Universidad de las Ciencias Informáticas

Facultad 2

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas.

Segmentación de opacidad capsular en imágenes en retroiluminación mediante detección de texturas.

Autor:

Jessie Romero Pérez

Tutor:

Ing. Michel Álvarez Cancio

La Habana, junio de 2018

Declaración de autoría

Declaro ser el único autor del trabajo de diploma*”* Segmentación de opacidad capsular en imágenes en retroiluminación mediante detección de texturas*”*, concedo a la Universidad de las Ciencias Informáticas la autorización a hacer uso del mismo en su beneficio.

Para que conste firmo el presente documento a los ___ días del mes de junio del año 2018.

Jessie Romero Pérez

_____

Firma del autor

Ing. Michel Alvarez Cancio

_____

Firma del tutor

Dedicatoria

Le dedico este trabajo a toda mi familia que siempre me ha apoyado en especial a mi mamá y a mi papá por su amor y apoyo incondicional, a mis abuelos aunque tristemente dos de ellos no me pudieron ver culminar mis estudios, a mi hermana por ser la hermana que más yo quiero y a mi novio Víctor por sacarme una sonrisa cuando más la necesitaba.

Agradecimiento:

A mi mama y mi papá, por ser el motor impulsor de cada proyecto trazado en mi vida, por nunca decirme que no, por ese apoyo incondicional que me han mostrado siempre. Gracias por su sacrificio e inagotable cariño, por su confianza, por ser mí meta, por guiarme siempre por un camino de bien y apoyarme siempre que lo he necesitado. A ustedes, una y mil veces. Muchas Gracias.

A mis tios y tias que siempre me han apoyado, a mi hermana y a mi prima Juanita por apoyar en mis momentos de flaquesa, a mi novio el cual me ha soportado en estos meses tan convulsos, a la familia de mi novio la cual me recibió como un miembro mas de su familia.

En estos 5 años conocí a personas muy especiales a mi mejor amiga de la UCI Glenda la cual ha sido mi compañera de cuarto durante estos 5 años y a su mamá, personas maravillosas que me brindaron mucho cariño. A Greizy la cual entre lineas de detergente a demostrado ser una buena amiga. A Sheila la super jau por tantas horas de chachara. A mi mejor amigo el cual se ha convertido en un hermano para mí, Miguel gracias por aguantar tanta jodedera. A la Sochy que tantos momentos felices me ha proporcionado. A los amigos del aula con los que compartí momentos inolvidables. A mi tutor por darme la oportunidad de defender un tema que me gustó tanto.

En general agradezco a todas las personas que de una forma u otra han contribuido a la realización de este trabajo de diploma.

Jessie Romero Pérez

Resumen

La operación de la catarata es una de las más comunes a nivel mundial, su complicación postoperatoria más frecuente es la opacidad en la cápsula posterior. Para el diagnóstico de dicha enfermedad el equipo oftalmológico más usado es la lámpara de hendidura de la cual se obtienen distintos tipos de imágenes entre las que se encuentra las imágenes en retroiluminación. En la comunidad médica internacional no existe consenso alguno acerca de la correcta cuantificación de la opacidad de la cápsula posterior, por lo que los sistemas de detección automática son de gran importancia. Para ayudar a resolver este problema se desarrolló un algoritmo de segmentación basado en texturas para la identificación de la opacidad de la cápsula posterior en pacientes operados de cataratas utilizando la matriz de coocurrencia. Este algoritmo realiza un preprocesamiento aplicando filtro lineal y mejora del contraste, se inicializan las piscinas del procesamiento multihilo que pose Matlab para posteriormente obtenerse las variables de textura provenientes de la matriz de coocurrencia, las cuales son agrupadas con el método k-means y posteriormente segmentadas. Se obtuvo como resultado una herramienta capaz de segmentar las regiones de opacidad en imágenes provenientes de lámpara de hendidura, además se comprobó experimentalmente su eficacia en la detección de la opacidad en la cápsula posterior. Se realizó una comparación estadística que permitió comprobar que el algoritmo que emplea matriz de coocurrencia ofrece mejores resultados que el de marcos aleatorios de Markov, donde se emplearon las métricas Tasa de correctos equilibrados, Modificación de la Distancia de Hausdorff Normalizada y Sokal – Sneath.

Palabras claves: Matriz de coocurrencia, opacidad en la cápsula posterior, procesamiento digital de imágenes.

