



Universidad de las Ciencias Informáticas

AIRI, Artificial Intelligence: Research and Innovation, Facultad 3

Algoritmo para la detección y realce de bordes
en imágenes oblicuas provenientes de las
lámparas de hendidura.

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

Autor:

Jeidy Rosell Aldana

Tutores:

Ing. Michel Álvarez Cancio

Lic. Reyder Cruz de la Osa

La Habana. Junio, 2016.

Declaratoria de autoría

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Jeidy Rosell Aldana

Autor

Ing. Michel Álvarez Cancio

Tutor

Lic. Reyder de la Cruz Osa

Tutor

Agradecimientos

A toda mi familia porque son mi mayor tesoro y sostén.

Ante todo quiero agradecer a mis padres, por su dedicación constante, amor incondicional y apoyo.

A mi madre por ser mi amiga y estar pendiente de cada paso que doy, por su cariño incondicional, por levantarme cuando no podía hacerlo yo sola.

A mi papá por apoyarme, por su paciencia, y porque siempre confió en mí y está a mi lado siempre que lo necesito.

A mis abuelos, en especial por su incansable preocupación.

A mi tía, mi primo que siempre me han dado mucho amor y cariño.

A los amigos de toda la vida, a los que están lejos y a los de siempre que nunca fallan en los momentos difíciles.

A los amigos que conocí durante estos 5 años, gracias por su apoyo.

A Luis Ángel (chino) por brindarme siempre su apoyo en el momento exacto.

A todas las personas que me ayudaron, en especial a Osvaldo que me ayudó en todo lo que necesité incondicionalmente.

A los profesores que marcaron cada etapa de mi aprendizaje.

En fin, a todos los que contribuyeron a mi formación como profesional, a los que hicieron que el camino fuera más fácil y siempre confiaron en que podía llegar.

Jeidy Rosell Aldana.

Dedicatoria

A mis padres por ser mi inspiración y brindarme todo el apoyo, por estar para enseñarme y educarme, por brindarme la fuerza cuando he sentido que no puedo levantarme sola y mostrarme que hay que sacrificarse cuando se quiere lograr algo.

Contenido

Declaratoria de autoría	II
Agradecimientos	III
Dedicatoria	IV
Resumen	VIII
Introducción	5
Capítulo 1 Fundamentación Teórica	9
1.1 Imagen Digital.....	9
1.1 Imágenes a Color.....	10
1.1.2 Imagen Oblicua.....	11
1.2 Opacidad de la Cápsula Posterior.	12
1.3 Procesamiento de Imágenes.	13
1.3.1 Preprocesamiento de imágenes.....	14
1.3.2 Segmentación de imágenes.....	23
1.4 Herramientas usadas durante el desarrollo	28
1.4.1 Lenguajes de Programación	28
1.4.2 Entorno integrado de desarrollo	29
1.4.3 Herramienta de prueba	29
1.4.4 Herramienta para el modelado.....	30
1.4.5 Programa Analizador de Opacidad Capsular: PANDOC	30
1.4.6 Metodología de desarrollo.....	31
1.4.7 Patrones de diseño de software	32
1.5 Conclusiones del capítulo.....	33
Capítulo 2. Propuesta de solución	34
2.1 Descripción de la selección de las técnicas.....	34
2.2 Descripción de la solución	37
2.3 Fase de Planeación.....	39
2.3.1 Historias de Usuario.....	39
2.3.2 Fase de Planificación de la Entrega	40
2.3.3 Requisitos No Funcionales	41
2.3.4 Plan de iteraciones	41
2.3.5 Plan de entrega	42
2.4 Diseño de la solución	43
2.4.1 Tarjetas CRC	43
2.4.2 Patrones de diseño utilizados	44
2.4.3 Estándares de codificación	45
2.4 Conclusiones del capítulo.....	45

Capítulo 3. Resultados y validación del sistema	47
3.1 Fase de Implementación	47
3.2 Resultados de aplicar el algoritmo	51
3.3 Pruebas.....	52
3.4 Prueba de Caja Blanca.....	53
3.5 Pruebas de Caja Negra	54
3.5.2 Pruebas de Aceptación	55
3.6 Validación de la solución	58
3.6.1 Validación con Operador Canny	59
3.6.2 Métrica de reducción de ruido propuesta por Xiang Zhu	60
3.6.3 Métrica basada en Diagramas de dispersión	61
3.7 Conclusiones del capítulo.....	62
Conclusiones generales.....	64
Recomendaciones	65
Referencias	66
Glosario de Términos.....	72

Ilustración 1. Espectro Electro-Magnético	9
Ilustración 2. Imagen Digital	10
Ilustración 3. Modelo RGB	11
Ilustración 4. Imagen Oblicua	12
Ilustración 5. Transformación Logarítmica	15
Ilustración 6. Transformación de Potencia	16
Ilustración 7. Transformación de Potencia	16
Ilustración 8. Histograma de una Imagen (Escalante B. , 2006)	17
Ilustración 9. Ejemplo de proceso de filtrado	18
Ilustración 10. Ejemplo de vecindad	19
Ilustración 11. Máscara de convolución	20
Ilustración 12. Máscara binomial de 3x3 (Hernanz, 2008)	26
Ilustración 13. Técnicas de ajuste de contraste empleadas	34
Ilustración 14. Técnicas de filtrado aplicadas sobre imágenes oblicuas	35
Ilustración 15. Transformada circular de hough	36
Ilustración 16. Técnicas de ajuste de contraste empleadas	36
Ilustración 17. Propuesta de solución representada en un diagrama de actividades	37
Ilustración 18. Resultados del preprocesado de tres imágenes oblicuas	52
Ilustración 19. Código fuente del método detectarCirculos()	53
Ilustración 20. Grafo de flujo asociado al método detectarCirculos() de la clase transformadaCircularHoug	53
Ilustración 21. Resultados de las pruebas	57
Ilustración 22. Imagen oblicua original cargada en la interfaz del software PANDOC. 57	
Ilustración 23. Proceso de preprocesamiento del algoritmo propuesto en este caso se muestra la imagen original y la imagen luego de aplicar ecualización del histograma adaptativo.	58
Ilustración 24. Resultado final del procesamiento aplicando como última técnica el operador Laplaciano pasó alto.	58
Ilustración 25. Aplicación del operador Canny	59
Ilustración 26. Comparación de Imágenes: Reducción de Ruido	61
Ilustración 27. Medidas de calidad basada en diagramas de dispersión de la magnitud del gradiente. Los píxeles por encima de la curva $y = x$, son los que se agudizan (color rosa), los que se encuentran por debajo son los suavizados (color rojo).	62

Resumen

La opacidad de la cápsula posterior es actualmente uno de los aspectos más importantes en la cirugía de catarata de los tiempos modernos. Sigue siendo la complicación postoperatoria tardía más frecuente tras la cirugía de catarata asociada con disminución de la agudeza visual, deterioro de la sensibilidad al contraste y problemas de deslumbramiento que conllevan importantes repercusiones sociales, médicas y económicas.

El software PANDOC provee al oftalmólogo de una herramienta por medio de la cual este es capaz de cuantificar numéricamente y detectar diferencias de opacidad (a veces imperceptibles para el ojo humano), logrando así una evaluación objetiva del grado de opacidad, lo cual se realiza de manera manual y depende mucho el resultado de la experticia del especialista. Dentro de las imágenes tratadas en los pacientes operados de catarata se encuentran las imágenes oblicuas, las cuales no son procesadas hasta el momento por el software PANDOC.

Por tal motivo la presente investigación se enmarca en la realización de un algoritmo, que detecte y realce automáticamente los bordes de las imágenes oblicuas, para que el software PANDOC le permita al especialista detectar diferencias de opacidad, a veces imperceptibles al ojo humano, a fin de propiciar un mejor diagnóstico.

Palabras claves: catarata, imagen oblicua, opacidad de la cápsula posterior, lámpara de hendidura, realce de bordes.

Introducción

En la actualidad cuando se necesitan servicios médicos, en muchos de los casos para poder obtener un diagnóstico acertado de las enfermedades, es indispensable para el clínico, apoyarse en resultados de pruebas de laboratorio, lo cual es posible debido al veloz desarrollo y proliferación de las tecnologías de adquisición de imágenes médicas, lo cual está revolucionando la medicina. Estas imágenes juegan un papel prominente en el diagnóstico y tratamiento de enfermedades, debido a que permiten a los científicos obtener información vital observando el interior del cuerpo humano de una forma no invasiva. La diversidad de principios de obtención de estas imágenes hace que existan varias modalidades de las mismas, entre las que aparecen las de resonancia magnética (MRI), tomografía axial computarizada (TAC o CT), ultrasonido (US), y microscopía celular, entre otras.

En el área de la oftalmología, la opacidad de la cápsula posterior (OPC) es un elemento fundamental de análisis en la actualidad, es uno de los aspectos más importante en la cirugía de catarata de los tiempos modernos (Alvarez Cancio, Rodríguez Puente, & Hernández López, 2013). Siendo la complicación posoperatoria tardía más frecuente tras la cirugía de catarata asociada con disminución de la agudeza visual, deterioro de la sensibilidad al contraste y problemas de deslumbramiento que conllevan a repercusiones sociales, médicas y económicas, siendo una preocupación a nivel mundial y que fundamentalmente interesan a nuestro país, los aspectos y procedimientos de la cirugía de cataratas, siendo las cataratas la causa más importante de ceguera en todo el mundo (López I. H., 2010).

Para el diagnóstico de las cataratas, se suele emplear una lámpara de hendidura, técnica usada para el examen de las diferentes estructuras del ojo. La catarata es la opacificación total o parcial del cristalino, donde la opacificación provoca que la luz se disperse dentro del ojo y no se pueda enfocar en la retina, creando imágenes difusas (Arntz, 2011). Con el tiempo se depositan partículas de un color café-amarillo que poco a poco van opacando el cristalino. El cristalino es un componente del ojo humano con forma de lente biconvexa que está situado tras el iris y delante del humor vítreo. Su propósito principal consiste en permitir enfocar objetos situados a diferentes distancias (Arntz, 2011). Este objetivo se consigue mediante un aumento, o disminución funcional de su curvatura y de su espesor, proceso que se denomina acomodación. Este hecho es el que le otorga su capacidad para refractar la luz, ayudando a la córnea a formar las imágenes sobre la retina.

La imagen que se observa desde una lámpara de hendidura corresponde a las estructuras del polo anterior del ojo: párpado, conjuntiva, córnea, iris, cristalino y cámara anterior (López I. H., 2010). De cada una de las estructuras del ojo mencionadas, en la investigación se hace referencia específicamente al cristalino y su cámara posterior.

Existen herramientas como el software PANDOC la cual es capaz de cuantificar y detectar diferencias de opacidad (a veces imperceptibles para el ojo humano), para lograr una evaluación objetiva del grado de la misma, es necesario el análisis de imágenes médicas, en las cuales los especialistas tienen un papel esencial debido a que, en correspondencia con su nivel de experticia, será el diagnóstico emitido.

Dentro de las imágenes tratadas en la cirugía de catarata, están las imágenes oblicuas provenientes de la lámpara de hendidura, las cuales se utilizan para la identificación de la opacidad de la cápsula posterior, la cual es una imagen con alta presencia de ruido, con mucha incidencia de luz en ciertas regiones y escasa en otras, lo cual dificulta al especialista poder realizar un diagnóstico adecuado.

Dada la situación problemática planteada anteriormente se identifica el siguiente **problema a resolver**: ¿Cómo disminuir el ruido de las estructuras de opacidad capsular en imágenes oblicuas provenientes de la lámpara de hendidura?

Objeto de estudio: Procesamiento digital de imágenes médicas.

Objetivo general: Desarrollar un algoritmo para la detección y realce de los bordes en imágenes oblicuas provenientes de la lámpara de hendidura.

Campo de acción: Tratamiento de imágenes oblicuas de la lámpara de hendidura para pacientes operados de catarata.

Objetivos específicos:

- Elaborar el marco teórico de la investigación mediante el estudio de las técnicas actuales de procesamiento digital de imágenes.
- Identificar los requisitos funcionales y no funcionales para el desarrollo de la propuesta de solución.
- Diseñar un algoritmo para la detección y realce de bordes en imágenes oblicuas provenientes de la lámpara de hendidura.
- Implementar los elementos definidos en la etapa del diseño.
- Validar los resultados comprobando la efectividad de las técnicas y filtros de realce de la imagen oblicua.

Idea a defender: Con el desarrollo de un algoritmo para la identificación de los bordes de la opacidad capsular se ofrecerá al oftalmólogo una herramienta que permite la detección automática de la incidencia de la Opacidad de la Cápsula Posterior (OCP) en pacientes sometidos a cirugía de catarata, al disminuir el ruido en las imágenes oblicuas provenientes de la lámpara de hendidura.

Para darle cumplimiento a estos objetivos se han definido las siguientes **tareas investigativas**:

1. Identificación de las diferentes técnicas de procesamiento de imágenes médicas.
2. Revisión de las técnicas de filtrado y mejora en imágenes médicas.
3. Estudio del uso de algoritmos de detección de bordes en imágenes médicas.
4. Descripción de los pasos a seguir para el uso del algoritmo propuesto.
5. Desarrollo de una propuesta de solución en función de los requisitos especificados.
6. Implementación de las técnicas de filtrado y mejora de la imagen oblicua.
7. Implementación del algoritmo de detección de bordes propuesto.
8. Integración de la solución al software PANDOC.
9. Validación de la solución propuesta a partir de la aplicación de pruebas unitarias y funcionales.

La lógica de la investigación se desarrolla a partir de indagaciones teóricas y empíricas en las cuales rige el enfoque general dialéctico materialista del conocimiento.

Métodos teóricos:

- Histórico-Lógico: A partir de este método se realizó un estudio con el fin de conocer la evolución y desarrollo de las técnicas actuales utilizadas para la detección y realce de bordes.
- Analítico-Sintético: Se realizó una búsqueda y un análisis profundo acerca de los algoritmos, teorías y documentos relacionados con el tema y a partir de esto se seleccionaron los elementos más útiles para el desarrollo del presente trabajo.

Métodos empíricos

- Experimento: A partir de este método se pudo probar las distintas técnicas y métodos de realce y mejora de imágenes médicas.

La presente investigación está estructurada en tres capítulos; a continuación se describe el objetivo principal de cada uno de ellos:

Capítulo 1: Fundamentación Teórica

En este capítulo se realiza la descripción y análisis desde el punto de vista teórico del problema general en que se enmarca la investigación, también se realiza un estudio del arte de los algoritmos y técnicas para el procesamiento digital de imágenes. Se describen las herramientas y tecnologías a utilizar durante el desarrollo de la solución. Son expuestos los argumentos que conllevan a la selección de dichas herramientas y tecnologías. Se explican las principales características de las metodologías de software.

Capítulo 3: Propuesta de Solución

En este capítulo se realiza una descripción detallada de la solución propuesta con todos los aspectos definidos en la fundamentación teórica. Esta solución contiene los artefactos necesarios para el desarrollo de la investigación, entre ellos se encuentran las historias de usuarios y las tarjetas CRC.

Capítulo 4: Resultados y validación del sistema.

En este capítulo se muestra el conjunto de pruebas realizadas a la herramienta y los resultados que arrojaron las mismas, como también definen pruebas de software para conocer si el sistema implementado satisface las necesidades del cliente.

Capítulo 1 Fundamentación Teórica

En el presente capítulo se realizará un estudio sobre los fundamentos teóricos de la investigación con el fin de obtener un mejor entendimiento del procesamiento digital de imágenes. Se realizará énfasis dentro de la etapa de preprocesamiento y segmentación, para poder obtener los conocimientos necesarios, y llevar a cabo la propuesta de un algoritmo que dé solución a la complicación postoperatoria que es la OPC (Opacidad de la Cápsula Posterior). Además de fundamentarse la necesidad de la realización de esta tarea de manera automática, desarrollándose un algoritmo que realce y detecte los bordes en imágenes oblicuas.

1.1 Imagen Digital

En numerosas ocasiones se piensa en "lo que se ve" como en "lo que está ahí", cuando en realidad el ojo humano solo percibe una pequeña zona del espectro electro-magnético (E-M) conocida como "espectro visible". Con el avance de la tecnología existe un amplio rango de la radiación E-M que se puede detectar y registrar como imágenes, para posteriormente analizar, transmitir o almacenar (Nuñez, 2013).

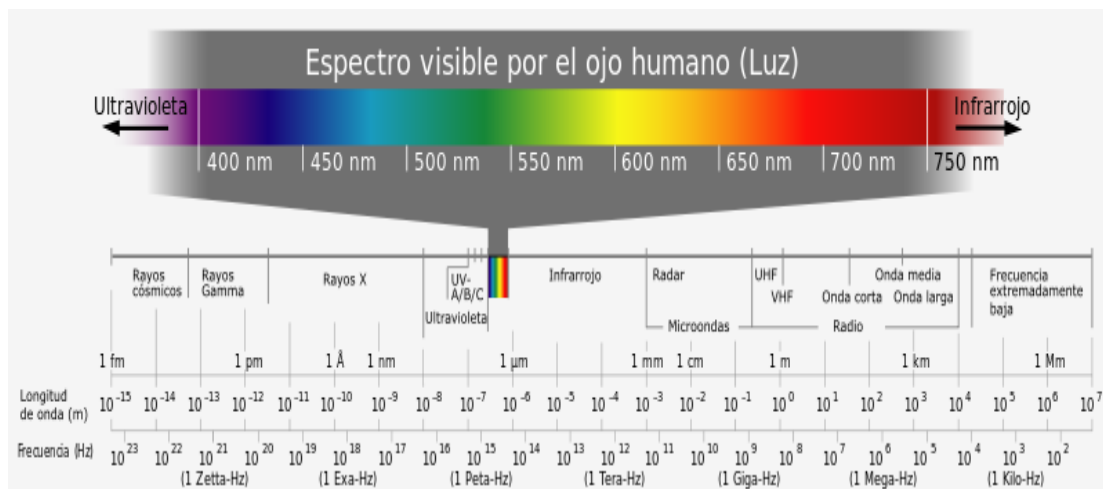


Ilustración 1. Espectro Electro-Magnético

Una imagen digital, suponiendo que se trate de una imagen monocromática, puede considerarse como una matriz cuyos índices de filas y columnas identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto. Esto es lo que se conoce como píxel, abreviatura del término en inglés "picture elements". Si por el contrario se tratara de una imagen en color, la información de ese punto se codificaría según el espacio de color que se fuera a utilizar, creando una imagen multicanal. Estos espacios de color proporcionan un método para

representar, ordenar y manipular los colores. El principio en el que se basan la mayoría de estos modelos consiste en que a partir de la combinación de tres colores primarios se puede obtener cualquier color de la naturaleza (Núñez, 2013) (Escalante B. , 2006).

