	

Historia de Usuario	
Número: 7	**Nombre de la Historia de Usuario:** Realce
Modificación a la Historia de Usuario: 0	
Usuario: Michel Álvarez Cancio	**Iteración asignada:** 2
Prioridad en negocio: Alta	**Puntos estimados:** 3 semanas

Riesgo en desarrollo: medio	**Puntos reales:** 3 semanas
Programador responsable: Yoandy García Barrios	
Descripción: después de haberse efectuado la convolución adaptativa, se procede a reducir los efectos borrosos causados por el proceso de captura de las imágenes de BR.	
Observaciones: se mejora el contraste y se definen mejor los bordes de la imagen de AR.	

Historia de Usuario	
Número: 8	**Nombre de la Historia de Usuario:** Mostrar resultados
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	**Iteración asignada:** 2
Prioridad en negocio: Alta	**Puntos estimados:** 1 semanas
Riesgo en desarrollo: medio	**Puntos reales:** 1 semanas
Programador responsable: Elizabet Peña Machado	
Descripción: se muestra la imagen de AR resultante de la aplicación de las técnicas de SR.	
Observaciones: la imagen debe tener mayor resolución que las imágenes de muestra.	

2.2.3 Fase de planificación de la entrega

En la fase de la planificación de entrega se definen las prioridades de cada historia de usuario y consecuentemente se realiza una estimación del esfuerzo necesario de cada una de ellas. Se definen las entregas con el cliente donde se tiene por norma que una entrega debe adecuarse a tres meses como máximo.

En la tabla 1 se muestran cada una de las historias de usuario, así como la estimación del tiempo en que se cumplirá, donde las tareas de máxima duración son: “filtrado”, “detección de rotación”,

“detección de desplazamiento”, “convolución robusta” y “convolución adaptativa”, obteniéndose una duración total estimada de 19 semanas.

Tabla 1.Estimación de tiempo por historia de usuario

No	Historias de usuario	Puntos de estimación (semanas)
1.	Cargar imágenes de BR	1
2.	Filtrado	3
3.	Detección de rotación	3
4.	Detección de desplazamiento	3
5.	Convolución robusta	3
6.	Convolución adaptativa	3
7.	Realce de bordes	2
8.	Mostrar imagen de AR	1

2.2.4 Plan de iteraciones

Una vez definidas las historias de usuarios e identificar el tiempo para la su implementación, se diseña un plan de iteraciones donde las historias de usuario están contenidas, por tanto, se pretenden realizar el desarrollo en 2 iteraciones, referidas a continuación:

Iteración 1

La iteración tiene como fin estimar movimientos, donde se cargan las imágenes de BR, las cuales deben estar contenidas en una misma escena. Posteriormente se aplica un filtrado para eliminar ruido y suavizar las imágenes, y finalmente se realiza una detección de rotación y desplazamiento.

Iteración 2

Esta iteración se encarga de aplicar la técnica CNA y se ocupa de reconstruir la imagen de AR a partir de las imágenes de BR insertadas por el usuario. En esta iteración luego de obtener la imagen reconstruida se aplican técnicas de realce de bordes para luego mostrar la imagen de AR reconstruida.

Tabla 2.Plan de duración de entrega

Iteraciones	Historias de usuario	Duración
Iteración 1	Cargar imágenes de BR	10
	Filtrado	
	Detección de rotación	
	Detección de desplazamiento	
Iteración 2	Convolución robusta	9
	Convolución adaptativa	
	Realce de bordes	
	Mostrar imagen de AR	

2.3 Diseño de la solución

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

2.3.1 Tarjetas CRC

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) son utilizadas para representar las responsabilidades de las clases y sus interacciones. Estas, permiten trabajar con una metodología basada en objetos [44].

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 aplican técnicas de grupos como la tormenta de ideas o juego de roles y se ejecutan escenarios a partir la de especificación de requisitos, historias de usuarios o casos de uso. De esta forma, van surgiendo las entidades del sistema junto con sus responsabilidades y colaboraciones [45].

Tabla 3. Tarjeta CRC de la clase "estimation"

Clase: estimation

Responsabilidades	Colaboradores
Permite estimar la rotación en la secuencia de imágenes de BR.	cart2pol, fft2, fftshift, sort,floor
Corrige la rotación entre las imágenes de BR	imrotate
Permite estimar el desplazamiento en la secuencia de imágenes de BR.	fft2, fftshift, angle

2.3.2 Estándar de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada [46]. Para la realización del presente trabajo se establecen tres estándares de codificación:

Variable:

- Los nombres de las variables deben ser cortos y significativos.
- La elección de un nombre de variable debe ser mnemotécnica, esto es, diseñado para demostrar el propósito de su uso a cualquier observador.

Constantes:

- Los nombres de variables declaradas como constante deben ser todas en mayúsculas con palabras separadas por guión abajo (“_”).
- Las constantes ANSI deben ser evitadas para facilidad de la Depuración.
- Se debe seguir las mismas convenciones que se usan para variables con respecto a los prefijos para tipo de dato.