Índice

Índice de figuras

Índice de tablas

Índice de ecuaciones

Introducción

Las condiciones de ceguera y baja visión, constituyen la sexta causa de discapacidad a nivel mundial (1). Según la organización mundial de la salud (OMS), existen aproximadamente 285 millones de personas con discapacidad visual en el mundo, de las cuales 39 millones son ciegas y 246 millones presentan baja visión. La principal causa de ceguera en el mundo es la catarata, la cual es responsable del 51% de los casos de ceguera y del 33% de los casos de baja visión. (2)(3)

Dentro de los ojos existe un lente natural, este refracta los rayos de luz que ingresan en el ojo para ayudar en la visión, por lo que se considera que debe ser transparente. Con la aparición de la catarata este lente se nubla, causando la pérdida total o parcial de la visión. Con mayor frecuencia esta enfermedad tiende a aparecer en pacientes mayores de cincuenta años de edad. (4)

La operación de catarata es una de las más comunes donde la mayoría de los pacientes recuperan la visión parcial o totalmente con baja tasa de complicación. Durante las últimas décadas se han desarrollado distintas técnicas quirúrgicas para el tratamiento de las cataratas, que se han ido perfeccionando con el objetivo de reducir las complicaciones intra y postoperatorias. Sin embargo, en muchos casos la cirugía no es un éxito a largo o mediano plazo. (5)

La complicación más frecuente a largo o mediano plazo es la opacificación de la cápsula posterior (OCP). (5) Esta es una estructura que anatómicamente queda por detrás del lente intraocular implantado en el paciente operado de catarata. Según un artículo publicado en la Revista Cubana de Oftalmología su incidencia en la actualidad se encuentra entre 0,7- 47,6 % en los primeros cinco años de la cirugía. (5)(6)(7)

La OCP es la causa más frecuente de disminución de la agudeza visual en el postoperatorio. Para tratar esta enfermedad, se realiza una abertura en la cápsula posterior con un láser granate de neodimio: itrio aluminio, o por capsulotomía quirúrgica. Como toda operación médica este tratamiento no garantiza la eliminación total del padecimiento, por lo que se evidencia la necesidad de métodos para la detección o prevención de la misma. Debido a esto las investigaciones recientemente han dirigido su atención a estrategias de prevención de la OCP. Estas incluyen el diseño de los lentes intraoculares, técnicas quirúrgicas modificadas y/o mejoradas, implantes adicionales (*e.g.,* anillos de tensión capsular), empleo de fármacos y agentes citotóxicos, así como el uso de nuevas tecnologías. (6)

Disímiles investigaciones se han realizado con el objetivo de identificar los principales factores que influyen en la aparición de esta complicación postoperatoria. En la comunidad médica internacional no

existe consenso alguno acerca de la correcta cuantificación de la OCP (8). Varios médicos realizan el diagnóstico de la OCP mediante la lámpara de hendidura. No obstante, la evaluación biomicroscópica de la extensión y severidad de la opacidad es subjetiva y está sujeta a una amplia variación entre un observador y otro. Para obtener resultados confiables y comparables entre los múltiples estudios que sobre la prevención de la opacidad capsular se realizan, se hace sumamente necesaria la evaluación objetiva y cuantificación estandarizada del grado de OCP. (9) En los últimos años en todo el mundo se han desarrollado disímiles sistemas como POCO del inglés *Posterior Capsule Opacification*, EPCO (Evaluación de la opacidad de la cápsula posterior) (9), AQUA (Cuantificación automática de catarata secundaria) (10), y el sistema AA del inglés *Aslam Analysis* (11), dichos sistemas están basados en las imágenes en retroiluminación obtenidas por las lámparas de hendiduras. (12)

Las imágenes provenientes del artefacto constituyen una buena base para el análisis de la opacificación de la cápsula posterior, dado que una estructura transparente como la córnea o el cristalino no puede ser analizado con una iluminación directa.

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el Grupo de Investigación AIRI (*Artificial Intelligence Research and Innovation*), el cual tiene como principal resultado y línea de investigación el software PANDOC (Programa Analizador de Opacidad Capsular), realizado en colaboración con el Instituto Cubano de Oftalmología (ICO) "Ramón Pando Ferrer". 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), minimizando el sesgo de observación entre un médico y otro (8).