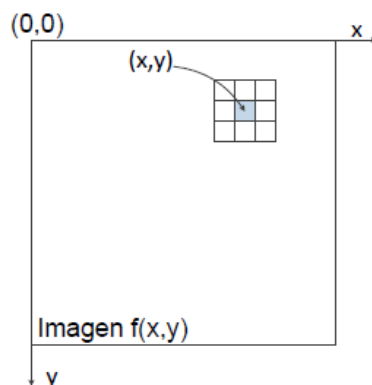


Ilustración 2. Imagen Digital

1.1 Imágenes a Color

Las imágenes digitales a color están gobernadas por los mismos conceptos de muestreo, cuantificación y resolución que las imágenes en escala de grises. Sin embargo, en lugar de un único valor de intensidad que expresa el nivel de gris, los píxeles de las imágenes a color están cuantificados usando tres componentes, que tendrán distinto significado según el modelo o espacio de color utilizado.

El modelo de color es una representación matemática del conjunto de colores que permite la especificación de un color de manera estándar. Los más conocidos son RGB (sigla en inglés de red, green, blue, en español «rojo, verde y azul»), usado en los gráficos por computador; CMY (siglas de cian, magenta y amarillo) ó CMYK (siglas de cyan, magenta, yellow y negro), usados en sistemas de impresión.

En este caso se utilizará debido al tipo de espacio de color e imagen a tratar, el cual cuantificará sus tres componentes o canales RGB.

En el modelo **RGB**, cada color aparece descompuesto en sus tres componentes espectrales primarias de **rojo, verde y azul**. Combinando distintas intensidades de estos tres colores primarios, se puede obtener todos los colores visibles.

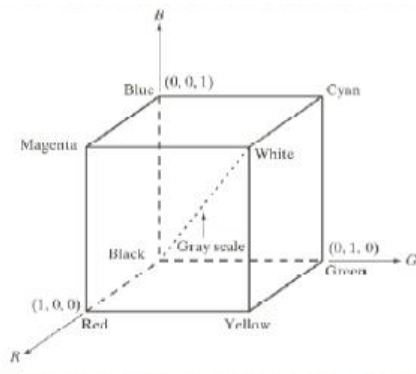


Ilustración 3. Modelo RGB

1.1.2 Imagen Oblicua

Imagen con un haz de luz que se proyecta como una franja de hendidura brillante, lo que permite el examen detallado del ojo en pequeños segmentos y que se detecten e identifiquen las regiones que presentan Opacidad Capsular. Estas son obtenidas a través de la lámpara de hendidura, las cuales presentan las siguientes características:

1. Se encuentran en un espacio de color RGB, por lo que está compuesta de los canales de Rojo, Verde y Azul.
2. Tienen como formato JPG, lo que significa que la imagen obtenida no es la misma que la deseada, esto es provocado por el algoritmo de compresión que utiliza dicho formato, que es un algoritmo de reducción con pérdida para disminuir el tamaño del archivo.
3. Tiene una profundidad de 24 bits (8 por cada uno de los canales del espacio de color), por lo que presenta variaciones de intensidad entre 0 y 255 colores.
4. Una resolución de 1024 x 768 píxeles.
5. Presenta una luz provocada por el mismo equipo que toma las imágenes que puede dificultar la detección de la Opacidad (esto provoca irregularidad en las intensidades y que existan bordes que no estén bien definidos).

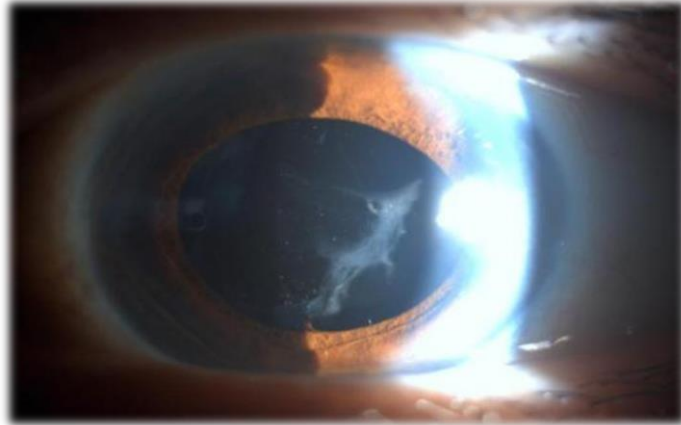


Ilustración 4. Imagen Oblicua

1.1.2.1 Lámparas de hendidura

La lámpara de hendidura es uno de los instrumentos de diagnóstico más comúnmente usados por un oftalmólogo de hoy. Proporciona iluminación y magnificación para examinar varias partes del ojo. Una serie de accesorios se pueden añadir a una lámpara de hendidura para convertirlo en un instrumento de medida. Uno puede medir la presión intraocular, la curvatura de la córnea, el espesor de la córnea, la distancia entre la córnea y el lente, el volumen de la cámara anterior, la opacidad, etcétera, utilizando accesorios diferentes. Algunas lámparas de hendidura tienen accesorios para conectar una cámara fotográfica. Las lámparas de hendidura también se utilizan para proporcionar energía láser en cualquier lugar del ojo para el tratamiento.

A continuación se describen los componentes principales del equipo y una descripción de los subsistemas principales: una lámpara de hendidura moderna consta de tres componentes principales (Srinivasan & R.D, 2003):

- Un sistema de iluminación - fuente de luz, espejos y prismas.
- Un sistema de magnificación - el biomicroscopio.
- Un sistema mecánico que une el sistema de magnificación con el sistema de iluminación y proporciona movimientos verticales y laterales para enfocar la luz en la parte deseada del ojo.

1.2 Opacidad de la Cápsula Posterior.

La cápsula posterior es una estructura que anatómicamente queda por detrás de la lente intraocular implantada en el paciente operado de cataratas, su transparencia es imprescindible para una buena recuperación visual del paciente. Desafortunadamente no son pocos los casos en los que se desarrolla esta anomalía, la opacificación de la cápsula posterior repercute negativamente en el resultado visual del paciente (Cancio, 2014).

1.3 Procesamiento de Imágenes.

El procesamiento digital de imágenes (PDI) es el procesado, entendiendo este como el almacenamiento, transmisión y representación de información, de imágenes digitales por medio de una computadora digital. El PDI consta con 5 etapas: (1) adquisición de la imagen, (2) pre-procesamiento de la imagen, (3) segmentación de la imagen, (4) representación y descripción y por último (5) reconocimiento e interpretación. Resulta conveniente para la investigación en curso, clasificar dichos procesos en categorías. De acuerdo a cada una de las etapas mencionadas, existen diferentes algoritmos involucrados, para un mejor entendimiento de las mismas se anuncian a continuación (Woods & González, 2002):

1. **Adquisición de la imagen:** Es el proceso a través del cual se obtiene una imagen.
2. **Pre-procesamiento:** incluye técnicas tales como la reducción de ruido y realce de detalles se realiza para que el resultado sea más apropiado para una aplicación en particular, dichas técnicas consisten en la mejora de la nitidez o aclarado de las imágenes desenfocadas, eliminación de ruido, mejora del contraste, mejora del ruido y detección de bordes.
3. **Segmentación de la imagen:** Subdivide la imagen en parte o aislar ciertos objetos de una imagen. Tareas habituales: búsqueda y selección de formas determinadas en la imagen máscaras de la imagen.
4. **Representación y descripción:** Mediante los procesos de descripción se obtienen características (tamaño, perímetro, etc.) convenientes para diferenciar un objeto de otro.
5. **Reconocimiento e interpretación:** El reconocimiento es el proceso que identifica los objetos y la interpretación le asocia un significado a un conjunto de objetos reconocidos. Generalmente es conveniente agrupar estas áreas de acuerdo con la complicación y el grado de detalles que lleva aparejada su implementación.

Según las necesidades de obtención de información dentro de las imágenes y las características propias de la imagen en cuestión, se analiza cuáles de los algoritmos existentes en cada una de las etapas planteadas muestra un mejor resultado. De las etapas planteadas anteriormente, las que serán objeto de esta investigación es el pre-procesamiento y segmentación. Se analizarán varias técnicas de procesamiento utilizadas en visión artificial.

En este capítulo se hará referencia a algunas técnicas, en correspondencia con la investigación, si se ajustan al tipo de imagen tratada y los resultados obtenidos.

1.3.1 Preprocesamiento de imágenes

Al digitalizar una imagen, es común la presencia de ruido o degradación, así como otros efectos indeseados como los reflejos luminosos ya mencionados, razón por la cual es importante restaurarla antes de ser procesada. En la etapa preprocesamiento de imágenes se intenta aumentar la calidad de la imagen con el fin de reconocer mejor las fallas que puedan existir en ella. Algunas de las técnicas empleadas en el Preprocesamiento son eliminación de ruido mediante filtros digitales, mejora del contraste y restauración (Oliva, 2014).

1. Técnicas de Ajuste de Contraste

Las técnicas de ajuste de contraste se enfocan en mejorar o realzar la calidad de las imágenes para facilitar el análisis visual de las mismas y la preparación para otros tratamientos digitales.

Técnicas de Punto

Las transformaciones hacen corresponder al nivel de gris de la imagen original (o de entrada) un nuevo nivel de gris dado por una función de transformación $T()$; en otras palabras:

$$s = T(r) \quad (1)$$

Donde r es el nivel de gris de entrada (de la imagen original) y s es el nivel de gris de salida (de la imagen de salida) (W. O. Achicanoy & L. F. Giraldo, 2005).

La cual produce un nivel s para cada valor del píxel r en la imagen original. Sin embargo, para llevar a cabo esta ecualización es necesario establecer dos restricciones, ya que sin estas se podrían obtener resultados erróneos:

$T(r)$ Debe ser creciente, esto indica que la pendiente de la función no puede ser negativa, ya que si fuera así, la imagen se vería afectada en su totalidad, porque lo que se invertían sus tonalidades.

$$\frac{dy}{dx} > 0$$

$T(r)$ debe tener inversa y debe ser creciente. Para resolver un problema como este se debe suponer que r y s son continuas, donde:

- r y s son variables aleatorias y continuas.
- $P(r)$ y $P(s)$ son funciones de densidad de probabilidad de r y s .

Transformación Logarítmica

En esta transformación T se define como

$$T(r) = c \log(1 + r) \quad (2)$$

Donde c es una constante positiva, $L - 1$ es el valor máximo del rango dinámico de niveles de gris codificados en la imagen de entrada y $0 \leq r \leq (L - 1)$. La ilustración 5 muestra la gráfica de la transformación logarítmica (W. O. Achicanoy & L. F. Giraldo, 2005).

Usada para expandir los valores de píxeles oscuros a píxeles más claros en una imagen mientras se comprimen los valores de alto nivel.

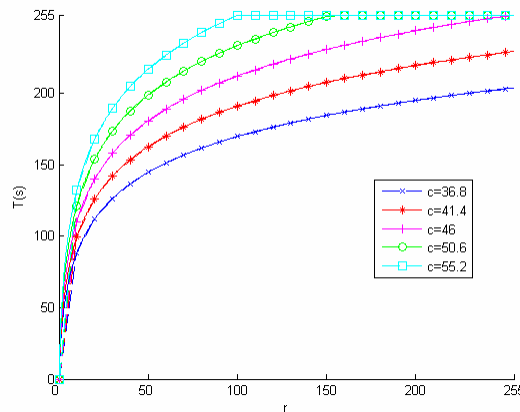


Ilustración 5. Transformación Logarítmica

Se observa la variación de la transformación con distintos valores del parámetro c , utilizando el logaritmo natural.

Transformación de potencia

Donde T se define como:

$$T(r) = c \cdot r^\gamma \quad (3)$$

c y γ son constantes positivas y $0 \leq r \leq (L - 1)$, $L - 1$ es el valor máximo del rango dinámico de niveles de gris codificados en la imagen de entrada. La ilustración 6 muestra la gráfica de la transformación de potencia.

La transformada de potencia con valores fraccionados de γ estrecha los valores oscuros de entrada y amplía los niveles más claros en la salida de la imagen.

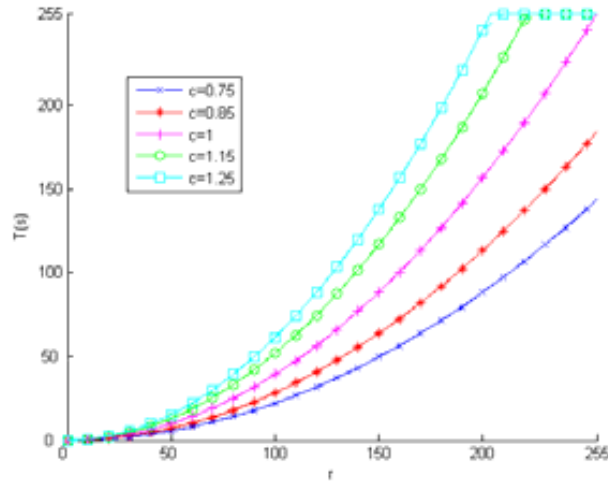


Ilustración 6. Transformación de Potencia

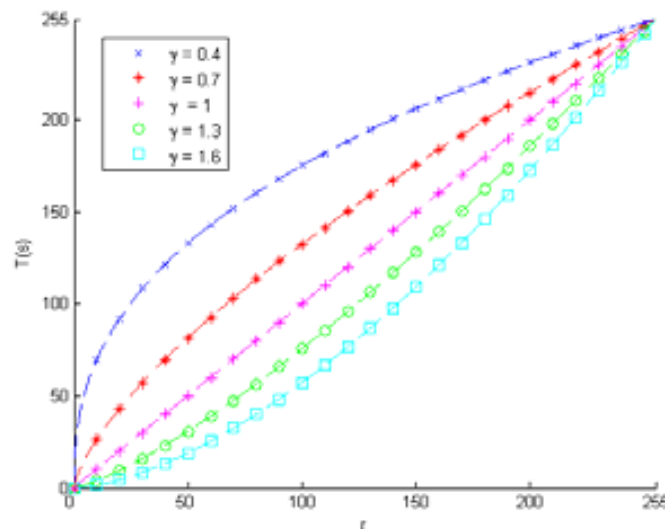


Ilustración 7. Transformación de Potencia

Se observa la variación de la transformación con distintos valores de los parámetros ilustración 6 c y ilustración 7 γ .

Histograma de una Imagen

El histograma de una imagen es la representación gráfica de la distribución que existe de las distintas tonalidades de grises con relación al número de píxeles o porcentaje de los mismos. La representación de un histograma ideal sería la de una recta horizontal, ya que eso indicaría que todos los posibles valores de grises están distribuidos de manera uniforme (Escalante B. , 2006) (Woods & González, 2002).

Un histograma muestra la distribución de los píxeles que hay en una imagen para cada valor de luminosidad, mostrado en **256** columnas verticales alineadas, y que cada una, muestra un distinto valor de luminosidad de forma progresiva entre el **0** a la izquierda

(negro profundo sin detalle) y **255** a la derecha (blanco total sin detalle) (Escalante B. , 2006).

Un histograma de una imagen en **RGB** de 8 bits, el eje horizontal representa los diferentes niveles de luminosidad de la imagen, es decir la cantidad de píxeles de cada grado de luminosidad encontrados en la imagen, en un rango que va como desde el **0** (negro profundo y sin detalle) a la izquierda del todo hasta el **255** (blanco puro sin detalle) a la derecha del todo (K.Pratt, 2001).

La altura del histograma representa la cantidad de píxeles que aparecen en la imagen con cada determinado valor de luminosidad, es decir un pico de gran altura en determinada zona del histograma, quiere decir que hay muchas zona de la imagen con el valor de luminosidad correspondiente a la zona donde se encuentra dicho pico en el histograma como se muestra en la siguiente ilustración 8:

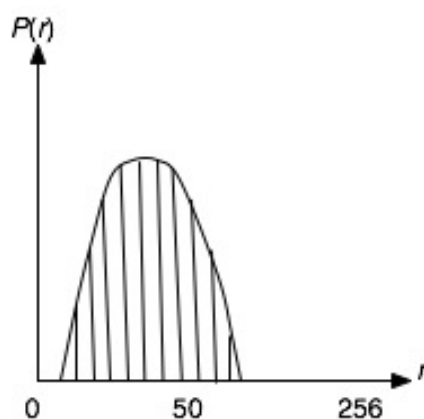


Ilustración 8. Histograma de una Imagen (Escalante B. , 2006)

Ecuación de Histogramas

La ecuación de un histograma consiste en encontrar una transformación, con la cual el histograma tenga una representación uniforme, es decir, con la ecuación se trata de igualar lo más posible el histograma de una imagen al histograma ideal. Sea r la variable que representa los niveles de gris de la imagen. Primeramente se asume que nuestro histograma ha sido normalizado en el intervalo $[0,1]$, donde $r = 0$, representa el nivel de gris negro y $r = 1$ el nivel de gris blanco.

Es una forma de manipulación de histograma que reduce automáticamente el contraste para las áreas claras y aumentar par las oscuras. También expande los niveles de gris

a lo largo de todo intervalo. Consiste en una transformación no lineal que considera la distribución acumulativa de la imagen original, para generar una imagen resultante cuyo histograma será aproximadamente uniforme (Felipe, Moya, Peña, & Segarra, 2001) (Escalante, 2006).

Ecuación de Histogramas Adaptativo (CLAHE)

Esta técnica opera en pequeñas regiones de la imagen en lugar de en toda la imagen. El mejoramiento de contraste es local, por lo que el histograma de la región de salida se ajusta aproximadamente al histograma especificado por el parámetro de distribución. Las vecindades de la zona local son entonces combinadas utilizando interpolación bilineal para eliminar artificialmente propiedades inducidas. El contraste, especialmente en áreas homogéneas, puede ser limitado para evitar incrementar cualquier ruido que se pueda presentar en la imagen (Fuente-Bermudez, Orlena-Rodriguez, & González-Barbosa, 2015).

Interpolación bilineal:

Usando interpolación bilineal, al píxel en cuestión se le asigna un valor medio ponderado de las intensidades de los cuatro píxeles que le rodean (vecindad 4), donde los factores de ponderación vienen dados por la distancia entre el píxel y los de su entorno. (Echeverri Arias, Manrique Losada, Javier Moreno, & Bravo, 2009).