Funciones:

- Los nombres de los métodos deben empezar con una letra mayúscula y el resto de letras deben estar escritas en minúscula.
- Los nombres de los métodos deben ser verbos o palabras que identifiquen de manera general el objetivo del método
- Los nombres de los métodos no pueden contener espacios ni caracteres especiales, sólo son permitidas las letras de la “a” a la “z” y los números del 0 al 9.
- Si el nombre de método requiere estar compuesto por más de una palabra, cada palabra adicional debe empezar con mayúscula.

2.3.3 Patrón de diseño

En la presente investigación se utiliza el patrón “*singleton*”, el cual asegura que exista una única instancia de una clase (ver fig 12). A primera vista, uno puede pensar que pueden utilizarse clases con miembros estáticos para el mismo fin. Sin embargo, los resultados no son los mismos, ya que en este caso la responsabilidad de tener una única instancia recae en el cliente de la clase. El patrón hace que la clase sea responsable de su única instancia, quitando así este problema a los clientes. Adicionalmente, si todos los métodos de esta clase son estáticos, éstos no pueden ser extendidos, desaprovechando así las capacidades polimórficas que nos proveen los entornos orientados a objetos. Este patrón se utiliza para la realización de la interfaz de usuario del sistema.

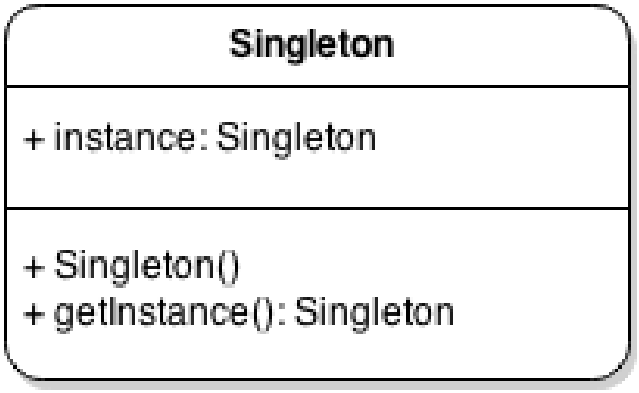


Fig 12. Diagrama de definición de patrón Singleton

2.4 Conclusiones del capítulo

En este capítulo fueron descritos los elementos necesarios para el desarrollo de la solución propuesta al problema de la investigación. La especificación de los requisitos funcionales, permitió definir el comportamiento y las características que el producto debe garantizar.

Con desarrollo de la metodología XP se generan los artefactos correspondientes, como las historias de usuario, las iteraciones donde se van a ejecutar cada una, y el plan de entrega, generando un total de 8 HU, que describen las funcionalidades que se ejecutan. Se definen diferentes estándares

de codificación, para garantizar la uniformidad del código permitiendo un diseño estándar y organizado que será de fácil comprensión para otros desarrolladores y se selecciona el patrón "*singleton*".

Capítulo 3: Resultados y validación

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. En el siguiente capítulo se exponen los resultados obtenidos mediante el detector de bordes Canny y luego se valida el sistema utilizando las pruebas de caja negra y caja blanca con las técnicas partición de equivalencia y camino básico respectivamente.