El software PANDOC trabaja con distintos tipos de imágenes, provenientes de equipos oftalmológicos, algunas de ellas son las imágenes en retroiluminación provenientes de la lámpara de hendidura. En estas imágenes es necesario identificar automáticamente las regiones con opacidad y cuantificar las mismas, para lo que se requieren tareas de realce, mejora y segmentación de dichas estructuras. En muchas imágenes los objetos no se encuentran diferenciados claramente en relación con el fondo, más bien tienen como característica espacial cierta textura visual, lo que dificulta la clasificación independiente de cada píxel, puesto que la textura está definida por regiones de píxeles y no por píxeles individuales.

El grupo de investigación AIRI de la UCI creó un algoritmo para la segmentación por texturas utilizando el campo aleatorio de Markov. Dicho algoritmo obtuvo buenos resultados, sin embargo, luego de su utilización solo se pueden observar las regiones con opacidad en una imagen en escala de grises donde

no se pueden distinguir correctamente las regiones y características de la imagen original. Para la realización del algoritmo anteriormente mencionado no se tuvo en cuenta de manera práctica la aplicación de la segmentación por textura utilizando matriz de coocurrencia la cual ha sido utilizada en el trabajo de imágenes médicas como es mapeo de imágenes digitales de fondo de ojo atendiendo a rasgos de textura. (13)

Por tal motivo la presente investigación trata acerca de la creación de un algoritmo que identifique las estructuras de opacidad de la cápsula posterior que pueden aparecer en las imágenes en retroiluminación.

A partir de la situación argumentada anteriormente se identifica el siguiente **problema a resolver**: ¿Cómo identificar opacidad de la cápsula posterior mediante la segmentación por textura en imágenes en retroiluminación de pacientes operados de catarata?

El problema se centra en el **objeto de estudio:** *Procesamiento de imágenes digitales*, enmarcado en el **campo de acción:** *Segmentación de textura en imágenes médicas*.

Partiendo de lo anterior se determina como **objetivo general:** Desarrollar un algoritmo para la segmentación por textura, utilizando matriz de coocurrencia, en imágenes en retroiluminación para la identificación de la opacidad de la cápsula posterior de pacientes operados de cataratas.

Teniendo en cuenta el problema a resolver se formuló la siguiente **Hipótesis**:

Hipótesis: Con el desarrollo de un algoritmo de segmentación por textura basado en matriz de coocurrencia para la identificación automática de opacidad capsular en imágenes en retroiluminación, se obtendrán mejores resultados en la identificación de la opacidad en la capsula posterior que con algoritmo de segmentación por textura basado en campo aleatorio de Markov.

El objetivo general de la investigación se desglosa en los siguientes objetivos específicos:

- Elaborar el marco teórico referente a los conceptos asociados al procesamiento digital de imágenes y la segmentación mediante texturas.
- Desarrollar un algoritmo para la segmentación en imágenes utilizando matriz de coocurrencia.
- Validar la solución propuesta mediante pruebas unitarias, de aceptación y métricas de segmentación.

Los **Métodos de investigación** utilizados para darle solución al objetivo propuesto fueron:

Métodos teóricos:

- Analítico – Sintético para descomponer el problema de investigación en tres elementos fundamentales: técnicas de procesamiento digital de imágenes, algoritmo para la detección de variables de texturas y algoritmos de agrupamientos; profundizar en su estudio y luego sintetizarlos en la solución propuesta.
- Histórico – Lógico se utilizó para analizar el surgimiento y evolución de las técnicas para la segmentación de imágenes y así facilitar la comprensión del objeto y campo de estudio.
- Modelación: para la creación del diagrama de actividades correspondiente a la propuesta de solución.

Métodos empíricos:

- Hipotético-Deductivo para la creación y verificación de la hipótesis que explique la superioridad del algoritmo que utiliza matriz de coocurrencia sobre el que utiliza campo aleatorio de Markov, en la segmentación por textura en imágenes en retroiluminación.
- Análisis documental: en la revisión de la literatura especializada para extraer la información relacionada con la segmentación de imágenes por texturas, que permitió realizar el proceso de investigación.
- Observación: para adquirir conocimientos relacionados con las texturas presentes en las imágenes con opacidad en la cápsula posterior y las técnicas para la segmentación de la mismas.

La estructura del documento se caracteriza por la presencia de: introducción, 3 capítulos, conclusiones, bibliografías y los anexos. El contenido de cada capítulo se describe a continuación:

Capítulo 1: Fundamentación teórica de la investigación, dedicado a los fundamentos teóricos de la investigación. Se analiza una variedad de técnicas para la preprocesamiento de imágenes, así como para la segmentación por texturas. Se introduce y aborda acerca de la utilización de la matriz de coocurrencia para la obtención de variables de texturas necesarias para la segmentación.