2. Técnicas de filtrado

El proceso de filtrado consiste en la aplicación de técnicas con el fin de reducir al máximo el ruido que pueda contener una imagen, producido por el tipo de iluminación, las sombras entre los objetos, pequeños puntos no deseados dentro de la imagen y otros efectos que pueden estar presentes en una imagen digital como resultado del muestreo, cuantización, transmisión o por perturbaciones en el sistema como pueden ser las partículas de polvo en el sistema óptico. Para la eliminación de todo este tipo de ruido se emplean diferentes tipos de filtros.

Filtrar una imagen $f()$ consiste en aplicar una transformación $T()$ para obtener una nueva imagen $g()$ de forma que ciertas características son acentuadas o disminuidas (Santillan, 2008):

$$g(x, y) = T[f(x, y)] \tag{4}$$

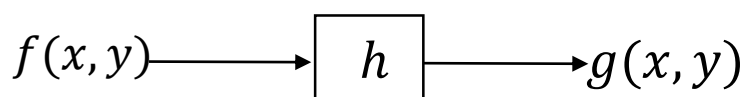


Ilustración 9. Ejemplo de proceso de filtrado

Se puede considerar que la señal (imagen) pasa a través de una caja (filtro) cuya salida es la imagen filtrada.

De acuerdo a la teoría de sistemas, al pasar una señal por un sistema lineal, la salida es la convolución de la transformación del sistema con la señal de entrada:

$$g(x, y) = h(x, y) * f(x, y) \quad (5)$$

Como punto de partida se describirán conceptos referentes al filtrado digital de imágenes y se explicarán una serie de conceptos para lograr un mejor entendimiento de estas técnicas.

Entorno de vecindad:

La vecindad es una matriz bidimensional de valores de píxeles con un número impar de filas y columnas. El píxel de interés que normalmente es reemplazado por un nuevo valor producto de la aplicación de un algoritmo, se ubica por lo general, en el centro de la vecindad (Escalante, 2006).

	x			x	x	x
x	o	x		x	o	x
	x			x	x	x

Ilustración 10. Ejemplo de vecindad

Convolución:

Cada píxel de la imagen resultado es combinación lineal de varios píxeles vecinos de la imagen original. Las máscaras de convolución tienen por lo general un número impar de filas y columnas y su tamaño frecuentemente es 3x3, su contenido depende del tipo de procesamiento que se desea implantar.

La operación convolución es una operación sobre un vecindario en el cual cada píxel de salida es una suma ponderada de los píxeles vecinos. La matriz de pesos es llamada núcleo de convolución, también conocido como filtro.

El filtrado espacial se realiza trasladando una matriz rectangular de dos dimensiones (también llamada ventana, kernel, máscara o núcleo) que contiene "pesos" o ponderaciones sobre la imagen en cada localización de píxel. Se evalúa el píxel central de la ventana de acuerdo con los píxeles de alrededor y sus valores de ponderación. Cuando un nuevo valor es así calculado, se desplaza la ventana sobre el siguiente píxel, realizando la misma operación. Este proceso de evaluar la vecindad ponderada del píxel se denomina "convolución bidimensional", y a la matriz de filtro se le conoce como "kernel de convolución". En general la convolución de una imagen f de MxN con una máscara h de mxn está dada por la siguiente expresión (Pinilla, Alcalá, & Arisa, 1997):

$$g(x, y) = \sum_{i=-1}^a \sum_{j=-b}^b f(x + i, y + j) \cdot h(i, j) \quad (6)$$

$h(-1,-1)$	$h(-1,0)$	$h(-1,1)$
$h(0,-1)$	$h(0,0)$	$h(0,1)$
$h(1,-1)$	$h(1,0)$	$h(1,1)$

Ilustración 11. Máscara de convolución

Existen diferentes algoritmos de filtrado digital entre los cuales se encuentran:

Filtros espaciales

El filtrado espacial es la operación que se aplica a una imagen para resaltar o atenuar detalles espaciales con el fin de mejorar la interpretación visual o facilitar un procesamiento posterior, y constituye una de las técnicas comprendidas dentro del realce de imágenes. Ejemplos comunes incluyen aplicar filtros para mejorar los detalles de bordes en imágenes, o para reducir o eliminar patrones de ruido. El filtrado espacial es una operación "local" en procesamiento de imagen, en el sentido de que modifica el valor de cada píxel de acuerdo con los valores de los píxeles que lo rodean; se trata de transformar los niveles de gris originales de tal forma que se parezcan o diferencien más de los correspondientes píxeles cercanos (Aldalur & Santamaría, 2002).

- **Filtros paso alto**

Enfatizan las altas frecuencias, para mejorar o afilar las características lineales como los límites en general, es decir realce de bordes. Realizan por tanto el efecto contrario a los filtros pasa bajos, eliminando las bajas frecuencias.

- **Filtros paso bajo**

Enfatizan las bajas frecuencias, suavizando las imágenes y atenuando ruidos. Se trata de asemejar el nivel de gris de cada píxel al nivel de gris de los píxeles vecinos, reduciendo la variabilidad espacial de la imagen. Ello produce un borronado de los bordes, perdiéndose en nitidez visual de la imagen, pero ganando en homogeneidad.

Filtros espaciales lineales

El proceso de convolución espacial se mueve a través de la imagen de entrada, píxel por píxel, ubicando los píxeles resultantes en la imagen de salida. El valor digital de brillo de cada píxel en la imagen de salida depende de un grupo de píxeles de entrada que

rodean al píxel que se está procesando. Al utilizar la información del brillo de los píxeles vecinos a un píxel central, la convolución espacial calcula la actividad de frecuencia espacial en esa área, y por lo tanto, es capaz de filtrar en base al contenido de frecuencia espacial existente. El proceso de convolución espacial utiliza un promedio ponderado del píxel de entrada y de sus vecinos inmediatos para calcular el valor de brillo del píxel de salida. El grupo de píxeles utilizados en el cálculo del promedio ponderado se conoce como núcleo (Aldalur & Santamaría, 2002).

Filtro de la media

La media del ruido en un entorno de vecindad es cero, si se calcula la media se elimina el ruido. La imagen es constante en un entorno de vecindad, por lo que su valor es igual a la media. Pero esta característica no se cumple en los bordes. Se logra haciendo convolucionar la imagen con una máscara de 7x7, 5x5 o 3x3 generalmente.

Se puede implementar también promediando los valores del entorno de vecindad (Cepero Ruz & Cabrera Rodríguez, 2010).

El algoritmo es muy usado para atenuar el ruido gaussiano y aleatorio. Este filtro tiende a la pérdida de detalles, disminuir la nitidez y difuminar los bordes de la imagen que será procesada. Por lo que mientras mayor sea la cantidad de valores a promediar más ruido se reducirá, pero también habrá una mayor pérdida de detalles.

Filtro Laplaciano

El operador Laplaciano se define como una derivada de segundo orden, puede trabajar con imágenes donde las variaciones de intensidad no sean abruptas. El filtro Laplaciano es un buen ejemplo de un operador de derivada de segundo orden, se distingue de los otros operadores porque es omnidireccional, es decir, destacará los bordes en todas las direcciones (Oyarzo, 2006). Este operador producirá bordes más acentuados que la mayoría de las otras técnicas, estos toques de luz incluyen pendientes positivas y negativas de la intensidad.

El borde Laplaciano de una imagen puede ser encontrado convolucionando con elementos estructurales, la imagen exhibe un cambio del signo del borde de la imagen. Estos cambios de signo son referidos como pasos cero. Después del operador de convolución, la imagen se debe procesar para encontrar que estos pasos cero y para fijar, por consiguiente, los píxeles de la salida (Oyarzo, 2006).

El filtro Laplaciano de una imagen $I(x, y)$ es una derivada de segundo orden definido por:

$$\nabla^2 f = \frac{d^2 f}{d^2 x} + \frac{d^2 f}{d^2 y} \quad (7)$$

En general, se suele tomar el valor negativo del Laplaciano. Al igual que en el caso del gradiente se puede implementar en forma digital de varias formas. Puesto que el

Laplaciano es un operador de derivación, la suma de los coeficientes debe ser cero. Además, el coeficiente asociado con el píxel central debe ser positivo y todos los demás coeficientes negativos o cero, o viceversa.

El Laplaciano da lugar a bordes dobles y no permite determinar direcciones. En general, juega un papel secundario en la detección de bordes para determinar si un píxel está en la zona clara o en la zona oscura del borde a través del signo del Laplaciano.

El Laplaciano se puede utilizar para realzar una imagen de la siguiente forma:

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{si el coeficiente central de la máscara es negativo} \\ f(x, y) + \nabla^2 f(x, y) & \text{si el coeficiente central de la máscara es positivo} \end{cases} \quad (8)$$

Los elementos estructurales del operador Laplaciano coinciden con los elementos estructurales de filtro paso alto. Esto se debe a que el Laplaciano realza los bordes, es decir, las altas frecuencias de la imagen, sin considerar la orientación, por lo que además de utilizarse para la detección de bordes se utiliza también para el filtrado paso alto de imágenes (Aldalur & Santamaría, 2002) (Gómez, 2009).

Filtros Espaciales No Lineales

Existen otros filtros espaciales que no son función lineal de los valores de brillo de los píxeles de una imagen. Es decir, no se calculan como una suma lineal de elementos (brillo de los píxeles) multiplicados por pesos constantes (coeficientes de la máscara). Estos filtros se conocen como filtros espaciales no lineales.

Constituyen también técnicas de procesamiento por grupo de píxeles, operando sobre un núcleo de píxeles de entrada que rodean a un píxel central. La diferencia es que en lugar de utilizar un promedio ponderado, emplean otras técnicas que combinan los valores de brillo del grupo de píxeles de entrada (Aldalur & Santamaría, 2002) (Woods & González, 2002).

Filtro de la mediana

Una de las técnicas no lineales más utilizadas para el procesado de imágenes, en cuanto a la reducción de ruido, es el filtrado de la mediana, partió de la idea de conseguir un suavizado de imágenes y eliminación de ruido aplicando una técnica no lineal, pero de simple implementación.

La idea es que en un conjunto de píxeles cercanos, valores muy alejados del resto, serán valores que corresponden a píxeles ruidosos. La operación consiste en analizar una matriz de píxeles de $N \times N$ y reemplazar el píxel central por el valor de la mediana de todos ellos (Woods & González, 2002) (Oliva, 2014).

Aplicando este método se deben escoger ventanas de $N \times N$ con N impar, para tener bien diferenciado el píxel central. Tamaños habituales son ventanas de 3×3 , de 5×5 y de hasta 7×7 . Esto lleva a pensar en el hecho de la elección del tamaño de ventana como algo importante, puesto que un valor pequeño puede no eliminar bien el ruido, mientras que un valor demasiado alto es capaz de distorsionar la imagen. Para la elección de la ventana, en definitiva, no existe una regla fija, sino que se trata de escoger el orden que mejor resultados se obtenga con una determinada imagen. En general, se suele decir que un tamaño de ventana es bueno si el número de píxeles ruidosos dentro de la ventana es menor que la mitad de píxeles de la ventana.

Otro de los factores importantes a escoger en el filtro de mediana es la forma de la ventana, lo que habitualmente se denomina máscara. De nuevo la elección de la máscara supone un grado de libertad más a la hora de filtrar nuestra imagen. La forma de la ventana dice los píxeles a los cuales se les obtiene la mediana, que finalmente deberá reemplazar al central (Cepero Ruz & Cabrera Rodríguez, 2010).

- La ventana cuadrada es la que mayor grado de reducción de ruido provoca, pero también es la que más fácilmente distorsiona la imagen.
- La ventana en cruz es buena para imágenes con muchos bordes.
- La ventana en equis es un término medio entre las dos anteriores.

1.3.2 Segmentación de imágenes

La segmentación es el proceso que subdivide una imagen en sus partes constituyentes u objetos, y es uno de los elementos fundamentales en el análisis automatizado de imágenes, debido a que es en esta etapa donde se extraen los objetos de interés para un procesamiento posterior, como puede ser su descripción y reconocimiento. El proceso de segmentación termina cuando los objetos de interés para una aplicación han sido aislados. En la práctica suelen distinguirse dos tipos de segmentación: de regiones o áreas y de objetos. La operación es común, lo que varía es aquello que se desea segmentar.

Existen técnicas de segmentación que se pueden ver en, (Valdés, 2010) (Azpiroz, 2000) y (Jahne, 2002). En general, los métodos de segmentación, se clasifican como se indica a continuación:

- Métodos basados en la comparación con umbrales en el histograma, en los cuales se obtiene un umbral de comparación para el agrupamiento de los píxeles.
- Métodos basados en la detección de discontinuidades, en donde se calculan las derivadas de la función de la imagen, éstos dividen la imagen a partir de cambios bruscos de los niveles de grises.

- Métodos basados en la propiedad de similitud de los valores de los niveles de grises, que permiten la agrupación de puntos a partir de ciertos criterios de homogeneidad.
- Métodos heurísticos de segmentación, los cuales se basan en el conocimiento previo de la imagen a segmentar y en la experiencia del observador, ellos incluyen en muchas ocasiones los métodos supervisados de segmentación.

Transformada de Hough

La transformada de Hough es una técnica utilizada para aislar características de forma particular dentro de una imagen. La idea básica es encontrar curvas que puedan ser parametrizadas como líneas rectas, polinomios y círculos (Urrea & Ospina, 2004) (Gómez, 2009).

Explícitamente la Transformada de Hough circular se utiliza para la detección de círculos los cuales pueden ser descritos mediante su centro y radio. La descripción de parámetros utilizada es (Arriaga & Riverón, 2012):

$$f(x, y, x_0, y_0, r) = (x - x_0)^2 + (y - y_0)^2 - r^2 = 0 \quad (9)$$

donde la terna (x_0, y_0, r) conforma el conjunto de parámetros que expande el espacio de Hough a uno tridimensional; luego el arreglo de acumulación $H(x_0, y_0, r)$ será de dimensión 3D.

Algoritmo de detección de círculos

- I. Iniciar el arreglo de acumulación bidimensional del espacio de parámetros de Hough, cuya cantidad de elementos va en función de la resolución de la imagen de origen.
- II. Para cualquier píxel que satisfaga (valor de referencia del campo gradiente, ajustable experimentalmente), incrementar en uno todos los elementos del arreglo de acumulación, que cumplan con las dos ecuaciones simultáneas:

$$r \text{ tal que } : \begin{cases} x_0 = x \pm r \text{ Cos}\varnothing \\ y_0 = y \pm r \text{ Sen}\varnothing \end{cases} \quad (10)$$

$$H(x_0, y_0, r) > T \quad (11)$$

(Umbral de detección, también asignado experimentalmente)

- III. Cualquier elemento representará un círculo de radio r y centro en (x_0, y_0) contenido en la imagen de origen (Z, F., & Arteaga, 2008).

Detección de bordes

La detección de borde es un elemento crítico en el procesamiento de imágenes, la función de la detección de bordes es identificar los límites de regiones homogéneas en

una imagen basado en sus propiedades como son la intensidad y la textura. Muchos algoritmos de detección de bordes han sido desarrollados sobre la base del cálculo de la intensidad del vector gradiente el cual en general es consciente al ruido en de la imagen (Roushdy, 2006) (Hermosilla, Bermejo, Balaguer, & Ruiz).

Es una técnica de procesamiento de imágenes para la búsqueda de los límites de objetos dentro de las imágenes. Funciona mediante la detección de discontinuidades en el brillo. La detección de bordes se utiliza para la segmentación de imágenes y extracción de datos en áreas como procesamiento de imágenes, visión por computadora, y la visión de máquina (Saleh Alamri, Kalyankar, & Khamitkar, 2010).

Uno de los problemas que se puede encontrar en la detección de bordes es el ruido. El ruido es toda información no deseada que contamina la imagen, pudiendo tener distintos orígenes.

Un borde, dentro de una imagen, es considerado un tipo de discontinuidad, por lo que este puede ser detectado usando derivadas de primer y segundo orden.

La derivada de primer orden para una imagen es el gradiente, el cual, para una función en dos dimensiones. La segunda derivada tiene una respuesta más fuerte ante detalles como puntos aislados o líneas (Valencia Delgado, Moreira, & Chavez, 2005).

Operador Sobel

Este operador detecta los bordes horizontales y verticales por separado sobre una imagen en escala de grises. El cual convierten 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 (Coto, 2005).

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 (Alamri, Kalyankar, & Khamitkar, 2010) (Shrivakshan & Chandrasekar, 2012) (Spontón & Cardelino, 2015).

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 (Coto, 2005).

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.

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

Ilustración 12. Máscara binomial de 3x3 (Hernanz, 2008)

Operador Canny

El Algoritmo de Canny fue desarrollado por John Canny en 1986. Este método está considerado como uno de los más eficientes para la detección de bordes mediante el empleo de máscaras de convolución, porque es menos sensible al ruido y es capaz de detectar bordes débiles verdaderos (Eches Salcedo & Chiriboga Maldonado, 2003).

Es un método de detección de contornos. En éste se utiliza la primera derivada para la detección de contornos, ya que toma el valor cero en las regiones en donde la intensidad no varía y tiene un valor constante en toda la transición de intensidad (Saini, Bhavesh, & Shraey, 2010) (Shrivakshan & Chandrasekar, 2012).

Canny utiliza el cálculo de variaciones, una técnica que encuentra la función que optimiza una función dada. La función óptima definida por Canny está descrita por la suma de cuatro funciones exponenciales.

El algoritmo de Canny se divide diferentes etapas (Moeslund, 2009)

1. Suavizado de la Imagen
2. Encontrar gradientes
3. Suprimir los puntos que no sean máximos
4. Umbral doble

1-Suavizado de la imagen: El detector de bordes Canny utiliza un filtro que se basa en el concepto de la primera derivada gaussiana. Se convoluciona $f()$ con filtros gaussianos unidimensionales (La operación convolución consiste en recorrer la imagen global centrandolo la máscara en cada uno de los píxeles, aplicar la máscara sobre ellos y guardar el resultado en lo que será la nueva imagen.) y de esta forma la imagen se suaviza (eliminación de ruidos).

2-Encontrar el gradiente: Determinando el gradiente de la imagen se puede localizar dentro de la imagen los puntos de mayor intensidad. El borde de una imagen apuntar a varias direcciones por lo que este algoritmo trabaja con cuatro filtros para detectar bordes horizontales, verticales y diagonales. El operador de detección de bordes

(Roberts, Prewitt, Sobel, por ejemplo) devuelve un valor para la primera derivada en la dirección horizontal G_x y vertical (Corral, 2012) (Shrivakshan & Chandrasekar, 2012) (Saleh Alamri, Kalyankar, & Khamitkar, 2010).