3.1 Interfaz del sistema

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

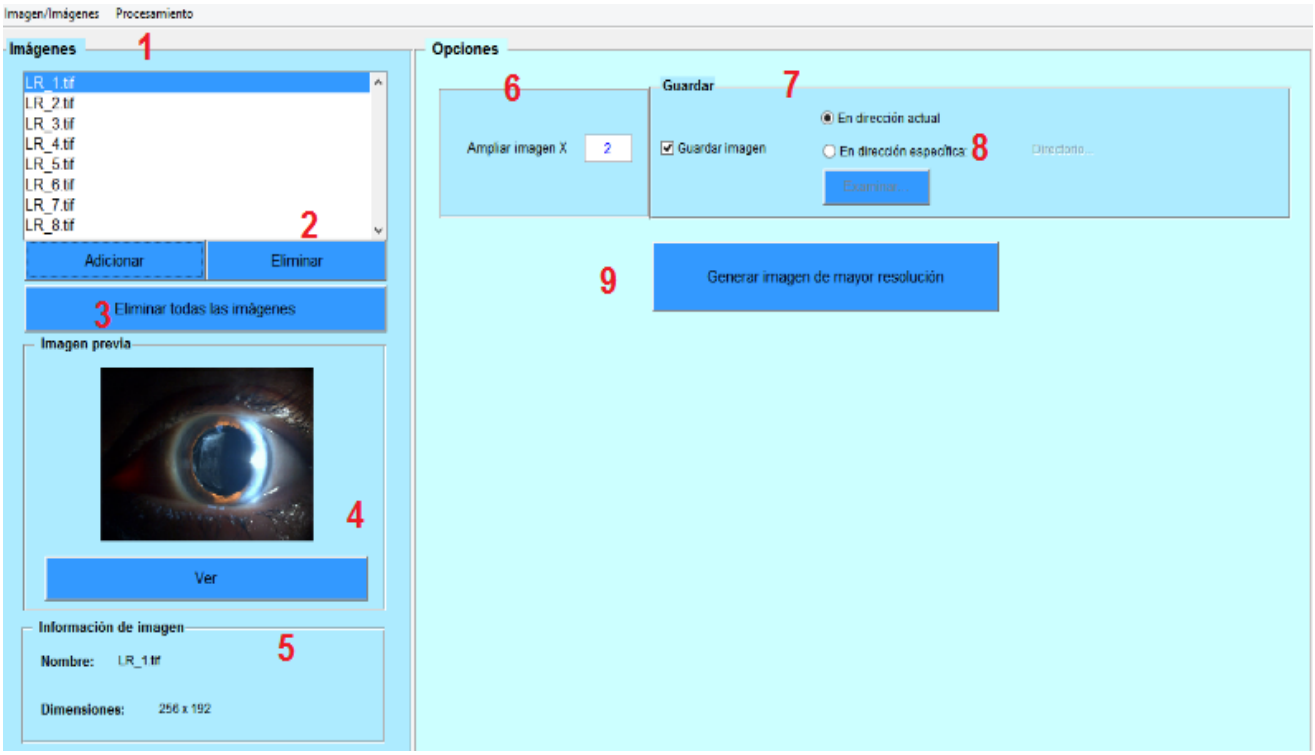


Fig 13. Interfaz para la aplicación del algoritmo de SR propuesto

- 1: Permite cargar las imágenes de BR.
- 2: Permite eliminar 1 imagen cargada.
- 3: Permite eliminar todas las imágenes que han sido cargadas por el usuario.
- 4: Permite tener 1 imagen previa de la imagen de referencia.
- 5: Muestra detalles de la imagen como nombre y dimensiones.
- 6: Permite ingresar el valor de interpolación de la imagen.
- 7: Permite guardar la imagen en la dirección donde ha sido cargada.
- 8: Permite guardar la imagen en una ruta específica indicada por el usuario.

- 9: Permite generar la imagen de AR resultante.

Resultados de aplicar el algoritmo

El algoritmo diseñado se le aplicó a 7 secuencia de imágenes de BR provenientes de la lámpara de hendidura. A continuación se muestra un ejemplo luego de aplicar el algoritmo propuesto, en la fif 15 b y d, muestra y resultado respectivamente, donde aparecen dos fragmentos de imágenes luego de aplicar zoom, para una mejor perspectiva visual.

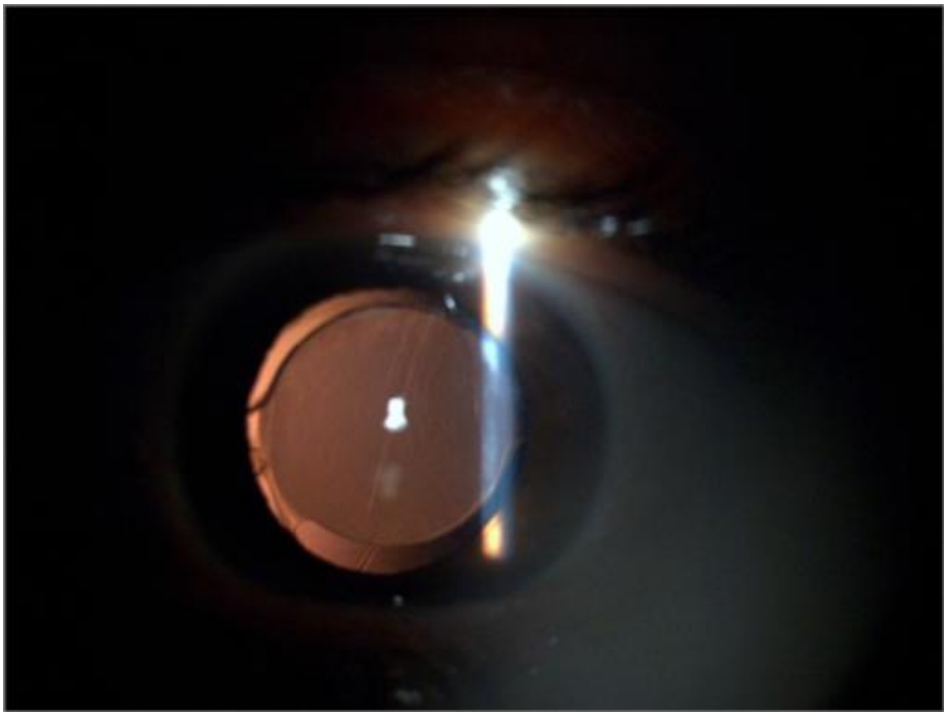


Fig 14. (a)

Fig 15. (b)

Fig 16. (c)