Capítulo 2: Herramienta para la segmentación de imágenes en retroiluminación con opacidad en la cápsula posterior, describe la herramienta para la segmentación de imágenes en retroiluminación, junto a las etapas de la metodología empleada en la investigación. Se describen las funcionalidades de la herramienta y las técnicas y configuraciones utilizadas para en su desarrollo.

Capítulo 3: Validación de resultados, contiene la evaluación de los resultados alcanzados en la investigación, a través de pruebas unitarias y de aceptación. Además, para validar la calidad de los resultados obtenidos se realizó validación estadística.

Capítulo 1. Fundamentación teórica de la investigación

En el siguiente capítulo se realizará una explicación desde el punto de vista teórico del problema general en que se enmarca la investigación. Se abordan los principales conceptos asociados al dominio del problema, y se realiza un estudio del arte del procesamiento de imágenes, haciendo énfasis en las principales técnicas y métodos de segmentación utilizados en imágenes médicas, para luego identificar un algoritmo que detecte regiones con opacidad capsular en pacientes sometidos a cirugías de cataratas.

1.1 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 catarata, 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 (7). En el campo de la oftalmología esto es conocido como OCP y constituye la complicación tardía más importante en la cirugía de catarata en los tiempos modernos. (13)

En Cuba el equipo oftalmológico más utilizado para la detención de OCP es la lámpara de hendidura, conocida también como biomicroscopio; es un microscopio binocular que emite una fuerte luz, gracias a la cual se puede ver en tres dimensiones y un aumento de entre cuatro y cuarenta veces el tamaño del ojo. (14) De este equipo se obtienen las imágenes en retroiluminación que son imagines o gráficos digitales.

Una imagen digital consiste en una representación bidimensional de una imagen a partir de una matriz numérica, frecuentemente en binario (unos y ceros). (15) Dependiendo de si la resolución de la imagen es estática o dinámica, puede tratarse de una imagen matricial (o mapa de bits) o de un gráfico vectorial (16). Las imagines digítales presentan características como: contraste, brillo, intensidad y elementos de textura. La textura es una propiedad innata a la naturaleza de toda superficie, esta es usada para caracterizar la superficie de un objeto o fenómeno dado y es indudablemente uno de los rasgos principales usados en el procesamiento de imágenes y en el reconocimiento de patrones. (17)

1.2 Procesamiento digital de imágenes.

En los últimos años han surgido varias herramientas de software capaces de expresar el grado de incidencia de la OCP. La ventaja principal de estos sistemas radica en que se reduce la variabilidad del observador y aumenta la exactitud, aunque en ocasiones no son los más favorables ya que trabajan con

imágenes obtenidas de las lámparas de hendidura, dichas imágenes poseen una desventaja muy peculiar al presentar una mancha pronunciada de luz o reflejo especular, producto del resultado de la reflexión del artefacto utilizado. Luego, es necesario realizar un proceso de mejora y tratamiento de la imagen digital para disminuir la proyección de la luz y de esta manera adquirir una imagen mejorada para la obtención de resultados investigativos que den cumplimiento al objetivo propuesto. (17)

Una de las áreas más ligadas a la visión computacional es el procesamiento de imágenes, el cual tiene como objetivo realizar de forma sistemática un conjunto de operaciones con el fin de reformar y adaptar una imagen original para su posterior utilización o interpretación. El tratamiento digital de imágenes consiste en realizar un mapeo de una imagen a puntos definidos discretamente, a los cuales se les asigna un par de coordenadas y un valor de intensidad. La alteración de los valores de intensidad por medio de una computadora permite efectuar con gran facilidad operaciones de realce y de análisis de la imagen.

El procesamiento de imágenes se divide en cinco etapas fundamentales (18) (ver Figura 1) .

Figura 1. Etapas del procesamiento de imágenes. (18)

Durante el presente trabajo la investigación estará englobada hasta la etapa de segmentación de la imagen, el cual es objetivo fundamental de la investigación para cuantificar objetivamente la opacidad de la cápsula posterior.