Las magnitudes del gradiente, que son los puntos fuertes de los bordes, se pueden determinar mediante la distancia Elucidela y aplicando el teorema de Pitágoras:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (9)$$

Se puede calcular simplificando utilizando la distancia y así disminuir complejidad computacional.

$$|G| = |G_x| + |G_y| \quad (10)$$

Para saber dónde están exactamente los bordes se calcula, pues la magnitud de los gradientes indica los bordes pero en ocasiones son gruesos y no sabe exactamente por lo que se utiliza la siguiente ecuación:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (11)$$

3-Suprimir los puntos que no sean máximos: Se realiza, por tanto, un thresholding, (Es una técnica utilizada para convertir imágenes en color o escala de grises a imágenes binarias, a partir de un determinado valor umbral, por ejemplo, si píxel $(i, j) < \text{valor umbral} \rightarrow \text{píxel}(i, j) = 0$, si píxel $(i, j) \geq \text{valor umbral} \rightarrow \text{píxel}(i, j) = 1$) eliminando aquellos píxeles que no tienen una magnitud (módulo del gradiente) alta.

- Los píxeles que resultan ser máximos, se marcan con un recuadro de borde blanco. El resto de píxeles se suprimirá.

4-Umbral doble: Los bordes que superen el umbral alto, son clasificados como fuertes, los que estén debajo del umbral bajo se eliminan, y los bordes intermedios se clasifican como débiles. La umbralización con histéresis requiere de dos umbrales, alto y bajo. Haciendo la suposición de que los bordes importantes deben ser curvas continuas en la imagen, esto permite poder descartar algunos píxeles ruidosos que no constituyen una línea, se comienza la umbralización mediante la aplicación de un umbral alto. Esto marca los bordes que pueden ser reales.

Los bordes fuertes son incluidos en la imagen final ya que son interpretados como bordes reales y los bordes débiles solo se adicionan en la imagen final si son conexos con un borde fuerte.

Los bordes débiles debido a los bordes reales son mucho más propensos a estar directamente conectados a los bordes fuertes. Sin embargo, los bordes débiles que vienen de variaciones de color estarán distribuidos de forma independiente a los bordes fuertes (Moeslund, 2009) (Marmolejo, 2012).

1.4 Herramientas usadas durante el desarrollo

En el presente epígrafe se describen las herramientas y tecnologías a utilizar durante el desarrollo del sistema, el análisis realizado, llevará a conocer cuáles son las más adecuadas para el desarrollo.

1.4.1 Lenguajes de Programación

Un lenguaje de programación es un idioma artificial diseñado para expresar instrucciones que pueden ser llevadas a cabo por un ordenador. Puede usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión o como modo de comunicación humana. Permiten especificar de manera precisa sobre qué datos debe operar una computadora, cómo deben ser almacenados o transmitidos y qué acciones debe tomar bajo una gran cantidad de opciones posibles. Todo esto, a través de un lenguaje que intenta ser relativamente próximo al lenguaje humano o natural (Suárez, 2015).

Para el desarrollo de esta aplicación se escogió como lenguaje de programación a Java, por ser un lenguaje Orientado a Objetos, lo que le proporciona una gran reusabilidad; independencia de la plataforma además de ser el lenguaje en que está desarrollado el software, al cual se le agregará esta propuesta de solución.

Dentro de las características fundamentales de Java se encuentran (Fernández, 2004):

- Simple: elimina la complejidad de otros lenguajes como C y se enfoca en el contexto de los lenguajes orientados a objetos.
- Robusto: maneja la memoria de la computadora para que el programador no se tenga que preocupar por ello, además de realizar verificaciones en busca de errores lo mismo en tiempo de compilación que en tiempo de ejecución.
- Portable: un programa compilado de Java puede ser utilizado por cualquier computadora que tenga instalado el intérprete de Java, ya que su código compilado es interpretado.
- Multiproceso: puede ejecutar diferentes líneas de código al mismo tiempo.
- Dinámico: no es necesario que compile todas las clases de un programa para que este funcione.

Al efectuar al menos un cambio en alguna de las clases, Java se encarga de realizar un enlace dinámico o una carga dinámica para encontrar las clases.

- Interpretado: corre en máquina virtual.

1.4.2 Entorno integrado de desarrollo

Para el desarrollo de la solución se seleccionó como Entorno de Desarrollo Integrado (IDE, según sus siglas en inglés) el NetBeans 8.0. Es libre y de código abierto; posee todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java. Es conocido por la integración con el lenguaje Java, facilita el desarrollo utilizando funcionalidades como completamiento de código, coloreo de sintaxis, permite la utilización y edición de los componentes visuales de forma sencilla. Incluye también el control de versiones, lo cual representa una ventaja debido a que permite administrar las diferentes versiones del código fuente. (Oracle. Java., 2015).

1.4.3 Herramienta de prueba

Para la realización de pruebas de diversos algoritmos y técnicas sobre la imagen oblicua, se utilizó la herramienta de software MATLAB (abreviatura de **MATrix LABoratory**, "laboratorio de matrices"). MATLAB, en su contenido básico, es un entorno integrado de trabajo que permite el análisis y la computación matemáticos interactivos de forma sencilla con más de 500 funciones matemáticas, estadísticas y técnicas implementadas, así como la visualización a través de todo tipo de gráficos de ingeniería y científico (López, 2002). Además, el entorno básico de MATLAB se complementa con una amplia colección de cajas de herramientas que contienen funciones específicas para determinadas aplicaciones en diferentes ramas de las ciencias y la ingeniería. Se hizo énfasis específicamente en el área del procesamiento de imágenes MATLAB.

Específicamente se trabajó con la Toolbox de procesamiento digital de imágenes de MATLAB, Image Processing Toolbox (IPT) le proporciona a MATLAB un conjunto completo de algoritmos y herramientas gráficas para el procesamiento, análisis y visualización de imágenes y para el desarrollo de aplicaciones y de nuevos algoritmos en el campo del procesamiento y análisis de imágenes digitales. Además MATLAB, el entorno matemático sobre el que se sustenta, es ideal para este procesamiento digital de imágenes ya que las imágenes digitales son matrices al fin y al cabo (Fiter, 2012).

Entre las funciones principales de la IPT destacan (Fiter, 2012):

- Mejora y filtrado de imágenes y enfoque de imágenes borrosas.
- Análisis de imágenes, incluyendo segmentación, morfología, extracción de funciones y medición.
- Transformaciones geométricas y métodos de registro de imágenes basados en intensidad.

1.4.4 Herramienta para el modelado

La herramienta de modelado seleccionada es el Visual Paradigm para UML (VP-UML v8.0), una de las líderes del mercado de las llamadas herramientas de Ingeniería de Software Asistida por Computadora (CASE, según sus siglas en inglés).

VP-UML v8.0 soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML, según sus siglas en inglés), SysML, BPMN, XMI, entre otros. Ofrece un conjunto completo de herramientas, brindando a los equipos de desarrollo de software todo lo necesario para la obtención de requisitos, planificación de software, planificación de controles, modelado de clases y modelado de datos.

Ofrece interoperabilidad entre diagramas ya que es capaz de exportar los diagramas de un modelo a otro con mucha facilidad, ahorrando de esta manera tiempo, lo cual es crucial para el desarrollo. Hace posible la generación de código Java desde los diagramas.

1.4.5 Programa Analizador de Opacidad Capsular: PANDOC

El software PANDOC es una herramienta para la cuantificación y análisis de la OCP. Es un software intuitivo con una curva de aprendizaje pequeña, el cual enfoca sus resultados en pos del análisis estadístico que aporten información de alto valor para el estudio y disminución de esta anomalía postoperatoria. Es un software de nuevo tipo, ya que a diferencia del resto toma las imágenes de los tomogramas de Scheimpflug del Pentacam y brinda datos antes no vistos por la comunidad oftalmológica mundial como la correlación opacidad/transparencia en el centro y la periferia, el mismo comprende funcionalidades automáticas y semiautomáticas orientadas solo al uso de imágenes de tomogramas del Pentacam, lo cual hace más intuitivo la interfaz actual y la focaliza en su propósito principal. Con la presente investigación se pretende que el software PANDOC sea capaz de detectar la OCP no solo en las imágenes del Pentacam, sino que también lo realice con las imágenes oblicuas provenientes de la lámpara de hendidura. PANDOC es un software que forma parte del grupo de investigación que engloba a esta investigación por lo que es de suma importancia su fortalecimiento para el crecimiento del grupo (Cancio, 2014).

Funcionalidades del PANDOC (Cancio, 2014).

1. Extraer de manera automática la porción relevante de las imágenes brindadas por los tomogramas del Pentacam. Esta funcionalidad permite abstraer al especialista de realizar dicha operación manualmente según su experiencia.

2. Calcular el grado de la OCP sin incluir los píxeles del fondo. Esta funcionalidad permite que los píxeles de fondo no influyan de forma negativa en la obtención de la intensidad media de los píxeles.
3. Extracción de la intensidad media de píxeles de áreas irregulares de la imagen. Esta funcionalidad amplía la flexibilidad de la aplicación y le da una mayor maniobrabilidad al usuario a la hora de extraer la información que sea relevante al especialista.
4. Mostrar al usuario el porcentaje del área que fue seleccionada considerándose opacidad. Con esta funcionalidad se muestra una variación del color de fondo en las zonas que se consideran opacidad.
5. Determinar el centro de la estructura ocular, la cual mediante la identificación de variación de colores en los píxeles y trilateración celular, define automáticamente el centro de la estructura para funcionalidades posteriores.
6. Calcular el valor de la OCP. Con esta funcionalidad se calcula el valor de la OCP teniendo en cuenta la intensidad de píxeles de esa área multiplicado por el porcentaje de opacidad de la misma, esta expresa dicho valor también con respecto al centro y a la periferia de la estructura ocular.

1.4.6 Metodología de desarrollo

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 (Sánchez Mendoza, 2004).

Según la filosofía de desarrollo se pueden clasificar las metodologías en dos grupos. Las metodologías 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 (Juristo, Moreno, & Vegas, 2006).

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 (Joskowicz, 2008) (Sánchez Mendoza, 2004), 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 (Sánchez Mendoza, 2004):

- 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 (Bustamante & Rodriguez, 2014):

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.

Codificación: 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.

1.4.7 Patrones de diseño de software

Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Sus soluciones están basadas en los problemas del diseño que se repiten y que se presentan en situaciones particulares (Mühlrad, 2008).

Un patrón es un conjunto de información que proporciona respuesta a un conjunto de problemas similares. Para ello se aíslan los aspectos comunes y su solución y se añaden cuantos comentarios y ejemplos sean oportunos. Los patrones ayudan a capturar conocimiento y a crear un vocabulario técnico, hacen el diseño orientado a objetos más flexible, elegante y en algunos casos reusable (Gamma, Helm, Johnson, & Vlissides, 1995).

Entre los patrones de diseño se pueden mencionar los patrones de asignación de responsabilidades GRASP (patrones generales de software para asignación de responsabilidades, siglas de General Responsibility Assignment Software Patterns) y los patrones GOF (siglas de Gang of Four) que es el nombre con el que se conoce comúnmente a los autores del libro Design Patterns. Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos. Dentro de este grupo de patrones se encuentran los siguientes: experto, creador, bajo acoplamiento, alta cohesión, controlador, fabricación pura, indirección, variaciones protegidas, no hables con extraños y polimorfismo (Gamma, Helm, Johnson, & Vlissides, 1995).

1.5 Conclusiones del capítulo

En el capítulo se puede arribar a las siguientes conclusiones parciales:

- Después del estudio se define hacia donde está orientada la problemática de la investigación, definiéndose que el PANDOC no resuelve el problema de identificar la opacidad de la cápsula posterior en imágenes oblicuas; determinándose la necesidad de la realización de un algoritmo para la detección y realce de bordes.
- Con el estudio del procesamiento digital de imágenes, se da solución al problema empleándose diferentes técnicas de preprocesamiento y segmentación.
- Se definen las herramientas y tecnologías como Matlab, IDE Netbeans y lenguaje de programación Java, puesto que es el lenguaje en el que está implementado el software PANDOC.
- Se define la metodología XP por las bondades que brinda al proceso de desarrollo y por ser la definida por el grupo de desarrollo (AIRI).

Capítulo 2. Propuesta de solución

En el capítulo se expone la propuesta de solución para detectar y realzar los bordes en imágenes oblicuas provenientes de la lámpara de hendidura, se describe la fase inicial de la metodología XP: planificación y se obtienen los artefactos importantes de esta fases, tales como, las Historias de Usuarios, Plan de Iteraciones, Plan de Duración de Iteraciones y Plan de Entregas.

2.1 Descripción de la selección de las técnicas

En este epígrafe se muestra una imagen original utilizada para la investigación y los métodos de ajuste de contraste aplicados a la misma, para obtener de manera visual cuál es la técnica más apropiada para ajustar el contraste de la imagen.

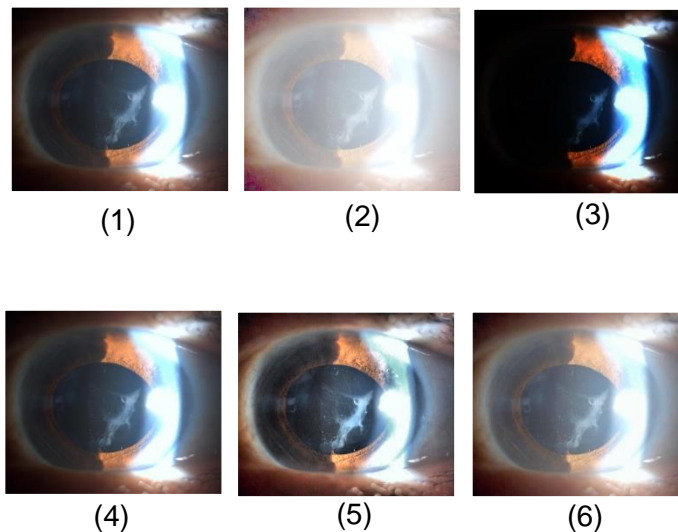


Ilustración 13. Técnicas de ajuste de contraste empleadas

- 1: Imagen oblicua original.
- 2: Ajuste de contraste usando la transformación Logarítmica.
- 3: Ajuste de contraste usando la transformación de Potencia.
- 4: Ajuste de contraste usando ecualización del histograma.
- 5: Ajuste de contraste usando ecualización de histograma adaptativo.
- 6: Ajuste de contraste usando primeramente transformación logarítmica y luego transformación de potencia.

Tras probar con diferentes métodos se determinó, que la técnica que mejores resultados arrojó, es ecualización del histograma adaptativo, pues se observa que las áreas oscuras de la imagen alcanzan mayores valores de intensidad, sin enfatizar las regiones

de alta intensidad, observándose en los restantes casos que la imagen es muy oscura o muy clara.

Luego se lleva a cabo un estudio de las técnicas de filtrado, para distinguir cuales de ellas muestran mejores resultados, en correspondencia con los resultados arrojados, será la técnica que se selecciona, a continuación en la ilustración 14 se muestra la imagen original con distintos filtros de suavizado y realce:

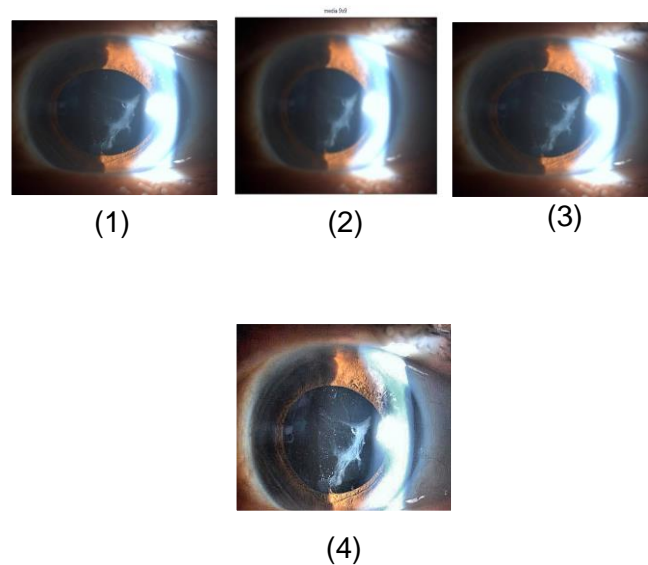


Ilustración 14. Técnicas de filtrado aplicadas sobre imágenes oblicuas

- 1: Imagen oblicua original.
- 2: Técnica de filtrado de la Media.
- 3: Técnica de filtrado de la Mediana.
- 4: Técnica de mejoramiento de bordes filtro Laplaciano paso alto.

Se aplicaron todas estas pruebas sobre diferentes imágenes oblicuas identificándose como mejores resultados visuales las técnicas de filtrado de la Mediana y el filtro Laplaciano paso alto, pues a pesar que el filtro Laplaciano define bien los bordes dentro de las imágenes también introduce ruido en ellas, por lo que se decidió usar la mediana que elimina el ruido de la misma.

En cuanto a la segmentación se realizaron diversas pruebas de las imágenes con la transformada circular de Hough se muestra en la ilustración 15, el cual se utiliza ampliamente en la bibliografía para el tratamiento de imágenes médicas, en cuanto a definición de iris y pupila; en este caso se utilizará para definir las región dela pupila que es donde se encuentra la complicación postoperatoria, que la opacidad de la cápsula posterior, en la cual se mejoraran los bodes de la región para que la imagen en cuestión

presente la calidad que debe para que el especialista pueda ofrecer un diagnóstico más certero.

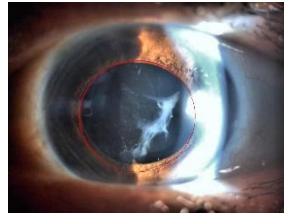


Ilustración 15. Transformada circular de hough

En la ilustración 16 se muestra el resultado de la aplicación del algoritmo por cada de una de sus etapas, con unas de las imágenes pasando por la etapa del preprocesado y segmentación, con cada una de las técnicas de ajuste de contraste y de filtrado seleccionadas.