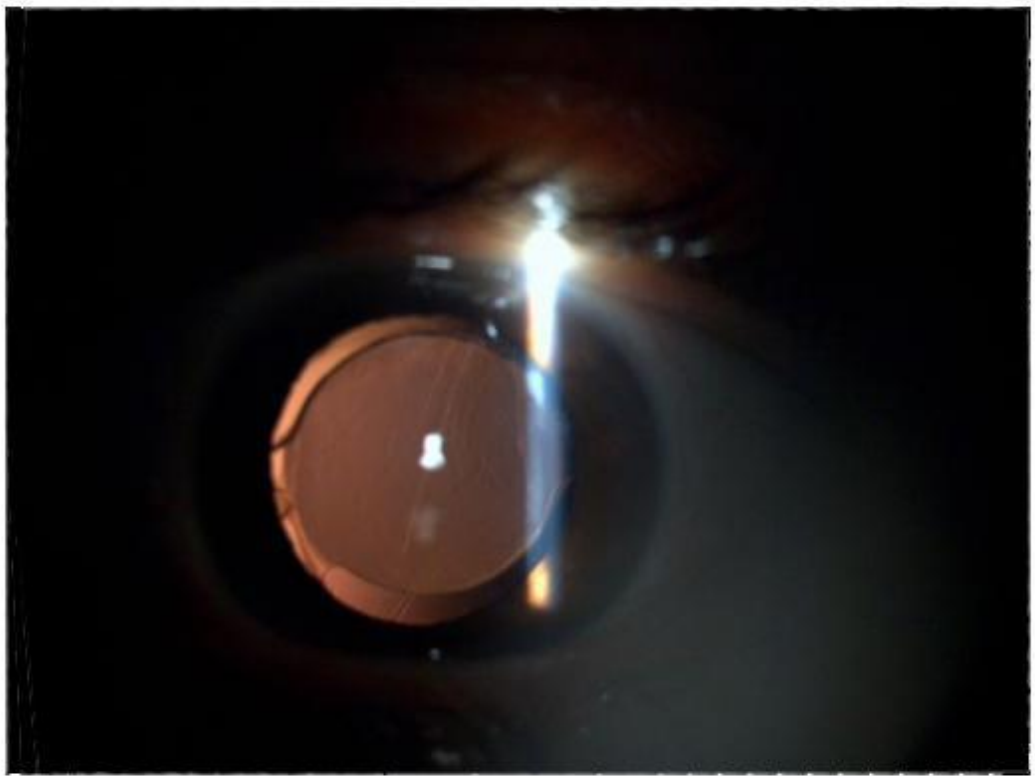


Fig 17. (d)

- Imagen original (a)
- Imagen original aumentada (b)
- Imagen resultante (c)
- Imagen resultante aumentada (d)

3.2 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 [47]. Su detección resulta útil en diversas disciplinas, como cartografía o medicina, cuando se pretende preservar importantes propiedades estructurales o cuando se plantea la necesidad de distinguir formas o reconocer Figuras dentro de una imagen [48]. Unos de los métodos más importantes para realizar una detección global de bordes sobre una imagen es el conocido como método de Canny [49]. Esta técnica, que se caracteriza por estar optimizada para la detección de bordes diferenciales. Consta de tres pasos a seguir:

1. Cálculo del gradiente: primero 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 podrían perder detalles de la imagen y provocar un pésimo resultado final. Este suavizado se obtiene promediando los valores de intensidad de los píxeles en el entorno de vecindad con una máscara de convolución de media cero y desviación estándar σ. Una vez que se suaviza la imagen, para cada píxel 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 normal 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 los bordes adelgazados. El procedimiento es el siguiente: se consideran cuatro direcciones identificadas por las orientaciones de 0º, 45º, 90ºy 135º con respecto al eje horizontal. Para cada píxel se encuentra la dirección que mejor se aproxime a la dirección del ángulo de gradiente.

Algoritmo: Obtención de Gradiente
Entrada: imagen I
máscara de Convolución H, con media cero y desviación estándar σ.
Salida: imagen Em, de la magnitud del gradiente
Imagen Eo de la orientación del gradiente1.

- Suavizar la imagen I con H mediante un filtro gaussiano y obtener J como imagen de salida.
- Para cada píxel (i, j) en J, obtener la magnitud y orientación del gradiente basándose en las siguientes expresiones:

El gradiente de una imagen f(x,y) en un punto (x,y) se define como un vector bidimensional dado por la ecuación:

2.18 $$\boldsymbol{G}[\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial_x} f(x,y) \\ \frac{\partial}{\partial_y} f(x,y) \end{bmatrix}$$

siendo un vector perpendicular al borde, donde el vector G apunta en la dirección de variación máxima de f en el punto (x,y) por unidad de distancia, con la magnitud y dirección dadas por:

2.19 $$|\boldsymbol{G}| = \sqrt{G_x^2 + G_y^2} = |G_x| + |G_y|$$

$$\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y}) = tan^{-1}\frac{G_y}{G_x}$$

- Obtener Em a partir de la magnitud de gradiente y Eo a partir de la orientación, de acuerdo a las expresiones anteriores.

Posteriormente se observa si el valor de la magnitud de gradiente es más pequeño que al menos uno de sus dos vecinos en la dirección del ángulo obtenida en el paso anterior. De ser así se asigna

el valor 0 a dicho píxel, en caso contrario se asigna el valor que tenga la magnitud del gradiente [50].

Algoritmo: Supresión no máxima

Entrada: imagen Em de la magnitud del gradiente

Imagen Eo de la orientación del gradiente

Salida: imagen In

Considerar: cuatro direcciones d1, d2, d3, d4, identificadas por las direcciones de 0º, 45º, 90º y 135º con respecto al eje horizontal.

1. Para cada píxel (i, j)
   - Encontrar la dirección dk que mejor se aproxima a la dirección Eo (i, j), que viene a ser la perpendicular al borde.
   - 1.2. Si Em (i, j) es más pequeño que al menos uno de sus dos vecinos en la dirección dk, al píxel (i, j) de In se le asigna el valor 0, In (i, j)= 0, (suspensión), de otro modo In (i, j) = Em (i, j).
2. Devolver In

3 Histéresis. En este paso se definen los bordes definitivos, seleccionando sólo 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 a 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 ello conlleva un aumento del error en su posición [48].