1.2.1 Captura de imágenes digitales.

El objetivo de la captura o adquisición de una imagen, es llevar la misma dentro de la computadora, donde pueda ser almacenada o visualizada para luego ser manipulada y mejorada. El principal elemento

en la adquisición de una imagen es una cámara que captura las imágenes de un objeto. (16) Entre las modalidades de adquisición de imágenes médicas se encuentra el Pentacam, que es un equipo oftalmológico de alta tecnología capaz de reconstruir imágenes tridimensionales de alta resolución del polo anterior del ojo. Esto lo realiza a partir de múltiples fotografías tomadas mediante una cámara rotacional del sistema Scheimpflug con que cuenta el equipo.

El análisis de los tomogramas en 3D obtenidos a partir del sistema Scheimpflug puede potencialmente ser utilizado para la cuantificación objetiva de la OCP. (14) El principal inconveniente para la utilización del artefacto es que en el país solo se cuenta con pocos equipos, por lo que se hace necesario trabajar con otro equipo para la adquisición de las imágenes. Entre las modalidades de adquisición de imágenes médicas se encuentran las lámparas de hendiduras, estas se encuentran dentro de los instrumentos de diagnóstico más comúnmente usados por los oftalmólogos.

La lámpara de hendidura proporciona iluminación y magnificación para examinar las varias partes del ojo. La luz se proyecta como una franja o hendidura brillante, lo que permite el examen detallado del ojo en pequeños segmentos. Se utiliza en el examen del segmento anterior del ojo, incluyendo el lente. Con lentes suplementarios la lámpara de hendidura es útil en el examen de la región posterior del ojo, el fondo del ojo y buena parte de la retina. Una serie de accesorios se pueden añadir a una lámpara de hendidura para convertirla en un instrumento de medida, estos pueden 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, entre otras. (19) Una de las ventajas que tiene la utilización de este artefacto es que en el país por cada consulta de oftalmología se encuentra como mínimo una lámpara de hendidura, lo que hace que la mayoría de las imágenes utilizadas para la identificación de la OCP son prevenientes de la misma. (20)

Las imágenes utilizadas para la identificación de la OCP en la presente investigación son imágenes en retroiluminación. Estas imágenes presentan cierta desventaja como resultado de la reflexión de la luz proveniente del artefacto, por lo que se hace necesario realizar un preprocesamiento para lograr resultados favorables en la etapa segmentación.

1.2.2 Preprocesamiento de imágenes

Al digitalizar una imagen, es común la presencia de ruido o degradación, así como otros efectos indeseados como los reflejos luminosos mencionados, razón por la cual es importante restaurarla antes de ser procesada. En la etapa de 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. (17, 21)

Mejora del contraste.

A continuación, se muestran algunas de las técnicas utilizadas para mejorar el contraste en las imágenes.

La ecualización adaptativa del histograma (CLAHE) es una técnica muy utilizada para mejorar el contraste en las imágenes. Un histograma es una función que muestra una imagen con los distintos niveles de gris que contiene. La ecualización del histograma consiste en una expansión del histograma, dotando al mismo de mayor linealidad y haciendo que éste ocupe el ancho del espectro de tonalidades grises por completo, lo cual implica una mayor utilización de los recursos disponibles y un aumento del contraste. (22)

El algoritmo White Patch (WP), o Parche blanco en español toma en cuenta el valor más grande encada componente de color como una representación del blanco de la imagen. Computacionalmente, este parche blanco es calculado al encontrar la máxima intensidad en cada canal. El algoritmo WP puede hacerse más robusto si se calcula un histograma para cada componente de color y se considera el iluminante como un alto porcentaje de la acumulación del mismo. Este método es usado frecuentemente para remover sombras en la imagen. (22)

El algoritmo Gray *World* (GW), o Mundo gris en español es el método más conocido de constancia de color, usado como referencia por otros algoritmos, el GW está basado en la suposición de que, en promedio, el mundo es gris, y estima el iluminante usando el color promedio de todos los píxeles. Se asume que la información dada por el promedio de cada canal de la imagen representa el nivel gris. (22)

Filtros digitales

Los filtros son operaciones que se aplican a los píxeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella. El proceso de filtrado puede llevarse a cabo sobre los dominios de frecuencia y/o espacio. El dominio de frecuencia se basa en modificaciones de la transformada de Fourier. Este método trata de calcular la transformada de Fourier de la imagen a intensificar, multiplicar el resultado por la función de transferencia de un filtro y, finalmente, tomar la transformada de Fourier inversa para llegar a una imagen mejorada. El dominio espacial trabaja con técnicas de manipulación de píxeles de la imagen. (23)(9)