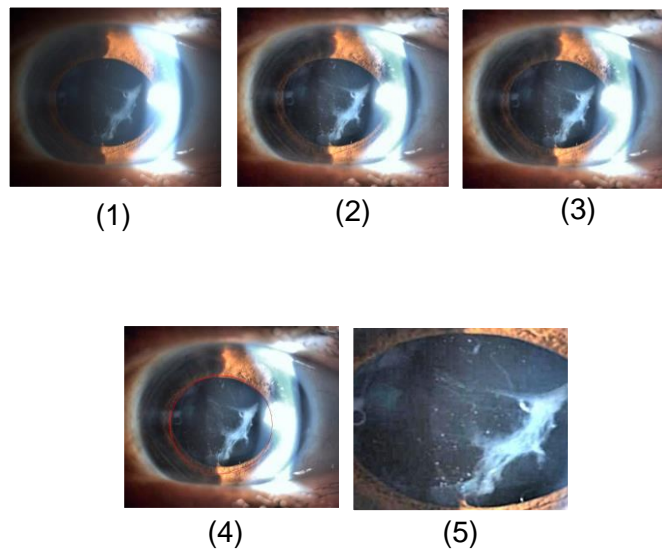


Ilustración 16. Técnicas de ajuste de contraste empleadas

- 1: Imagen oblicua original.
- 2: Ajuste de contraste usando Ecuación de Histograma Adaptativo.
- 3: Aplicación del filtro de la Mediana.
- 4: Aplicación de los métodos de segmentación usando la Transformada circular de Hough.
- 5: Aplicación de técnica de filtrado digital para la mejora de bordes como es el filtro Laplaciano paso alto.

Destacar que cada uno de estos métodos fueron probados por la herramienta Matlab la cual fue la que generó los resultados mostrados anteriormente, lo facilitó el trabajo en

cuanto a la selección de la técnicas de ajuste y filtrado, y definición del área de interés lo que fue realizado con la transformada circular de Hough.

2.2 Descripción de la solución

Para establecer la propuesta de solución se seleccionaron diferentes métodos de procesamiento de imágenes, los cuales se encuentran contenidos en las etapas de preprocesamiento y segmentación, con el objetivo de detectar y realzar los bordes de las imágenes oblicuas, de tal forma que permita al especialista una visión más amplia y con más calidad para los pacientes operados de cataratas. Se seleccionó una secuencia de pasos, que le permitirá al especialista obtener resultados más precisos en cuanto al diagnóstico de la OCP. A continuación se muestra un diagrama de actividades que modela la secuencia de pasos del algoritmo.

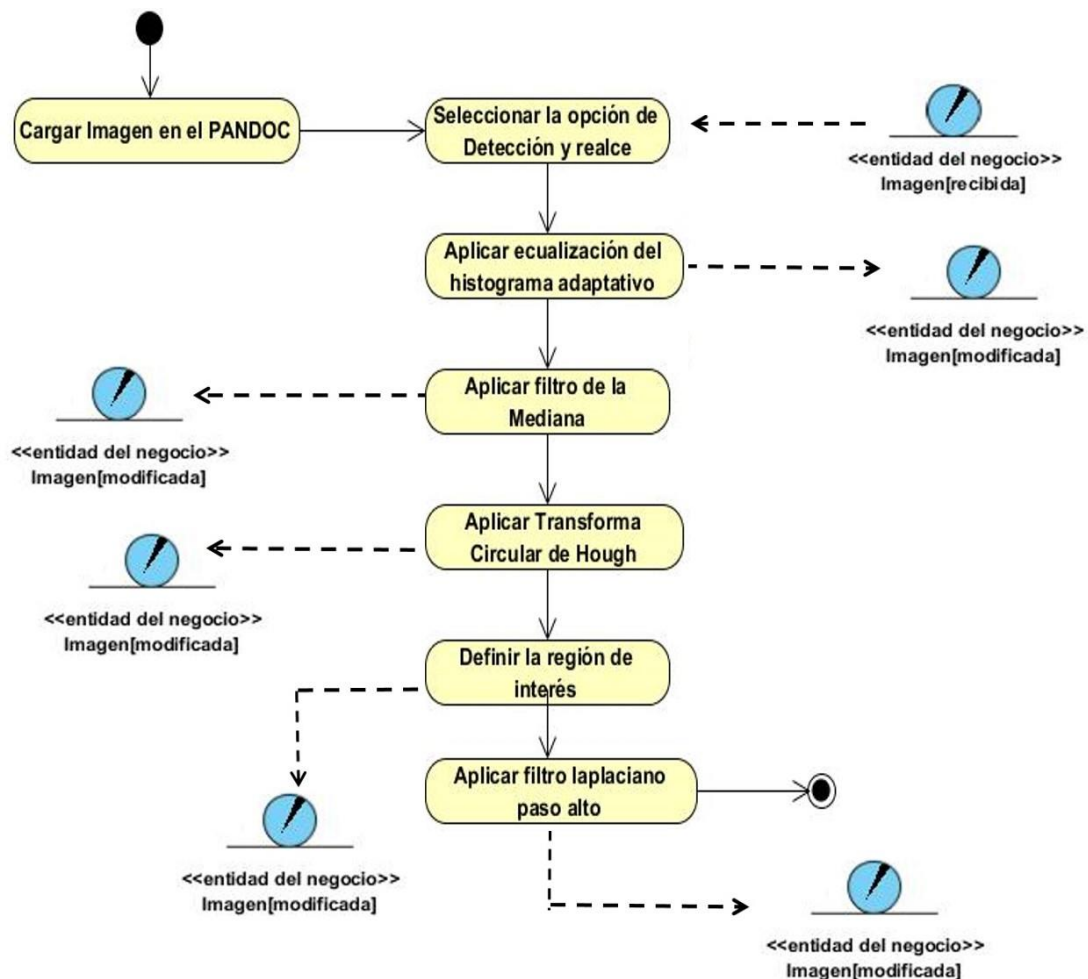


Ilustración 17. Propuesta de solución representada en un diagrama de actividades

Paso 1. Ecualización del Histograma Adaptativo

Dentro de Procesamiento digital de imagen existen varias etapas por las que debe transcurrir la imagen para su tratamiento con cualquier tipo de fin. En este caso en la etapa de preprocesamiento, se ha identificado que luego del estudio de las técnicas de

ajuste de contraste, referidas en el capítulo 1, las que mejor resultados brinda según las características propias de la imagen, es la ecualización del histograma adaptativo dado que al operar de forma local en la imagen, aumenta el contraste en las regiones de baja intensidad de la imagen, evitando enfatizar el contraste en zonas de alta intensidad, puesto que según el parámetro especificado, es la vecindad de píxeles a la que se le aplicará la ecualización, regulando así el contraste en esa región.

Paso 2. Técnica de filtrado de la Mediana

Dentro de la etapa del preprocesamiento se encuentran las técnicas de filtrado digital las cuales dada una imagen, permite acentuar o disminuir características específicas. En este caso luego de aplicar la ecualización del histograma adaptativo, se aplicará el filtro de la mediana el cual funciona de manera muy sencilla, al píxel central del núcleo le asigna el valor de la mediana de los píxeles vecinos de dicho núcleo y así va operando por toda la matriz de píxeles, lo cual permite disminuir el ruido presente en la imagen, sin eliminar los bordes o pérdida de detalles, lo cual facilita poder obtener al final del procesamiento más información que previo a los tratamientos digitales, no se podían observar.

Paso 3. Transformada Circular de Hough

Se utiliza específicamente la transformada Circular de Hough, mediante parámetros como el alto, ancho y el radio, con los cuales esta transformada identificará posibles familias de circunferencias dentro de la imagen, mediante un sistema de conteo en el cual para cada píxel existe una familia de circunferencias que pasan por él y se escoge la circunferencia que más puntos de contorno tenga. En este caso se utilizará para definir la región de interés donde se encuentra la opacidad de la cápsula posterior (OCP).

Paso 4. Técnica de Mejoramiento de Bordes: filtro Laplaciano paso alto.

En el paso anterior del algoritmo propuesto, se define el área de interés usando el método de segmentación se a transformada circular de Hough, que se usa para detectar la región de la pupila a la cual se aplica el filtro de mejoramiento de bordes, Laplaciano paso alto el cual no va a proporcionar un realce de los bordes, para una mejor observación de la opacidad capsular, se aplica el filtro a esa área puesto que es el área que interesa dentro de la imagen, también reduciendo el área de aplicación del algoritmo se reduce la matriz de intensidades, por lo que resultará computacionalmente menos costoso.

2.3 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.

2.3.1 Historias de Usuario

Las historias de usuario son utilizadas como herramientas para dar a conocer los requerimientos del sistema, las mismas son pequeños textos en el que el cliente describe una actividad del sistema. Las historias de usuario solo muestran la silueta de una tarea o requisito (Suaza, 2015). Estimar el tiempo que tomara para realizar las entregas, en cada entrega se puede desarrollar una o varias historias de usuarios en dependencia de la complejidad de las mismas.

Las buenas historias de usuario deben (Hernandez & Machado, 2010) (Beck, 2000):

- Ser entendibles por el cliente
- Entregar valor al cliente
- Formarse de una o más frases que describan algo importante para el cliente
- Mientras más corta mejor
- Ser independientes una de las otras

La tabla 1 muestra la historia de usuario de “Ajustar el contraste usando ecualización del histograma” la cual pertenece a la iteración número 1 con un tiempo estimado de 2 semanas para su realización, la misma presenta un prioridad alta, puesto que es fundamental para una correcta ejecución del algoritmo que da solución al problema identificado.

Tabla 1. Historia de usuario # 1

Historia de usuario	
Número: 1	Nombre: Ajustar el contraste usando ecualización del histograma.
Iteración asignada: 1	
Prioridad en negocio: Alta (Alta / Media / Baja)	Tiempo estimado: 2 semanas

Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Programador responsable: Jeidy Rosell Aldana
Descripción: Ajustar el contraste de la imagen trabajando específicamente con el histograma, la cual consiste en lograr la uniformidad del grado de luminancia de la imagen, obteniendo como resultado de esta funcionalidad que la imagen se logrará observar mejor en las áreas más oscuras, permitiendo observar sus detalles.	
Observaciones:	

2.3.2 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 2 se muestran cada una de las historias de usuario así como la estimación del tiempo en que se llevará a cabo su cumplimiento, identificándose como las tareas de máxima duración: “Ajustar el contraste usando ecualización de histograma adaptativo” y “Integrar al PANDOC”, obteniéndose una duración total estimada de 18 semanas.

Tabla 2. Puntos de estimación por historias de usuario

No	Historias de usuario	Puntos de estimación(semanas)
1.	Ajustar el contraste usando ecualización de histograma.	2
2.	Ajustar el contraste usando ecualización de histograma adaptativo.	3
3.	Aplicar técnica de filtrado espacial no lineal Mediana.	2
4.	Segmentar usando la Transformada de Hough Circular.	2
5.	Definir región de interés.	2
6.	Mejorar los bordes de la imagen aplicando el Laplaciano de paso alto.	2
7.	Guardar imagen.	1
8.	Salir.	1

9.	Integrar al PANDOC.	3
----	---------------------	---

2.3.3 Requisitos No Funcionales

Los requerimientos se deben descubrir antes de empezar a construir un producto, y puede ser algo que el producto debe hacer o una cualidad que el producto debe tener. Un requerimiento existe ya sea porque el tipo de producto demanda ciertas funciones o cualidades, o porque el cliente quiere que ese requerimiento sea parte del producto final. Así que si no se tienen los requerimientos correctos, no se puede diseñar o construir el producto correcto y, consecuentemente, el producto no permitirá a los usuarios finales realizar su trabajo.

Requisitos de software: tener instalada la maquia virtual de java.

Requisitos de hardware: las capacidades del cómputo deben ser:

1. 512 MB de memoria RAM
2. Procesador Intel(R) Celeron(R) 2.6 GHz

2.3.4 Plan de iteraciones

Luego de definir las historias de usuarios e identificar el tiempo y el esfuerzo para el desarrollo de cada una de ellas, se propone un plan de iteraciones donde las historias de usuario están contenidas. El plan se propone, regulando el rango establecido para cada una de ellas; teniendo en cuenta todos los aspectos antes tratados, se pretende realizar el desarrollo en 3 iteraciones, referidas a continuación:

Iteración 1

La iteración tiene como finalidad ajustar el contraste de la imagen oblicua mediante algunas técnicas de puntos, para la eliminación del ruido y aclarar las partes oscuras de la imagen.

Iteración 2

Esta iteración se encarga de aplicar la técnica de filtrado para el suavizado de la imagen y se ocupa de la segmentación para definir el área de interés usando la transformada circular de Hough.

Iteración 3

En esta iteración luego se define la región de interés de la imagen oblicua, es decir definir la región en que se encuentra la Opacidad de la Cápsula Posterior (OCP), con dichas historias de usuarios también se aplican técnicas de mejora de borde y luego se procede a la integración con software PANDOC.

Tabla 3. Plan de duración de las iteraciones

Iteraciones	Historias de usuario	Duración
Iteración 1	Ajustar el contraste usando ecualización de histograma.	5
	Ajustar el contraste usando ecualización de histograma adaptativo.	
Iteración 2	Aplicar técnica de filtrado espacial no lineal Mediana.	6
	Segmentar usando la Transformada de Hough Circular.	
	Definir región de interés.	
Iteración 3	Mejorar los bordes de la imagen aplicando el Laplaciano de paso alto.	7
	Guardar imagen.	
	Salir.	
	Integrar al software PANDOC	

2.3.5 Plan de entrega

El plan de entregas tiene como objetivo definir el número de liberaciones que se realizarán en el transcurso del proyecto y las iteraciones que se requieren para desarrollar cada una.

Tabla 4. Plan de entrega

	Iteración 1	Iteración 2	Iteración 3
Cantidad de HU	2	1	5

Fecha de entrega	4/03/2016	20/03/2016	30/05/2016
------------------	-----------	------------	------------

2.4 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 e implementable que a la larga costará menos tiempo y esfuerzo desarrollar. Define que se debe utilizar una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código (Bustamante & Rodriguez, 2014) (Beck, 1999).

2.4.1 Tarjetas CRC

La metodología XP en lugar de utilizar diagramas para desarrollar modelos representa las clases mediante tarjetas. Las tarjetas CRC (Clase, Responsabilidad y Colaboración) ayudan a definir actividades durante el diseño del sistema.

Estas tarjetas representan una entidad del sistema, a la cual asignar responsabilidades y colaboraciones. El formato físico de las tarjetas CRC facilita la interacción entre los participantes del proyecto, en sesiones en las que se aplican técnicas de grupos como 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 (Casas & Reinaga, 2009).

Estas tarjetas permiten trabajar con una metodología basada en objetos, permitiendo que el equipo de desarrollo completo contribuya en la tarea del diseño. El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente.

A continuación se muestra en la tabla 5, una de las tarjetas CRC de la clase `cc_Principal` que presenta como responsabilidades 3 métodos y 7 clases que colaboradoras como son: `imagenAPI`, `clahe`, `mediana`, `galeria`, `ventanaAux`, `laplaciano` y `filtroCircularHough`.

Tabla 5. Tarjeta CRC de la clase cc_Principal

cc_Principal	
Descripción: Se encarga de controlar el flujo del procesamiento del algoritmo propuesto.	
Atributes:	
Nombre	Descripción
BufferedImage img1	contiene la imagen cargada desde el archivo
LinkedList<ImagenAPI> listadeImágenes	lista de imágenes procesadas
LinkedList<ImagenAPI> listadeImágenesCopia	lista de imágenes a procesar
BufferedImage imgSalida	contiene la imagen preprocesada
clahe filtroClahe	objeto de la clase
mediana med	objeto de la clase mediana
laplaciano filtroLaplacian	objeto de la clase laplaciano
Responsabilidades:	
Nombre	Collaborator
procesar()	imagenAPI
reajustar()	clahe
getImgSalida()	mediana
	galería
	ventanaAux
	laplaciano
	filtroCirculoHough

2.4.2 Patrones de diseño utilizados

Experto: Es un patrón que se usa más que cualquier otro al asignar responsabilidades y expresa simplemente la "intuición" de que los objetos realizan acciones relacionadas con la información que poseen (Mühlrad, 2008).

- Este patrón se utiliza en diferentes clases, que contienen información y varias funcionalidades tal como: suavizado de la imagen en la clase mediana y detección de la región de interés en la clase transformaCircularHough.

Creador: El patrón creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debe conectar con el objeto producido en cualquier evento (Mühlrad, 2008).

- Dentro de la aplicación este patrón se observa en las acciones del controlador cc_Principal, en la cual se crean objetos para el procesamiento de la imagen.

```
galeria vAux8 = new galeria(listadeImágenes, listadeImágenesCopia, "Detección y Realce.");
filtroCircularHough fch = new filtroCircularHough();
```

Controlador: Mantiene el control actuando como intermediario entre una determinada interfaz y el algoritmo que la implementa, además permite dividir los eventos del sistema

en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento (Larman, 2012).

- Este patrón es utilizado en las clases, donde cada una tiene las responsabilidades moderadas y colabora con las otras para llevar a cabo las tareas, ejemplo de ello la clase ImagenAPI.

2.4.3 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. 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, establezca un estándar de codificación para asegurarse de que se trabaje de forma coordinada. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente (Garcia, 2011).

- Utiliza los comentarios para explicar qué hace el código.
- Usa nombres significativos para las clases, métodos, propiedades y variables.
- Los métodos de clases, se define que la primera letra debe ser minúscula, si son varias palabras se debe de intercalar entre minúsculas y mayúsculas.
- Se define una declaración por línea.
- Se evita asignar el mismo valor a más de una variable en la misma sentencia, pues dificulta su lectura.

2.4 Conclusiones del capítulo

En este capítulo se definen las siguientes conclusiones parciales:

- Mediante cada una de las técnicas investigadas, se identifican con mejores resultados, la técnica de ecualización del histograma adaptativo, la Mediana, el filtro Laplaciano paso alto y como técnica de segmentación la transformada circular de Hough.
- Con descripción de la propuesta de solución, en cada una de sus etapas, se demuestra el motivo de la selección de cada una de las técnicas empleadas, en las etapas del preprocesamiento y la segmentación de imágenes.
- El desarrollo de las dos primeras etapas, de la metodología XP, genera cada uno de sus artefactos como las historias de usuario, la iteración donde se va a ejecutar cada una, y el plan de entrega, generando un total de 9 historias de usuario, que describen las funcionalidades que se llevarán a cabo.

- En el diseño de la solución, con la elaboración de las tarjetas CRC se logra identificar cuáles de las clases se utilizan de soporte y se definen diferentes estándares, para garantizar la uniformidad del código y se emplean diferentes patrones de asignación de responsabilidades.