Algoritmo: Histéresis de umbral a la supresión no máxima

Entrada: imagen In obtenida del paso anterior

Imagen Eo de la orientación del gradiente

umbralt1

umbralt2, donde t1< t2

Salida: imagen G con los bordes conectados de contornos

1. Para todos los puntos de In y explorando In en orden fijo:
   - Localizar el siguiente punto de borde no explorado previamente, In (i, j), tal que In (i, j) > t2.
   - Comenzar a partir de In (i, j), seguir las cadenas de máximos locales conectados en ambas direcciones perpendiculares a la normal de borde, siempre que In (i, j) > t1.
   - Marcar todos los puntos explorados y, salvar la lista de todos los puntos en el entorno conectado encontrado.

2. Devolver G formada por el conjunto de bordes conectados de contornos de la imagen, así como la magnitud y orientación, describiendo las propiedades de los puntos de borde.

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 [51].

El resultado muestra como abrupta o suavemente cambia una imagen en cada punto analizado y en consecuencia, cuán probable es que esté represente un borde en la imagen, como la orientación a la que tiende ese borde [52].

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 [51].

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 las siguientes: G x se aplica en la dirección x, que corresponde a las columnas de la imagen, mientras que G y se aplica dirección en la que avanzan las filas.

$$G_X = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_Y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Fig 18. Máscara binomial de 3x3

Para la validación mediante el detector de bordes Canny se tomó una muestra de 8 imágenes de BR y su versión restaurada mediante el algoritmo propuesto. Teniendo ambas con las mismas dimensiones con ayuda de la función **imresize** de Matlab, se procede a aplicar el detector de bordes Canny y contabilizar la cantidad de bordes en cada una de las imágenes (la de AR y la de BR).

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 borde de 3932, y la segunda es el resultado de aplicar el algoritmo de superresolución con una cantidad de bordes de 12837. Este cuenta con una diferencia de 8905 bordes. Por tanto, teniendo en cuenta 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). Por lo que se obtienen resultados satisfactorios.

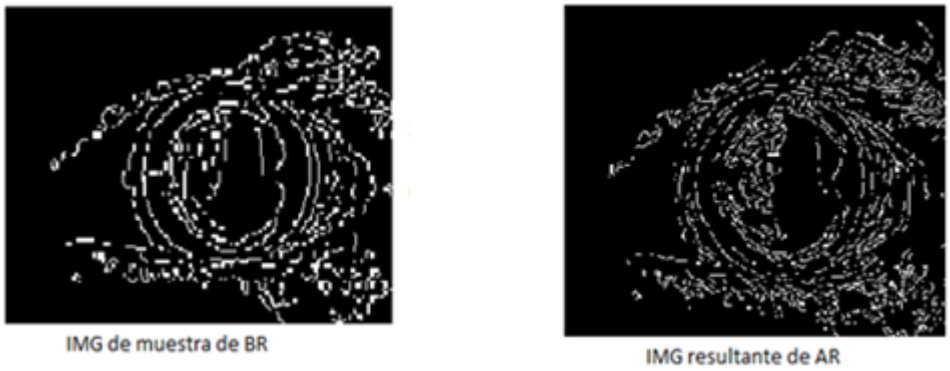


Fig 19. Resultados de la cuantificación de los bordes al aplicar Canny

Tabla 4. Resultados de aplicar Canny

Imágenes	Porcientos de píxeles blancos
Imagen Original	4.48%
Imagen Preprocesada	6.53%

3.3 Verificación del sistema

Los errores sencillos de un sistema que suelen estar ocultos, con el trascurso 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 en cuenta la aplicación de las pruebas de software en 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, se aplican diferentes métodos y técnicas. En este capítulo se expone la verificación del sistema.

3.3.1 Estrategia de pruebas

En el libro "Ingeniería del Software" de Roger S. Pressman, se define que, “una estrategia de prueba del software integra los métodos de diseño de caso de pruebas en una serie bien planeada de pasos que desembocará en la eficaz construcción de software. La estrategia proporciona un mapa que describen los pasos que se darán como parte de la prueba, indica cuándo se planean y cuándo se dan es dos pasos, además de cuánto tiempo, esfuerzo y recursos consumirán. Por tanto, cualquier estrategia de prueba debe incorporar la planeación de pruebas, el diseño de casos de pruebas, la ejecución de las pruebas y la recolección y evaluación de los resultados” [53].

Objetivos de las Pruebas

Las pruebas presentan una interesante anomalía para el ingeniero del software. Durante las fases anteriores de definición y de 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 prueba que intentan demoler el software construido [53]. Con el objetivo de:

- Detectar cualquier tipo de error 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 iniciada 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 fallos del producto.