En este proceso se hace corresponder, para cada uno de los puntos de la imagen, un conjunto de píxeles próximos al píxel objeto con la finalidad de obtener una información útil, dependiendo del tipo de filtro aplicado, que permita actuar sobre el píxel concreto en que se está llevando a cabo el proceso de filtrado. De este modo se puede mejorar la imagen y obtener datos que podrían ser utilizados en futuras acciones o procesos de trabajo sobre ella. Existen dos tipos de filtros para el dominio del espacio: filtros lineales (filtros basados en kernels o máscaras de convolución) y filtros no lineales. (24)

Los filtros lineales se basan en el concepto que afirma, que la función de transferencia y el impulso o función de distribución puntual de un sistema lineal son transformadas de Fourier. (21) Se encuentran en él los denominados filtros de bajas frecuencias o paso bajo que son aquellos que en el dominio de las frecuencias dejan pasar solamente las frecuencias bajas y eliminan altas frecuencias, se caracterizan por suavizar los bordes y dar la impresión de imágenes borrosas. Su objetivo es suavizar la imagen, son de gran utilidad cuando la imagen presenta una cierta cantidad de ruido que se desea eliminar. (25) Algunos de los filtros lineales son:

- Filtros gaussianos: simulan una distribución gaussiana. El valor máximo aparece en el píxel central y disminuye hacia los extremos, siendo más rápido cuanto menor sea el parámetro de desviación típica. El resultado será un conjunto de valores entre 0 y 1. Para transformar la matriz a una matriz de números enteros se divide toda la matriz por el menor de los valores obtenidos. Este filtro presenta desventajas ya que puede producir la pérdida de detalles, el aumento de borrosidad, y la disminución de la nitidez.
- Filtro de la media: consiste en sustituir cada píxel por la media aritmética de los puntos que tiene alrededor (incluido el mismo). Se selecciona una ventana de NxN puntos donde el punto a sustituir es el central, se toma la media de la suma de los valores de los píxeles presentes en la ventana y se sustituye el píxel en la nueva imagen por el valor obtenido.
- Filtro de media ponderada: los elementos de la matriz de filtrado no son todos 1 sino que se da más peso a uno de ellos (generalmente el central) para obtener un resultado más parecido a la imagen original y evitar que aparezca borrosa. (25)
- Filtros adaptativos: son considerablemente más complejos ya que los coeficientes de ponderación se recalculan para cada uno de los píxeles en función del histograma. Se han utilizado con gran éxito filtros adaptativos para eliminar el ruido de las imágenes de radar y para detectar con un solo filtro, diferentes elementos. (24)

En contraste los filtros de paso alto tienen como objetivo resaltar las zonas de mayor variabilidad eliminando lo que sería la componente media. Por otra parte, la respuesta de cada píxel está contaminada por la de los píxeles vecinos ya que la radiación reflejada por un píxel se reparte hacia los píxeles vecinos. Los filtros de paso alto consiguen también eliminar en parte esta contaminación (9). Existen diversos métodos:

- Filtro laplaciano: responde a las transiciones de intensidad, rara vez se utiliza en la práctica para la detección de bordes. Como es una derivada de segundo grado es inaceptablemente sensible al ruido. Además, produce bordes dobles y es incapaz de detectar direcciones de borde. Un empleo más general del laplaciano consiste en encontrar la ubicación de bordes utilizando sus propiedades de paso por cero. (18)
- Sustracción de la media: si se considera que un filtro de paso bajo sirve para resaltar componentes a gran escala eliminando la variabilidad local. La matriz de filtrado de este filtro de sustracción de la media puede calcularse directamente restando a la matriz de filtrado identidad.

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. (26)

Algunos de los filtros no lineales son:

- Filtro de la moda. La moda de un conjunto de valores se define como el valor que más se repite dentro de ellos. Por lo tanto, el filtro de moda consiste en calcular el valor más repetido dentro de todos los píxeles de una ventana. (18)
- Filtro de la mediana. Se realiza mediante la adopción de la intensidad de todos los píxeles en una vecindad ordenados en un vector de acuerdo con sus valores. El píxel que tenga la magnitud de la mediana del ordenamiento se utiliza entonces para reemplazar el píxel central de la vecindad. (18, 20)

Las técnicas de tratamiento no lineal de imágenes digitales son a menudo mejor que los filtros lineales en remover el ruido sin distorsiones en las características de la imagen. Sin embargo, el diseño y el análisis de filtros no lineales es mucho más difícil que para filtros lineales.