Capítulo 3. Resultados y validación del sistema

En el capítulo se generan las pruebas y validaciones del algoritmo las cuales constituyen un instrumento para comprobar el nivel de calidad de un producto. La metodología divide las pruebas en dos grupos: pruebas unitarias, encargadas de verificar el código, la cual es realizada por los programadores, como también plantea las pruebas de aceptación destinadas a evaluar si al terminar una iteración se consiguió la funcionalidad requerida diseñadas por el cliente. Durante las iteraciones las historias de usuarios seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente.

3.1 Fase de Implementación

En esta iteración se implementaron las funcionalidades que obtuvieron los primeros resultados, relacionadas con las técnicas de ajuste de contraste de la imagen para su preparación para otros tratamientos digitales.

En la tabla 6 se especifican las historias de usuario de la iteración número 1, las cuales pertenecen a las técnicas de ajuste de contraste, las cuales nos demuestra que la estimación del tiempo, realmente fue la empleada como se observa a continuación.

Tabla 6. Tiempo de implementación de las HU de la iteración #1

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Ajustar el contraste usando ecualización del histograma	2	2
Ajustar el contraste usando ecualización de histograma adaptativo	3	3

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

Tabla 7. Tarea de ingeniería de la funcionalidad Ecualización del Histograma.

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre: Ajustar el contraste usando ecualización del histograma.	
Tipo de tarea: desarrollo	Puntos de estimación: 2

Programador responsable: Jeidy Rosell Aldana
Descripción: Se realiza una ecualización del histograma para mejorar la calidad visual de la misma.

Tabla 8. Tarea de ingeniería de la funcionalidad Ecualización del Histograma Adaptativo

Tarea	
Número de tarea: 2	Número de HU: 2
Nombre: Ajustar el contraste usando ecualización de histograma adaptativo.	
Tipo de tarea: desarrollo	Puntos de estimación: 3
Programador responsable: Jeidy Rosell Aldana	
Descripción: Se realiza una ecualización de la imagen en vez de forma uniforme en zonas específicas de la imagen, para no resaltar el grado luminosidad de algunas regiones de la imagen.	

Iteración 2

En esta iteración se implementa el filtro de la mediana y se define la región de interés. Se muestran los resultados reales del tiempo empleado en realizar una funcionalidad que la mayoría de los casos corresponden con las estimación realizada.

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

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Aplicar técnica de filtrado espacial no lineal Mediana.	2	2
Segmentar usando la Transformada de Hough Circular.	2	2
Definir región de interés.	2	2

Tabla 10. Tarea de ingeniería de la implementación del filtro de la mediana

Tarea

Número de tarea: 3	Número de HU: 3
Nombre: Aplicar técnica de filtrado espacial no lineal Mediana.	
Tipo de tarea: desarrollo	Puntos de estimación: 2
Programador responsable: Jeidy Rosell Aldana	
Descripción: Se enmarca en eliminar el ruido contenido en la imagen.	

Tabla 11. Tarea de ingeniería de la implementación del algoritmo la Transformada de Hough.

Tarea	
Número de tarea: 4	Número de HU: 4
Nombre: Segmentar usando la Transformada de Hough Circular.	
Tipo de tarea: desarrollo	Puntos de estimación: 2
Programador responsable: Jeidy Rosell Aldana	
Descripción: Este algoritmo es el encargado de identificar la región en la que se encuentra la opacidad de la cápsula posterior.	

Tabla 12. Tarea de ingeniería de la funcionalidad Definir región de interés.

Tarea	
Número de tarea: 5	Número de HU: 5
Nombre: Definir región de interés.	
Tipo de tarea: desarrollo	Puntos de estimación: 1
Programador responsable: Jeidy Rosell Aldana	
Descripción: Se identifican los bordes de la pupila	

Iteración 3

Es esta fase del desarrollo se realzan los bordes de la imagen oblicua, realiza la funcionalidad guardar, salir y se integra al software PANDOC, a continuación en la tabla 13 se aprecia cada una de las estimaciones de tiempos de realización.

Tabla13. Tiempo de implementación de las HU de la iteración # 3

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Mejorar los bordes de la imagen aplicando el Laplaciano de paso alto.	2	2
Guardar imagen.	1	1
Salir.	1	1
Integración al PANDOC.	3	3

Tabla 14. Tarea de ingeniería de la funcionalidad Filtro de Mejora de bordes: Laplacian.

Tarea	
Número de tarea: 6	Número de HU: 6
Nombre: Mejorar los bordes de la imagen aplicando el Laplaciano de paso alto.	
Tipo de tarea: desarrollo	Puntos de estimación: 3
Programador responsable: Jeidy Rosell Aldana	
Descripción: Mejora los bordes de la imagen.	

Tabla 15. Tarea de ingeniería de la funcionalidad Guardar

Tarea	
Número de tarea: 7	Número de HU: 7
Nombre: Guardar.	
Tipo de tarea: desarrollo	Puntos de estimación: 1
Programador responsable: Jeidy Rosell Aldana	

Descripción: Almacena la imagen preprocesada.

Tabla 16. Tarea de ingeniería de la funcionalidad Salir

Tarea	
Número de tarea: 8	Número de HU: 8
Nombre: Salir.	
Tipo de tarea: desarrollo	Puntos de estimación: 1
Programador responsable: Jeidy Rosell Aldana	
Descripción: Salir de la aplicación.	

Tabla 17. Tarea de ingeniería de la funcionalidad Integración al PANDOC

Tarea	
Número de tarea: 7	Número de HU: 7
Nombre: Integración al PANDOC.	
Tipo de tarea: desarrollo	Puntos de estimación: 3
Programador responsable: Jeidy Rosell Aldana	
Descripción: Se lleva a cabo la integración del algoritmo al software PANDOC.	

3.2 Resultados de aplicar el algoritmo

El algoritmo probado en 11 imágenes oblicuas, provenientes de pacientes que han sido operados de cataratas y poseen opacidad en la cápsula posterior. A continuación se muestran los resultados obtenidos de aplicar este algoritmo a tres imágenes provenientes de estos pacientes.

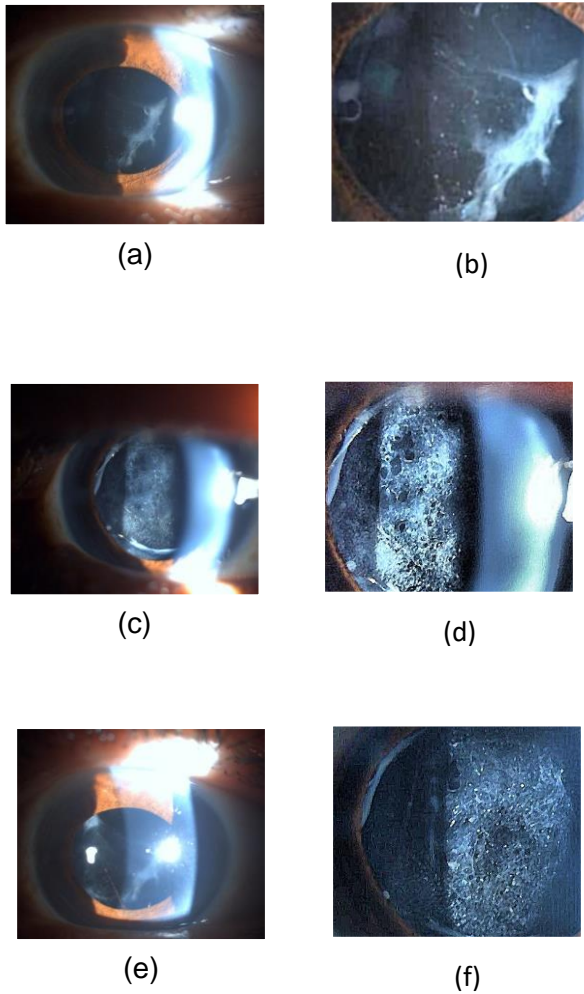


Ilustración 18. Resultados del preprocesado de tres imágenes oblicuas.

- a, c y e imágenes oblicuas originales de diferente pacientes.
- b, d y f imágenes preprocesadas.

3.3 Pruebas

El proceso de pruebas es sumamente importante para todos los negocios de cualquier tipo de industria. Mediante el mismo, se confía que se entregue un producto que sea completamente funcional y que cumpla con las normas y métricas de calidad establecidas. En el mundo del software no es diferente, ya que las pruebas juegan un rol decisivo en la confección de un producto software, tanto así que se han desarrollado nuevos paradigmas con respecto al proceso de desarrollo de software es el caso del Desarrollo Conducido por Pruebas (Test Driven Development).

3.4 Prueba de Caja Blanca

El método del camino básico permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez, dicha técnica se aplicará al método `detectarCirculos()` de la clase `TransformadaCircularHough`, a continuación se mostrara el fragmento de código al que se le aplicará la técnica del camino básico y se le calculará la complejidad ciclomática:

```
public void detectarCirculos(int[] imagen, int anchoI, int altoI, double radioC) {
    this.radio = radioC;
    int amax = (int) Math.round(8F * radioC);
    int[][] a = new int[2][amax];
    int i = 0;
    for (int j = 0; j < amax; j++) {
        double cita = (Math.PI * 2 * (double) j) / (double) amax;
        int roJ = (int) Math.round((double) radioC * Math.cos(cita));
        int citaJ = (int) Math.round((double) radioC * Math.sin(cita));
        if ((i == 0) || (roJ != a[0][i]) & (citaJ != a[1][i])) {
            a[0][i] = roJ;
            a[1][i] = citaJ;
            i++;
        }
    }
    amax = i;
    calcular(a, amax, imagen, anchoI);
}
```

Ilustración 19. Código fuente del método `detectarCirculos()`

Partiendo del fragmento de código tomado se obtiene el siguiente grafo de flujo.

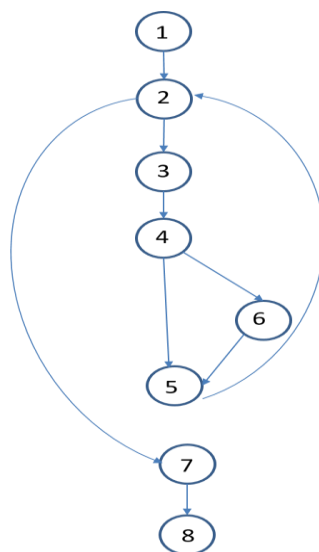


Ilustración 20. Grafo de flujo asociado al método `detectarCirculos()` de la clase `transformadaCircularHoug`

Partiendo del diseño del grafo como se muestra en la ilustración 20, se pasa a calcular la complejidad ciclomática, por medio de la misma se determinan 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.

Complejidad ciclomática

$$V(G) = \text{Regiones} = 3$$

$$V(G) = \text{Aristas (A)} - \text{Nodos (N)} + 2$$

$$V(G) = 9 - 8 + 2$$

$$V(G) = 3$$

$$V(G) = \text{Nodos Predicado} + 1 = 2 + 1 = 3$$

El valor $V(G)$ expresa la cantidad de caminos linealmente independientes de la estructura de control del programa, por lo que se definen los siguientes 3 caminos:

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

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

Camino básico 3: 1->2->7->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 18. Caso de prueba de caja blanca para el camino básico 1

Entrada	La imagen con la cual se va a trabajar con sus respectivas dimensiones y el radio para determinar el tamaño de la circunferencia.
Resultados Esperados	Cálculo del píxel por el cual podría pasar la familia de circunferencia más grande.
Condiciones	Se debe haber cargado previamente una imagen en el software.

3.5 Pruebas de Caja Negra

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos de prueba en que el módulo no se atiene a su especificación.

Esto se refiere a que se llevan a cabo para verificar el ajuste del sistema con los requerimientos determinados. Además se enfocan especialmente en los módulos que se relacionan con la interfaz de usuario. No requieren el conocimiento de la estructura interna del programa para su puesta en marcha

3.5.2 Pruebas de Aceptación

Las pruebas de aceptación cumplen con el objetivo de indicarnos cuando las funcionalidades de una iteración han sido completadas exitosamente. La validación del software se consigue mediante una serie de pruebas de caja negra que demuestran la conformidad del cliente con los requisitos. Una vez que se procede con cada caso de prueba de validación, puede darse una de las siguientes condiciones:

1. Las características de funcionamiento o de rendimiento están de acuerdo con las especificaciones y son aceptables.
2. Se descubre una desviación de las especificaciones y se crea una lista de deficiencias.

Para la descripción de estas pruebas se presenta el diseño del caso de prueba correspondiente a la historia de usuario “Ecuilización del histograma adaptativo” que incluye funcionalidades críticas para el sistema. La siguiente tabla muestra los resultados satisfactorios obtenidos en la prueba de aceptación “Ecuilización del histograma adaptativo”.

Tabla 19. Caso de prueba para la HU 2.

Caso de prueba de aceptación
Código: HU_P1 Historia de Usuario:2
Nombre: Ajustar el contraste usando ecualización del histograma adaptativo.
Descripción: Comprueba la funcionalidad que mejora el contraste de la imagen.
Condiciones de Ejecución: Se debe cargar la imagen oblicua proveniente de la lámpara de hendidura.
Entrada/Pasos de Ejecución: <ul style="list-style-type: none"> • El usuario selecciona la imagen. • El usuario selecciona la opción de detección y realce. • El PANDOC muestra la imagen preprocesada con el ajuste de contraste en regiones específicas de a imagen.

<p>Resultado Esperado:</p> <p>El resultado es una imagen RGB que en las zonas oscuras tendrá un mejor contraste y visualización.</p>
<p>Evaluación de la Prueba: Prueba satisfactoria.</p>

Resultados de las pruebas de aceptación

En correspondencia al nivel de satisfacción del cliente, atendiendo a sus necesidades se pudo obtener a partir de cada entrega realizada, en las cuales se realizaron 3 iteraciones de las pruebas de aceptación y se detectaron no conformidades significativas referentes a errores de validación, no significativas que se centraron fundamentalmente en errores ortográficos.

Las pruebas de aceptación se realizaron en correspondencia con las historias de usuario (HU) en donde se logró identificar un total 9 no conformidades en las 3 iteraciones del producto que fueron realizadas, en la tabla y el gráfico siguiente se muestran los resultados alcanzados de cada iteración.

Tabla 20. Resultados de aceptación del cliente

Número de iteración	No conformidades		No conformidades resueltas
	Significativas	No significativas	
1	2	2	4
2	1	2	3
3	1	1	2

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

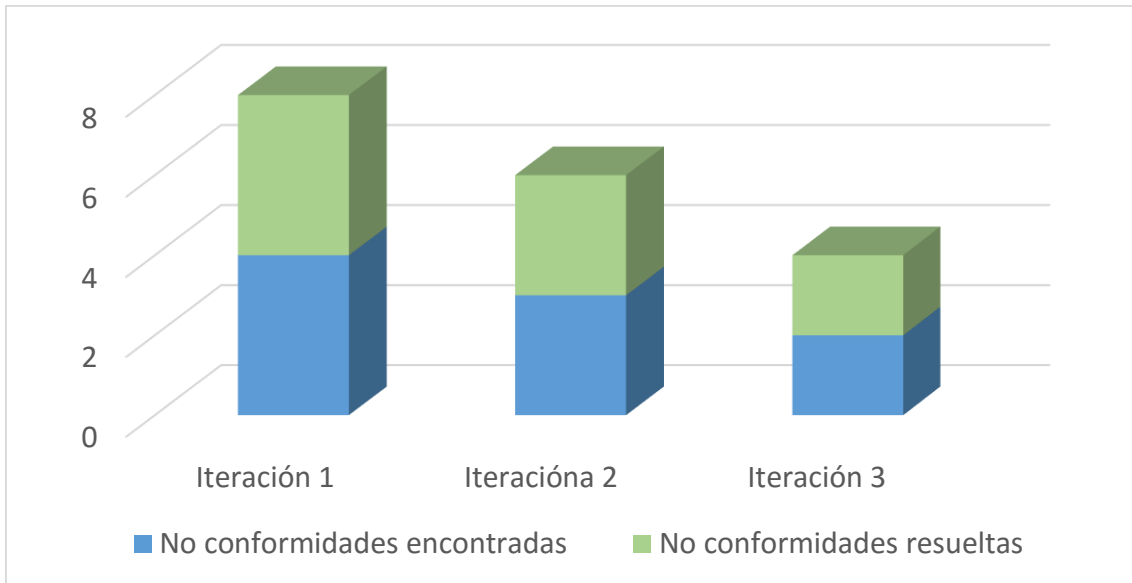


Ilustración 21. Resultados de las pruebas

Como parte de estas pruebas se procedió a la creación de 11 casos de estudio para verificar los resultados del software, en los mismos se tomaron imágenes oblicuas de diferentes pacientes, las cuales arrojaron los siguientes resultados:

- I. Caso de estudio aplicado a la imagen oblicua #1, procedente de la lámpara de hendidura.

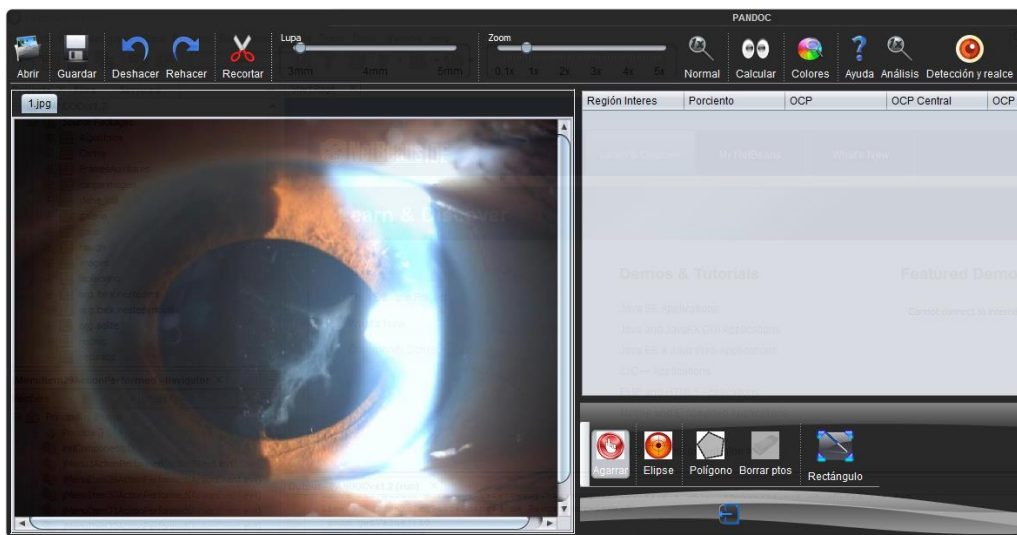


Ilustración 22. Imagen oblicua original cargada en la interfaz del software PANDOC.