3.3.2 Métodos de pruebas

Los métodos de la ingeniería de software indican cómo construir técnicamente el software. Los métodos abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento. Estos métodos dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas**.**

3.2.2.1 Pruebas de caja blanca

Enfoca todo análisis y estudio en el código fuente del sistema, en este se realizan casos de prueba que se ejercitan lo menos una vez todos los caminos independientes de cada método, mediante la técnica de camino básico.

```
function [delta_est, phi_est] = estimation(s,r_max,d_max)
if (nargin==1)
    r_max = 0.8;
    d_max = 8;
end

% estimo la rotación
[phi_est, c_est] = estimate_rotation(s,[0.1 r_max],0.1);

% ajusto la rotación según estimación anterior
s2{1} = s{1};
nr=length(s);
for i=2:nr
    s2{i} = imrotate(s{i},-phi_est(i),'bicubic','crop');
end

% estimo desplazamiento
delta_est = estimate_shift(s2,d_max);
```

Fig 20. Código fuente del método “estimation”

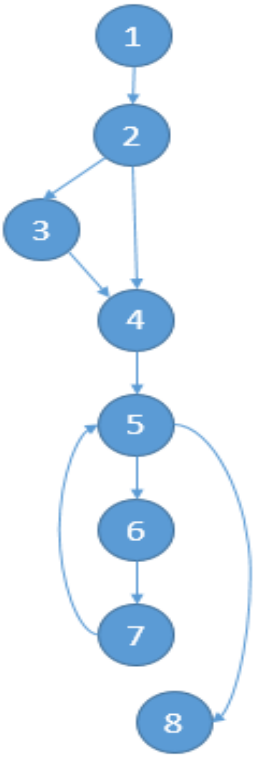


Fig 21. Grafo de flujo asociado al método “estimation”

Teniendo en cuenta el diseño del grafo como se muestra en la Fig.16, se procede a calcular la complejidad ciclomática, para determinar el número de caminos independientes y ofrece un límite inferior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez. El valor V (G) expresa la cantidad de caminos linealmente independientes de la estructura de control del programa.

Complejidad ciclomática

V (G) = Regiones

V (G) = 3

V (G) = Aristas (A) – Nodos (N) + 2

V (G)= 9-8+ 2

V (G)= 3

V (G) = Nodos Predicado + 1

V (G) = 2 + 1

V (G)= 3

Camino 1: 1->2->3->4->5->6->7->5->8

Camino 2: 1->2->4->5->6->7->5->8

Camino 3: 1->2->4->5->8

Cada camino independiente es un caso de prueba a realizar, de forma que los datos señalados causen que se visiten las sentencias vinculadas a cada nodo del camino. En el caso anterior se calcularon tres caminos básicos, por tanto surge la necesidad de hacer igual número de casos de prueba, para aplicar las pruebas a este método.

Tabla 5.Caso de prueba del camino básico aplicado al método estimation

Entrada	Conjunto de imágenes de baja resolución, ángulo máximo de rotación y máxima variación de desplazamiento
Resultados esperados	Conjunto de imágenes de baja resolución con rotación y desplazamiento corregido.
Condiciones	El conjunto de imágenes de entrada deben ser de la misma escena.

3.2.2.2 Pruebas de caja negra

Método que permite centrar la atención en generar casos de prueba que posibiliten ejercitar propuestas de los requisitos funcionales de un programa. Encargada de la verificación del correcto funcionamiento de la interfaz del sistema. En el proceso de validación del mismo se le aplicó la técnica de partición de equivalencia.

Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las Historias de Usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario 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 [53]. Por lo tanto, son los clientes los responsables de 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 los fallos. Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información [53].

Tabla 6.Caso de prueba 1

Caso de prueba de aceptación
Código: HU_I1 | Historia de Usuario:1
Nombre: Cargar imagen de BR
Descripción: Registra un conjunto de imágenes de BR de la misma escena.
Condiciones de Ejecución: Las imágenes deben estar en formato **tif**.
Entrada/Pasos de Ejecución: • El usuario selecciona las imágenes. • El sistema comprueba que estén en formato **tif** y presenten el metadato de “SamplesPixel” que brinda la cantidad de muestras por píxel. • Muestra en un panel las imágenes de BR cargadas.
Resultado Esperado: Se registran y se muestran en un panel las imágenes de BR para ser procesadas
Evaluación de la Prueba: Prueba satisfactoria.

Caso de prueba de aceptación
Código: HU_I2 | Historia de Usuario:2
Nombre: Filtrado
Descripción: Se realiza para desechar los pixeles no deseados por contener ruido.
Condiciones de Ejecución:

Entrada/Pasos de Ejecución: • El sistema comprueba que el píxel no sea ruidoso para tomarlo en cuenta en el registro posterior para su adición en las matrices a construir.
Resultado Esperado: Se generan las matrices correspondientes a cada etapa con los pixeles sin ruido.
Evaluación de la Prueba: Prueba satisfactoria.

Caso de prueba de aceptación
Código: HU_I3 | Historia de Usuario:3
Nombre: Estimación de rotación
Descripción: Estima los cambios de rotación de la imagen rotada con respecto a la imagen de referencia almacenando esta información para su empleo posterior en la corrección.
Condiciones de Ejecución: Las imágenes deben ser de la misma escena y contener movimiento.
Entrada/Pasos de Ejecución: • El sistema analiza la imagen rotada y la de referencia detectando el ángulo de rotación para su corrección y posterior reconstrucción.
Resultado Esperado: Se registran los cambios de rotación y se efectúa la corrección de la imagen.
Evaluación de la Prueba: Prueba satisfactoria.

Caso de prueba de aceptación
Código: HU_I4 | Historia de Usuario:4
Nombre: Estimación de desplazamiento
Descripción: Estima los cambios de traslación de la imagen trasladada con respecto a la imagen de referencia almacenando esta información para su empleo posterior en la corrección.
Condiciones de Ejecución: Las imágenes deben ser de la misma escena y contener movimiento.
Entrada/Pasos de Ejecución: • El sistema analiza la imagen desplazada y la de referencia detectando el desplazamiento para su corrección y posterior reconstrucción.
Resultado Esperado: Se registran los cambios de traslación y se efectúa la corrección de la imagen.
Evaluación de la Prueba: Prueba satisfactoria.

Caso de prueba de aceptación
Código: HU_I5 | Historia de Usuario:5
Nombre: Convolución robusta
Descripción: Se realiza para la reconstrucción de la imagen de alta resolución con el tratamiento de los pixeles ruidosos, los cuales son descartados en este proceso.
Condiciones de Ejecución: Las matrices de intensidades deben estar creadas.

Entrada/Pasos de Ejecución: • Analiza pixel por pixel, ignorándolo en caso de que contenga ruido, para la construcción de la matriz de intensidad de AR
Resultado Esperado: Matriz de intensidad de AR como entrada para un realce de bordes.
Evaluación de la Prueba: Prueba satisfactoria.

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

No. de iteración	No conformidades	Recomendaciones	No conformidades tratadas
1	4	3	4
2	0	0	0

Fig 22. Resultados de las pruebas de aceptación

3.3.3 Conclusiones del capítulo

En los avances del capítulo se abordó la implementación de la solución propuesta, así como la validación de los resultados. Se llevaron a cabo las pruebas realizadas a la solución, mostrando resultados satisfactorios, lo que demuestra un papel fundamental en el proceso de desarrollo de la utilización de una metodología ágil, en este caso XP. Se realizó además la validación del sistema la cual permitió demostrar que el algoritmo satisface cumple con el objetivo de la investigación.

Conclusiones

Una vez cumplidas las tareas de la investigación definidas para dar solución al problema inicialmente plateado, se arriba a las siguientes conclusiones:

- Se utilizaron técnicas de SR para mejorar la calidad de las imágenes oblicuas provenientes de la lámpara de hendidura, donde las técnicas utilizadas para ello fueron EM y R, mediante los algoritmos Vandewalle y CNA, obteniendo resultados satisfactorios.
- Se validaron los resultados mediante el detector de borde Canny, para comprobar la calidad de la imagen obtenida. Para medir dicho indicador de calidad, la imagen resultante debe tener mayor número de bordes que la imagen de referencia para lograr buenos resultados en las imágenes oblicuas.
- Se comprobó que el algoritmo propuesto cumple con los requerimientos especificados, ya que permite obtener una imagen de AR, de mayores dimensiones que las originales.
- 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 se podrá usar en aras de una mejor precisión al emitir un diagnóstico de la opacidad de la cápsula posterior.

Recomendaciones

- Se recomienda la implementación y prueba de otros algoritmos de estimación de movimiento y de reconstrucción y comparar los resultados con los propuestos.
- Ver la factibilidad (tiempo de ejecución vs resultados) del uso de algoritmos de superresolución basados en subpíxel, los cuales usan una sola imagen de referencia.