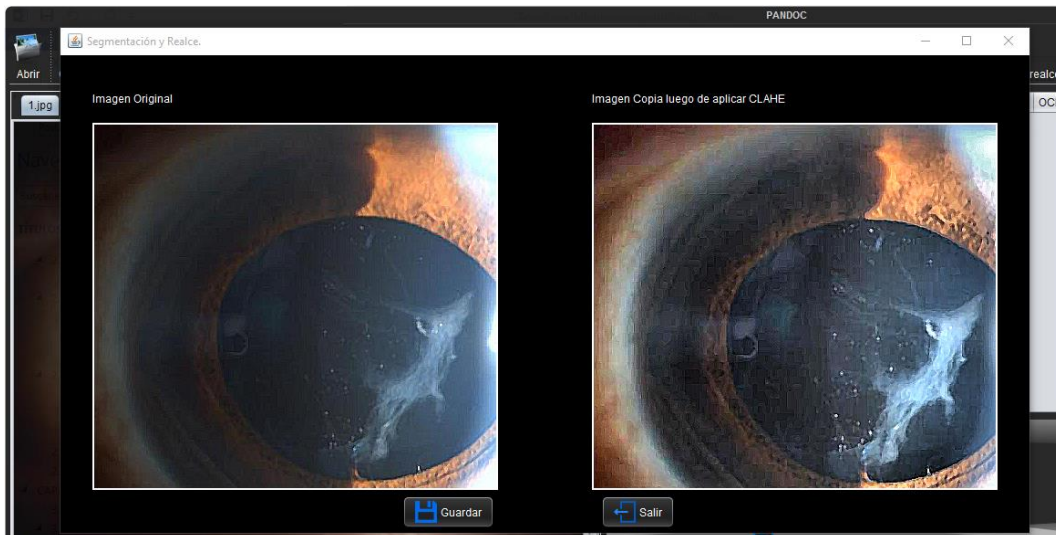


Ilustración 23. Proceso de preprocesamiento del algoritmo propuesto en este caso se muestra la imagen original y la imagen luego de aplicar ecualización del histograma adaptativo.

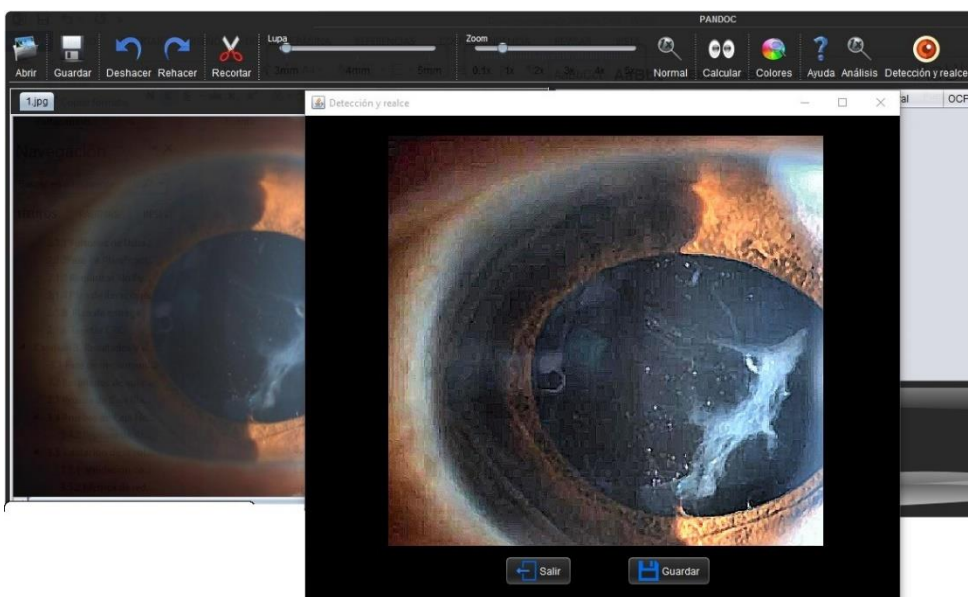


Ilustración 24. Resultado final del procesamiento aplicando como última técnica el operador Laplaciano pasó alto.

3.6 Validación de la solución

Las métricas dan una medida que cuan correcto y que calidad presenta es el resultado obtenido. A continuación se tratarán algunas de ellas y se mostrarán los resultados, los cuales son obtenidos mediante la herramienta Matlab.

3.6.1 Validación con Operador Canny

En el capítulo 1 se hace referencia a técnicas de segmentación específicamente para detección de bordes como son los operadores Canny y Sobel. En cuanto operador de Sobel es simple, pero su exactitud sufre condiciones ruidosas. Por el contrario, el detector de bordes de Canny tiene muchas características favorables, tales como efecto de suavizado para eliminar el ruido, y mejorar la relación de señal a ruido a través de un proceso conocido como supresión no máxima (Kim, 2013). Debido a que el operador de Canny solo se utiliza para las pruebas esta deficiencia, no afecta el producto por lo cual se utiliza sobre el método de Sobel.

Por lo que se decidió validar la solución con algoritmo de detección de bordes Canny debido a sus bondades, una vez seleccionado el algoritmo se procedió a identificar los bordes de la imagen original y la imagen preprocesada con dicho operador a continuación se muestran los resultados:

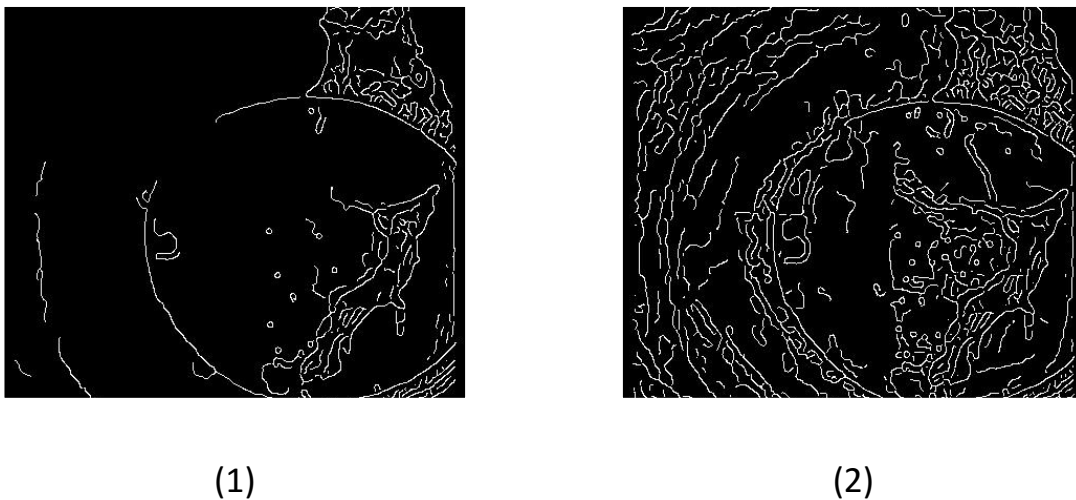


Ilustración 25. Aplicación del operador Canny

- 1: Imagen original con el operador Canny.
- 2: Imagen preprocesada con el operador Canny.

En estos resultados se puede observar la cantidad de información que contiene la imagen preprocesada, puesto que es más definida y nítida a continuación se muestran los valores de bordes promedios que logró alcanzar una con respecto a la otra:

Tabla 21: Valores de comparación de detección de Bordes

Imágenes	Cantidad de píxeles Blancos(1)	Cantidad de píxeles Blancos(1) en %
Imagen Original	8859	4.44%
Imagen Preprocesado	18658	9,35%

3.6.2 Métrica de reducción de ruido propuesta por Xiang Zhu

La métrica propuesta por Xiang Zhu, no depende de un previo conocimiento acerca de la varianza del ruido, datos experimentos reales se centraron en la optimización de parámetros para varios eliminación de ruido algoritmos utilizando la métrica la cual es en base a los gradientes locales de la imagen, como un paso intermedio para llegar a cuantificar la cantidad de desenfoque y ruido aleatorio (Zhu & Milanfar, 2010).

Dada una imagen con ruido, dividida en M parches no solapados de tamaño NxN calcular la coherencia local R_k usando la fórmula (Zhu & Milanfar, 2010):

$$R_k = \frac{s_1 - s_2}{s_1 + s_2} \quad (12)$$

Luego encontrar entre 0 y M parches determinados por un umbral, los valores de la coherencia local deben ser mayor que el umbral ($R_k > \tau$) el umbral τ está determinado por, dado un nivel de significancia dado por δ :

$$\delta = \int_{\tau}^1 f_R(r) dr = \left[- \left(\frac{1-r^2}{1+r^2} \right)^{N^2-1} \right]_{\tau}^1 = \left(\frac{1-r^2}{1+r^2} \right)^{N^2-1} \quad (13)$$

Luego se calcula la métrica determinada por la fórmula: $Q = S_1 \frac{s_1 - s_2}{s_1 + s_2}$ (14)

El valor de la métrica definido por Q muestra el resultado de la imagen en su totalidad (Zhu & Milanfar, 2010)

$$Q = (1/M) \sum_{k=1}^m Q_k$$

En la tabla 22 se muestra el resultado de aplicar la métrica de reducción de ruido anteriormente expuesta la cual fue probada en a herramienta Matlab, mediante las

funciones y las bondades que el mismo ofrece a continuación de muestra los resultados de aplicar esta métrica a la imagen original y a la imagen preprocesada, el error es más pequeño por lo que es notable que las técnicas de ajuste de contraste y filtrado digital aplicadas, han reducido el ruido en la imagen, a continuación se muestran los valores de la tasa de error por los diferentes canales de color(RGB).

Tabla 22.Comparación del margen de error entre la imagen original y la preprocesada.

	R	G	B
Imagen original	0.0172	0.0178	0.0177
Imagen resultante	0.0012	0.0011	0.0011

Se observa en la Tabla 22, que lo valores de ruido en la imagen son más pequeños por lo que se puede decir que en la imagen se podrán observar las regiones de opacidad capsular de una forma más precisa. A continuación se representan dichos valores de ruido, en el siguiente gráfico:

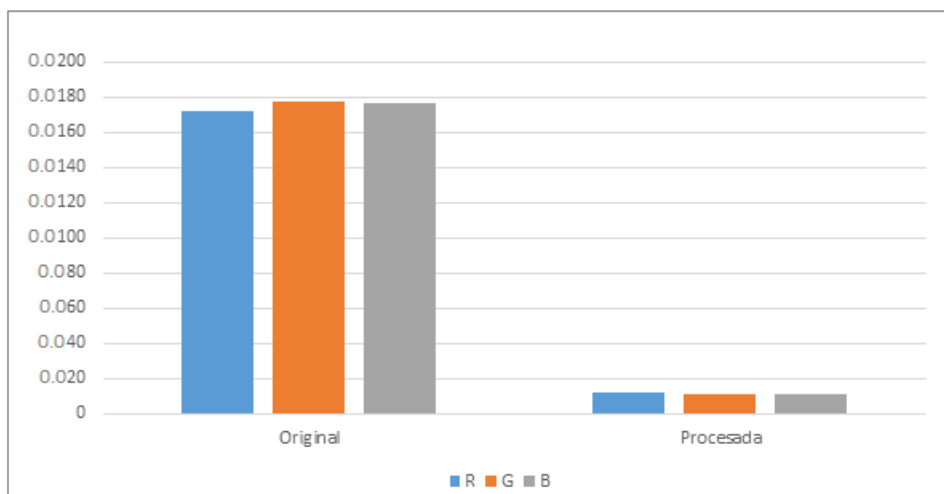


Ilustración 26. Comparación de Imágenes: Reducción de Ruido

3.6.3 Métrica basada en Diagramas de dispersión

Para determinar la calidad de la imagen, es necesario tener en cuenta la nitidez de las líneas, bordes y la suavidad en las áreas, ya que de estos atributos dependen los resultados de las operaciones de procesamiento de imágenes que se llevan a cabo (Oliva, 2014).

Como podrán observar en la ilustración 27, este método basado en diagramas de dispersión, ofrece una medida gráfica del suavizado y el agudizado en la imagen luego

de aplicar el algoritmo planteado, los píxeles que se sitúen por encima de la recta son los que han sido agudizados (de color rosa) y los que se sitúen por debajo los que han sido suavizados (de color rojo).

Se necesita que la propuesta de solución realce las imágenes oblicuas, siempre teniendo en cuenta el compromiso de suavizar la imagen sin perder bordes y características importantes dentro de la misma.

A continuación en la ilustración 27, se puede observar que por encima de la magnitud del gradiente, existe un realce de bordes, debido a la aplicación de las técnicas de realce de bordes con el operador Laplaciano como también existe un suavizado, pues se utilizó el filtro de la mediana para eliminar el ruido de la imagen, lo cual permite una mejor observación.

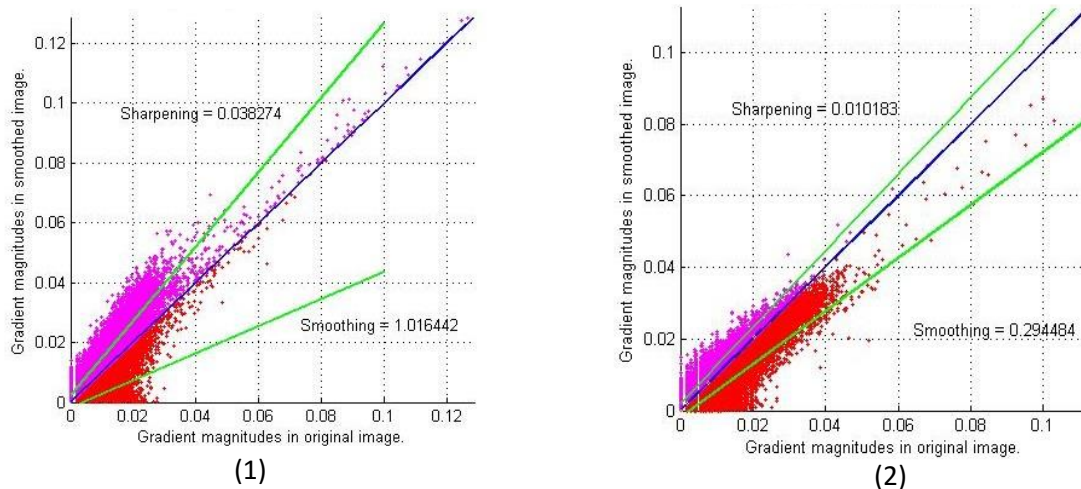


Ilustración 27. Medidas de calidad basada en diagramas de dispersión de la magnitud del gradiente. Los píxeles por encima de la curva $y = x$, son los que se agudizan (color rosa), los que se encuentran por debajo son los suavizados (color rojo).

- 1 y 2: Imágenes Preprocesadas.

3.7 Conclusiones del capítulo

Con el avance de la investigación en este capítulo se concluye que:

- La implementación de cada una de las funcionalidades demostró que la estimación del tiempo fue precisa, pues se logró terminar cada una en el tiempo estimado.

- Las métricas de validación realizadas arrojaron resultados favorables, dado que nos permiten identificar el grado de mejoría y calidad, que presenta la imagen oblicua luego de la realización algoritmo propuesto.
- Se aplican pruebas de software, las que ofrecen la medida de que se cumple con las funcionalidades que se definieron en la fase de planificación, por el cliente, brindando así un algoritmo para una mejor observación y dictamen médico de la imagen oblicua

Conclusiones generales

La realización de este trabajo responde a la necesidad de buscar una solución al problema planteado. Terminada la investigación se llegó a las siguientes conclusiones:

- A través del estado del arte se evidenció que el PANDOC no resuelve el problema planteado, fundamentándose la necesidad de desarrollar un algoritmo que detecte y realce bordes de las imágenes oblicuas provenientes de la lámpara de hendidura.
- Mediante el estudio de las técnicas de preprocesamiento, se logró definir las que mejores resultados brindan, son las técnicas de ecualización del histograma adaptativo, el filtro espacial de la Mediana y Laplaciano paso alto, en ese orden de ejecución.
- Se definió como técnica de segmentación la transformada circular de Hough, pues presenta mejores resultados para la selección de la región de interés.
- El desarrollo fue guiado por la metodología XP, lo permitió el diseño de la propuesta de solución, generándose así cada uno de los artefactos de la metodología.
- Las métricas para validar la calidad de la detección y del realce justificaron cuantitativamente la selección de la secuencia de métodos aplicados, mostrando buenos resultados en las imágenes oblicuas.
- La aplicación de pruebas de software, permitió validar que la propuesta de solución, cumple con las funcionalidades especificadas por el cliente y que se podrá usar en aras de una mejor precisión al emitir un diagnóstico de la opacidad de la cápsula posterior.

Recomendaciones

1. Para un mejor resultado visual se recomienda usar técnicas de súper-resolución.
2. En futuras versiones se implemente la cuantificación automática de la opacidad de la cápsula posterior para imágenes oblicua.

Referencias

1. Echeverri Arias, J. A., Manrique Losada, B., Javier Moreno, F., & Bravo, A. (2009). Mejoramiento de Imgenes Usando Funciones de Base Radial. Medellin.
2. Aldalur, B., & Santamaría, M. (2002). Realce de imágenes: filtrado espacial.
3. Alvarez Cancio, M., Rodríguez Puente, R., & Hernández López, I. (2013). PANDOC: software para la cuantificación objetiva de la opacidad de la cápsula posterior mediante tomogramas scheimpflug del pentacam.
4. Arntz, D. A. (2011). Curso de Oftalmología, Departamento de Oftalmología.
5. Arriaga, J. A., & Riverón, E. M. (2012). Segmentación del disco óptico en imágenes de retina mediante transformada de Hough y contornos activos.
6. Azpiroz, J. (2000). Procesamiento de imágenes biomédicas. 1ra ed, pp 169. México: Universidad Autónoma Metropolitana.
7. Beck, Kent. (1999). Extreme Programming Explained.
8. Bustamante, D., & Rodriguez, J. (2014). Metodología Actual: Metodología XP.
9. Cancio, M. A. (2014). PANDOC: Sistema basado en casos para la cuantificación objetiva de la opacidad capsular de un paciente operado de catarata.
10. Casas, S., & Reinaga, H. (2009). Aspectos tempranos : un enfoque basado en Tarjetas CRC. Argentina.
11. Cepero Ruz, R., & Cabrera Rodríguez, C. (2010). Implementación de algoritmos de reducción de ruido en las imágenes.
12. Corral, D. A. (2012). Mejora de un algoritmo de segmentación mediante contornos activos. Leganés, Madrid: UNIVERSIDAD CARLOS III DE MADRID.
13. Coto, E. (2005). Métodos de Segmentación de imágenes Médicas. Venezuela. Caracas: ISSN 1316-6239.
14. Eches Salcedo, E. A., & Chiriboga Maldonado, M. d. (2003). Implementación de algoritmos para preprocesamiento, detección de bordes y segmentación de imágenes digitales a color utilizando MATLAB.
15. Escalante, B. (2006). Procesamiento Digital de Imágenes.
16. Escalante, D. B. (2006). Procesamiento Digital de Imágenes.

17. Felipe, X., Moya, J. M., Peña, A. J., & Segarra, D. (2001). Modificación del Histograma de una imagen.
18. Fernández, O. B. (2004). Introducción al lenguaje de JAVA. Una guía básica.
19. Fiter, E. L. (2012). Descripción, comparación y ejemplos de uso de las funciones de la toolbox de procesamiento digital de imágenes. Madrid: Escuela Universitaria de Ingeniería.
20. Fuente-Bermudez, S., Orlena-Rodríguez, F. J., & González-Barbosa, J. J. (2015). Detección de masas y asimetrías en mamografía con distintos patrones parenquimales utilizando entropía de TSALLIS.
21. Gamma, E., Helm, R., J. R., & Vlissides, J. (1995). Elements of Design Patterns. Addison-Wesley.
22. Garcia, O. (22 de Agosto de 2011). Nomenclatura estándar del código Java. Obtenido de <http://www.elclubdelprogramador.com/2011/08/22/java-nomenclatura-estandar-del-codigo-java/>
23. Gómez, J. T. (2009). Análisis comparativo de técnicas de segmentación de iris. Madrid.
24. Hall, A. d. (2001). Visión por computador, fundamentos y métodos.
25. Hermsilla, T., Bermejo, E., Balaguer, A., & Ruiz, L. A. (s.f.). Interpolación lineal frente a esquemas de tipo no lineal. En Detección de bordes con precisión subpíxel en imágenes digitales. Valencia: Universidad Politécnica de Valencia.
26. Hernanz, I. B. (2008). Algoritmo de detección de imágenes ruidosas y duplicadas. Madrid.
27. Jahne, B. (2002). Digital Image processing. 5ta. ed. Springer-Verlag, pp 481-494.
28. Joskowicz, J. (2008). Reglas y prácticas en eXtreme Programming.
29. Juristo, N., Moreno, A. M., & Vegas, S. (2006). Técnicas de evaluación de software.
30. K.Pratt, W. (2001). Digital Image Processing: PIKS Inside, Third Edition. New York: John Wiley & Sons, Inc. ISBN 0-471-22132-5.
31. Kim, D. (2013). Sobel Operator and Canny Edge Detector.

32. Larman, C. (2012). UML y Patrones 2da Edición.
33. López, C. P. (2002). Matlab y sus aplicaciones en las Ciencias y la Ingeniería. Madrid.
34. López, I. H. (2010). Sistema de análisis de la opacidad de la cápsula posterior.
35. Marmolejo, L. F. (2012). Implementación y evaluación de algoritmos de identificación por reconocimiento del iris ocular,. Santiago de Cali.
36. Mendoza, S. (2004). Metodologías del desarrollo de software.
37. Moeslund, T. B. (2009). Canny Edge Detection.
38. Mühlrad, D. (2008). Patrones de diseño.
39. Nuñez, P. D. (2013). Realce y Restauración de imagen.
40. Oliva, K. R. (2014). Suavizado mediante técnicas de filtrado morfológico en imágenes de microscopía.
41. Oracle. Java. (14 de 1 de 2015). Obtenido de <http://www.java.com/es/about/>. [En línea]: <http://www.java.com/es/about/>.
42. Oyarzo, W. O. (2006). Visión Artificial: Análisis Teórico del Tratamiento Digital de Imágenes Para su aplicación en la identificación de objetos.
43. Pinilla, C., Alcalá, A., & Arisa, F. (1997). Filtrado de imágenes en el dominio de la frecuencia. Departamento de Ingeniería Cartográfica,Revista de Teledetección.
44. Pratt, W. K. (2001). Digital Image Processing: PIKS Inside, Third Edition. California: Wiley-Interscience Publication.
45. Pressman, R. S. (2005). Ingeniería del Software. Un enfoque práctico 6ta edición. Nueva York.
46. Roushdy, M. (2006). Comparative study of edge detection algorithms applying on the grayscale noisy image using morphological filter. Ain Shams University,GVIP Journal, Volume 6.
47. Saini, S., B. K., & S. B. (2010). COMPARATIVE STUDY OF IMAGE EDGE DETECTION. India: School of Computing Science and Engineering.
48. Saleh Alamri, M. S., Kalyankar, D. N., & Khamitkar, S. (2010). Image segmentation by using edge detection. Research Student Computer Science

- Dept., Yeshwant College, Nande,431606 (M.S., India), Vol. 02, No. 03, 2010, 804-807.
49. Sánchez Mendoza, M. A. (2004). Metodologías del desarrollo de software. .
 50. Santillan, E. G. (2008). Detección y clasificación de objetos dentro de un salón de clases empleando técnicas de procesamiento digital de imágenes”.
 51. Shrivakshan, & Chandrasekar. (2012). A Comparison of various Edge Detection Techniques used in Image Processing. India.
 52. Spontón, H., & Cardelino, J. (2015). A Review of Classic Edge Detectors. Instituto de Ingeniería Eléctrica, Universidad de la República, Uruguay.
 53. Srinivasan, V., & R. T. (2003). Instrumentos y Equipos Oftalmológicos,Manual de Cuidado y Mantenimiento. India.
 54. Suárez, M. (2015). Competencias en TIC: Colección de Fasículos Digitales. Recuperado el 5 de 4 de 2015, de Cuaderno 1: Introducción a la programación y sus lenguajes: http://competenciastic.educ.ar/pdf/lenguajes_de_programacion_1.pdf
 55. Suaza, K. V. (2015). Mejora de historias de usuario y casos de prueba de metodologías ágiles con base en TDD. Cuaderno Activa, 41-53.
 56. Urrea, J. P., & Ospina, E. (2004). IMPLEMENTACIÓN DE LA TRANSFORMADA DE HOUGH PARA LA DETECCIÓN DE LÍNEAS PARA UN SISTEMA DE VISIÓN DE BAJO NIVEL.
 57. Valdés, L. C. (2010). Segmentación de imágenes para el análisis de eritrocitos en microscopía celular. Santa Clara : Universidad Central “Marta Abreu” de Las Villas ,Facultad de Ingeniería Eléctrica .
 58. Valencia Delgado, V., Moreira, J., & Chavez, P. (2005). Implementación de un algoritmo para la detección y conteo de células en imágenes microscópicas.
 59. W. O. Achicanoy, & L. F. Giraldo. (2005). Mejora de la Calidad Visual de la Imagen.
 60. Woods, R., & González, R. (2002). Digital Image Processing Second Edition.
 61. Z, T. V., F., W. S., & Arteaga, F. (2008). Sistema de visión por computadora para la detección de objetos esféricos a través de la transformada de Hough.

62. Zhu, X., & Milanfar, P. (2010). Automatic Parameter Selection for Denoising. IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 19, NO. 12, DECEMBER 2010.

Anexos

Anexo#1 Historias de Usuario

Tabla 23. Historia de usuario #2

Historia de usuario	
Número: 2	Nombre: Ajustar el contraste usando ecualización de histograma adaptativo.
Iteración asignada: 1	
Prioridad en negocio: Alta (Alta / Media / Baja)	Tiempo estimado: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Programador responsable: Jeidy Rosell Aldana
Descripción: Se requiere realizar una adaptación de la tarea anterior, pues dicha técnica trabaja igualmente con el histograma, con la diferencia de que este trabajara en áreas específicas de la imagen en lugar de toda la imagen, para evitar que en la regiones de alta intensidad de agudicen los valores.	
Observaciones:	

Tabla 24. Historia de usuario #3

Historia de usuario	
Número: 3	Nombre: Aplicar técnica de filtrado espacial no lineal Mediana.
Iteración asignada: 2	
Prioridad en negocio: Alta (Alta / Media / Baja)	Tiempo estimado: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Programador responsable: Jeidy Rosell Aldana
Descripción: Consiste en aplicar a la imagen esta técnica de filtrado digital, que se encarga de eliminar el ruido que pueda contener la imagen, trabajando con una matriz de convolución, en la cual el píxel central del núcleo se le va a asignar el valor de la mediana de los píxeles del entorno.	
Observaciones:	

Tabla 25. Historia de usuario #4

Historia de usuario	
Número: 4	Nombre: Definir el área de interés.
Iteración asignada: 3	
Prioridad en negocio: Alta (Alta / Media / Baja)	Tiempo estimado: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Programador Responsable: Jeidy Rosell Aldana
Descripción: En esta tarea se identifica el área de interés de la imagen, que es donde se encuentra la complicación postoperatoria, opacidad de la cápsula posterior, en la pupila.	
Observaciones:	

Tabla 26. Historia de usuario #5

Historia de usuario	
Número: 5	Nombre: Segmentar usando la Transformada de Hough Circular.
Iteración asignada: 3	
Prioridad en negocio: Alta (Alta / Media / Baja)	Tiempo estimado: 3 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Programador Responsable: Jeidy Rosell Aldana
Descripción: Esta funcionalidad trabaja con un sistema de conteo mediante, el píxel contiene una familia de la circunferencia en las cuales, la que más puntos de contorno tenga será la región seleccionada.	
Observaciones: Definir el píxel por el cual podría pasar la familia de circunferencia más grande.	

Tabla 27. Historia de usuario #6

Historia de usuario	
Número: 6	Nombre: Mejorar los bordes de la imagen aplicando el Laplaciano de paso alto.

Iteración asignada: 3	
Prioridad en negocio: Alta (Alta / Media / Baja)	Tiempo estimado: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Programador Responsable: Jeidy Rosell Aldana
Descripción: Este filtro es aplicado luego de la selección y definición del área de interés para el realce de los bordes de la imagen.	
Observaciones:	

Tabla 28. Historia de usuario #7

Historia de usuario	
Número: 7	Nombre: Integrar al PANDOC.
Iteración asignada: 3	
Prioridad en negocio: Alta (Alta / Media / Baja)	Tiempo estimado: 3 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Programador responsable: Jeidy Rosell Aldana
Descripción: La última de las tareas es la integración al software PANDOC.	
Observaciones:	

Tabla 29. Historia de usuario #8

Historia de usuario	
Número: 8	Nombre: Guardar
Iteración asignada: 3	
Prioridad en negocio: Alta (Alta / Media / Baja)	Tiempo estimado: 1 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Programador responsable: Jeidy Rosell Aldana

Descripción: Almacena la imagen en los archivos.

Observaciones:

Anexo #2 Tarjetas CRC

Tabla 30. Tarjeta CRC de la Clase Clahe

clahe	
Descripción: Se encarga del trabajo con el histograma de imagen, brindándole uniformidad a la matriz de intensidad.	
Atributos:	
Nombre	Descripción
int radioBloque	cantidad de píxles de region actual.
int bins	cantidad de bits con la que trabaja el histograma.
float declive	% intensidad luego de calcular la ponderación del vecindario.
Responsabilidades:	
Nombre	Collaborator
llevarEscalaGris(es(ImagenAPI imagen)	imagenAPI
aplicar(ImagenAPI imagen)	
redondearPositivo(float a)	

Tabla 31. Tarjeta CRC de la Clase Mediana

mediana	
Descripción: Su función es realizar el suavizado de la imagen y la eliminación del ruido.	
Atributos:	
Nombre	Descripción
int radius	es el tamaño de la vecindad.
Responsabilidades:	
Nombre	Collaborator
getRadius()	imagenAPI
calcLines(int radius)	
aplicar(imagenAPI fastBitmap)	
setRadius(int radius)	

Tabla 32. Tarjeta CRC de la Clase Convolución

convolución	
Descripción: Se encarga de realizar las operaciones del filtro laplaciano.	
Atributes:	
Nombre	Descripción
int ancho	almacena el ancho de la imagen a procesar
int alto	almacena el alto de la imagen a procesar
double[][] kernel	matriz de convolución por la cual se multiplican los píxeles de la vecindad.
int division	almacena el valor por el cual se divide la información de los píxeles, que se están analizando dentro de la vecindad.
Responsabilidades:	
Nombre	Collaborator
aplicar(ImagenAPI imagen)	imagenAPI
calculaLineas(double[][] kernel)	

Tabla 33. Tarjeta CRC de la Clase CirculoHough

CirculoHough	
Descripción: Se encarga dibujar la circunferencia resultante.	
Atributes:	
Nombre	Descripción
int maxx	valor máximo del desplazamiento en x
int maxy	valor máximo del desplazamiento en y
double radio	valor del radio por el cual se conforma la circunferencia
Responsabilidades:	
Nombre	Collaborator
draw(Graphics g)	Graphics
getRadio()	

Tabla 34. Tarjeta CRC de la Clase Laplacian

laplaciano	
Descripción: Su función es enfatizar los bordes de las estructuras de opacidad capsular.	
Atributos:	
Nombre	Descripción
double[][] kernel	matriz de convolución por la cual se multiplican los píxeles de la vecindad.
Responsabilidades:	
Nombre	Collaborator
calculaKemel()	imagenAPI
aplicar(ImagenAPI imagen)	

Anexo 3.Pruebas de Aceptación

Tabla 35. Caso de prueba de aceptación HU 1

Caso de prueba de aceptación
Código: HU_P1 Historia de Usuario:1
Nombre: Ajustar el contraste usando ecualización de histograma.
Descripción: Comprueba la funcionalidad que mejora el contraste de la imagen.
Condiciones de Ejecución: Se debe cargar la imagen oblicua proveniente de la lámpara de hendidura.
Entrada/Pasos de Ejecución: <ul style="list-style-type: none"> • El usuario selecciona la imagen. • El usuario selecciona la opción de detección y realce. • Se muestra la imagen preprocesada con el histograma de luminosidad uniforme.
Resultado Esperado: El resultado de esta funcionalidad es que la imagen se perciba el cambio de contraste, que se puedan observar mejor las zonas oscuras que no se ven por la dispersión de la luz.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 36. Caso de prueba de aceptación HU 3

Caso de prueba de aceptación
Código: HU_P3 Historia de Usuario:3
Nombre: Aplicar técnica de filtrado espacial no lineal Mediana.
Descripción: Comprueba la funcionalidad que mejora el contraste de la imagen.
Condiciones de Ejecución: La imagen debe ser cargada por el software PANDOC.
Entrada/Pasos de Ejecución: <ul style="list-style-type: none"> • El usuario selecciona la imagen. • El usuario selecciona la opción de detección y realce. • Se muestra la imagen preprocesada con la técnica de ecualización del histograma adaptativo aplicada y el filtro de la mediana.
Resultado Esperado: Se espera que el resultado sea la imagen con el histograma modificado especialmente en las regiones de baja intensidad y con el filtro de la media aplicado para un mejor suavizado.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 37. Caso de prueba para la HU 4.

Caso de prueba de aceptación
Código: HU_P1 Historia de Usuario:4
Nombre: Segmentar usando la Transformada de Hough Circular.
Descripción: Define el área de la pupila para realce de los bordes de la opacidad de a cápsula posterior.
Condiciones de Ejecución: Se debe cargar la imagen oblicua proveniente de la lámpara de hendidura.

<p>Entrada/Pasos de Ejecución:</p> <ul style="list-style-type: none"> • El usuario selecciona la imagen. • El usuario selecciona la opción de detección y realce. • Se muestra la imagen preprocesada con el área de la pupila definida.
<p>Resultado Esperado:</p> <p>El resultado es una imagen RGB que contenga el área de la pupila marcada para una posterior definición de esta región.</p>
<p>Evaluación de la Prueba: Prueba satisfactoria.</p>

Tabla 38. Caso de prueba para la HU 5.

Caso de prueba de aceptación
Código: HU_P2 Historia de Usuario:5
Nombre: Definir la región de interés.
Descripción: Recorta la imagen en correspondencia con el resultado de la transformada circular de Hough.
<p>Condiciones de Ejecución:</p> <p>Se debe cargar la imagen oblicua proveniente de la lámpara de hendidura.</p>
<p>Entrada/Pasos de Ejecución:</p> <ul style="list-style-type: none"> • El usuario selecciona la imagen. • El usuario selecciona la opción de detección y realce. • Se observa la imagen preprocesada con el área de la pupila enmarcada en el área del JFrame.
<p>Resultado Esperado:</p> <p>El resultado es el área de la pupila enmarcada para un mejor análisis de la opacidad capsular.</p>
<p>Evaluación de la Prueba: Prueba satisfactoria.</p>

Tabla 39. Caso de prueba de aceptación HU 6

Caso de prueba de aceptación
Código: HU_P6 Historia de Usuario:6
Nombre: Mejorar los bordes de la imagen aplicando el Laplaciano de paso alto.
Descripción: Comprueba la funcionalidad de realce de los bordes de la pupila.
Condiciones de Ejecución: Debe haberse cargado la imagen oblicua con la funcionalidad de cargar del software PANDOC.
Entrada/Pasos de Ejecución: <ul style="list-style-type: none"> • El usuario selecciona la imagen. • El usuario selecciona la opción de detección y realce. • Se muestra la imagen preprocesada con la técnica de ecualización del histograma adaptativo aplicada y el filtro de la mediana. • Se debe observar que la región de interés, la pupila, debe estar definida.
Resultado Esperado: Se espera que el resultado una imagen los bordes de la opacidad de la cápsula posterior, resaltados para que se pueda obtener una mayor precisión.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 40. Caso de prueba de aceptación HU 9

Caso de prueba de aceptación
Código: HU_P9 Historia de Usuario:9
Nombre: Integración al PANDOC.
Descripción: Comprueba la funcionalidad de realce de los bordes de la pupila.
Condiciones de Ejecución: Cargar la imagen con el PANDOC.

Entrada/Pasos de Ejecución:

- El usuario selecciona la imagen.
- El usuario selecciona la opción de detección y realce.
- Se debe haber ejecutado todo el preprocesamiento y la segmentación.

Resultado Esperado:

Se espera el resultado del preprocesamiento y la segmentación en una ventana emergente.

Evaluación de la Prueba: Prueba satisfactoria.

Glosario de Términos

Trilateración (medida de distancia)

Este método consiste en que en vez de medir ángulos se miden distancias entre todos los lados con distanciómetro. Las distancias que se obtienen en campo hay que reducirlas al horizonte, por ello deberán medirse también los correspondientes ángulos de inclinación, es decir se deben tomar las lecturas cenitales.