Referencias Bibliográficas

1. Boyd, K., *What Are Cataracts? .* American Academic of Ophthalmology, 2014.
2. Acosta R, H.L., Roman R, Comas M, Castilla M, Castells X, Revisión sistemática de estudios poblacionales de prevalencia de cataratas, 2006. **81**: p. 509-516.
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 Alvarez Cancio, E.P.V., Rafael Rodríguez Puentes, SilenaHerold Garcia, *Algorithm for the identification of the posterior capsule opacity in images from PENTACAM.* Revista Cubana de Ciencias Informáticas 2017. **11**: p. 153-167.
6. Alvarez de Zayas, C., *Metodología de la investigación científica*. 1995, Centro de estudios de Educación Superior "Manuel F. Gran"", Santiago de Cuba.
7. 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
8. Michel Alvarez Cancio, A.H.B., Rafael Rodríguez Puentes Y Iván Hernández López, *PANDOC: software para la cuantificación objetiva de la opacidad de la cápsula posterior mediante tomogramas scheimpflug del pentacam.* 2013.
9. Raciel Cepero Ruz, C.C.R., Implementación de algoritmos de reducción de ruido en las imágenes, 2010.
10. Calhoun, N.; Available from: www.ehowenespanol.com/considera-imagen-resolucion-info_262419/.
11. Pascual, D.R. *Proyecto Ocularis*. Available from: http://ocularis.es/blog/41/.
12. V. SRINIVASAN, R.D.T., *Instrumentos y Equipos Oftalmológicos.* 2003.
13. S. C. Park, M.K.P., and M. G. Kang, *Super-resolution Image Reconstruction: a Technical Overview.* IEEE Signal Processing Magazine, 2003. **20**: p. 21-36.
14. Delrieux, P.S.H.y.D.C., *Superresolucion: Reconstruccion de datos en 1D.* 2013.
15. S. Farsiu, D.R., M. Elad, and P. Milanfar Advances and Challenges, 2004.
16. L. D. Alvarez, R.M., and A.K. Katsaggelos *High Resolution Images from a Sequence of Low Resolution Observations.* Digital Image Sequence Processing, Compression and Analysis, 2004. **9**: p. 233-259.
17. R. R.Schultz, L.M., and R. L.Stevenson *Subpixel Motion Estimation for Super-Resolution Image Sequence Enhancement,.* J. Visual Communication and Image Representation, 1998. **9**: p. 38-50.
18. Chaudhuri, S., *Super-Resolution Imaging*. 2002, KLUWER ACADEMIC PUBLISHERS.
19. Milanfar, P., *Super-resolution imaging*. 2010, CRC Press.
20. Andrews, H.C.a.H., Bobby Ray, *Digital image restoration*. Prentice-Hall signal processing series. 1977: Englewood Cliffs, N.J. : Prentice-Hall, c1977. .
21. Katsaggelos, A.K., *Digital Image Restoration*. 1991, Springer-Verlag New York, Inc. Secaucus, NJ, USA ©1991.
22. Patrick Vandewalle, S.S., and Martin Vetterli. A Frequency Domain, *Approach to Registration of Aliased Images with Application to Super-Resolution.* EURASIP Journal on Applied Signal Processing (special issue on Super-resolution), 2006: p. 14.
23. D Keren, S.P., and R Brada, *Image sequence enhancement using sub-pixel displacement.* In Proceedings IEEE Conference on Computer Vision and Pattern Recognition, 1988: p. 742–746.
24. Peleg, M.I.a.S., *Improving resolution by image registration.* Graphical Models and Image Processing, 1991: p. 231-239.
25. Flusser, B.Z.a.a.J., *Image registration methods: a survey.* Image and Vision Computing Journal, 2003: p. 977-1000.
26. Richards, J.A., *Remote Sensing Digital Image Analysis*. 1999.
27. Papoulis, A., *A new method in image restoration.* JSTAC, 1973.
28. *Scrum methodology.* 1999.
29. Jeffries, R., *What is extreme programming.* 1999.

30. Tuan Q. Pham, L.J.v.V., and SchutteKlamer, *Robust Fusion of Irregularly Sampled Data Using Adaptive Normalized Convolution.* EURASIP Journalon Applied Signal Processing, 2006.
31. Herrera-Pereda, R., *Sobre la superresolución como método de preprocesamiento de imágenes digitales confocales de la córnea*, in *Facultad de Matemática Física y Computación*. 2016, Universidad Central "Marta Abreu" de Las Villas.
32. VILLANUEVA, M.V.S., Manuel Berenguel y CANTARERO, Teodoro Álamo, *Tutorial de Introducción a MATLAB.* 2012.
33. Lidia Fuentes, A.V., *Una introducción a los perfiles UML.* 2004.
34. Visconti, M.a.A., Hernán, *Fundamentos de ingeniería de software.*
35. *VISUAL PARADIGM. UML tool*. Available from: http://www.visual-paradigm.com.
36. Julio Benítez López, J.L.H.P., Introducción a MATLAB.
37. Sánchez Mendoza, M.A., *Metodologías del desarrollo de software.* 2004.
38. Juristo, N., Moreno, A. M., & Vegas, S. , *Técnicas de evaluación de software.* 2006.
39. Joskowicz, J., *Reglas y prácticas en eXtreme Programming.* 2008.
40. Rodriguez Bustamante, D., *Metodología Actual:Metodología XP.* 2014.
41. Grau, R., Lauenroth, K., Requirements engineering and agile development.
42. Méndez, G., *Ingeniería de requisitos.* Universidad Complutense de Madrid, 2008.
43. Sommerville, I., *Ingeniería del Software*. 2005: España: PEARSON EDUCACIÓN.
44. BECK, K., *Extreme Programming Explained.* 1999.
45. Casas, S., & Reinaga, H., *Aspectos tempranos: un enfoque basado en Tarjetas CRC.* 2009.
46. *Estándares de Codificación*. Available from: https://docs.google.com/document/d/1rbxDFM0zsbFDNRZeM2FoXfRDbYSiSt6tCdbYPA0qdzs/edit?hl=en_US#bookmark=id.6e6a203b40fe.
47. Miguel A. Jaramillo, J.Á.F.a.E.M.d.S. *Implementación del Detector de Bordes de Canny sobre Redes Neuronales Celulares*.
48. T. Hermosilla, E.B., A. Balaguer and L.A. Ruiz, *Detección de bordes con precisión subpíxel en imágenes digitales: Interpolación lineal frente a esquemas de tipo no lineal.* 2006.
49. Canny, J.F., *A Computational Approach to Edge Detection.* 1986: p. 679-698.
50. Valverde-Rebaza, J. *Detección de bordes mediante el algoritmo de Canny*. 2007.
51. Coto, E., *Métodos de Segmentación de imágenes Médicas.* 2005.
52. Saleh Alamri, M.S., Kalyankar, D. N., & Khamitkar, S. , *Image segmentation by using edge detection.* . 2010. **2**: p. 804-807.
53. Roger, P., *Ingeniería de software*. 6ta edición ed. 2011.