Segmentación de imágenes:

La segmentación subdivide una imagen en sus partes constituyentes u objetos, con el fin de separar las partes de interés del resto de la imagen, por lo que el nivel al que se lleva a cabo esta subdivisión depende del problema a resolver. (27) Los algoritmos de segmentaciones más usados son la segmentación por detección de bordes, segmentación por regiones, umbralización, particionamiento gráfico, métodos basados en el histograma y segmentación por textura.

1.3 Segmentación por texturas

La segmentación de textura incluye la identificación de regiones con texturas uniformes, en una imagen dada. Para decidir si una región tiene textura uniforme se necesitan medidas apropiadas para describirla. Para *Sklanky* (1978) (17), *"… una región de la imagen posee una textura constante si un conjunto de estadísticos locales u otras propiedades locales de la imagen permanecen constantes, varían lentamente o aproximadamente periódicas…"*. Para los autores la definición de textura es apropiada en el contexto de la segmentación, sin embargo la textura posee connotaciones tanto locales como globales, y está caracterizada por la invariancia de ciertas medidas locales o propiedades más allá de una región de la imagen.

A continuación, se explican algunos de los métodos principales para lograr la segmentación por textura:

Energía de la transformada de Fourier

En una imagen donde se tengan diferentes regiones y cada una de ellas se caracterice por presentar una frecuencia espacial distinta, se puede asumir que son regiones con diferentes texturas. Si se aplica a esta imagen un detector de borde convencional (Robert, laplaciano, etc.) este no podrá discriminar el entorno de cada región de texturas diferentes, ni localizar zonas de igual textura. (28)

El análisis de Fourier ha sido ampliamente utilizado en el procesamiento de imágenes, ya que tiene varias propiedades útiles. El análisis de Fourier es robusto frente a las perturbaciones que a menudo aparecen en las imágenes, por ejemplo, los cambios de iluminación y los ruidos aditivos. El espectro de frecuencia de una imagen puede calcularse utilizando el algoritmo de transformada rápida de Fourier (FFT) que es práctico y eficiente desde el punto de vista de la computación. (29)

Desventaja de la transformada de Fourier:

Como el espectro de potencia de Fourier se calcula típicamente sobre toda la imagen, estas características recopilan información global en lugar de propiedades de textura localizadas.

Modelo marco aleatorio gaussiano – markovino

Los modelos de campo aleatorio de Markov (MRF), han sido bastante exitosos para el modelado de texturas y la segmentación. Capturan las características locales de una imagen asumiendo una distribución de probabilidad condicional local. Cuando esta distribución es gaussiana, el modelo se denomina campos aleatorios Gaussian-Markov (GMRF). (23)(17)

Basándose en la propiedad markoviana, que es simplemente la dependencia de cada píxel en la imagen solo en sus vecinos, y utilizando un modelo de campo aleatorio de Markov gaussiano (GMRF) para vecinos de Markov de tercer orden, los parámetros de GMRF se estiman usando método de estimación de errores mínimos cuadrados (30).

El modelo GMRF se define por la siguiente fórmula:

Ecuación 1. Modelo GMRF

$$p\left(I_{ij}\middle|I_{kl}\ (k,l) \in N_{ij}\ \right) = \frac{1}{\sqrt{2\sigma^2}} \exp\{\frac{I_{ij} - \sum_{l=l}^{n} \alpha_l S_{kl;l}}{2\sigma^2}\}$$

El lado derecho de la ecuación representa la probabilidad de que un píxel (i, j) tenga un valor de gris específico I_{ij}, dados los valores de sus vecinos, *n* es el número total de píxeles en el vecindario N_{kl} del píxel I_{ij}, lo que influye en su valor, α_l es el parámetro con el que un vecino influye en el valor de (i, j) y $s_{kl;l}$ es el valor del píxel en la posición correspondiente, donde:

$$S_{ij;1} = I_{i-1,j} + I_{i+1,j} \text{(1.1)}$$

$$S_{ij;2} = I_{i,j-1} + I_{i,j+1} \text{(1.2)}$$

$$S_{ij;3} = I_{i-2,j} + I_{i+2,j} \text{(1.3)}$$

$$S_{ij;4} = I_{i,j-2} + I_{i,j+2} \text{ (1.4)}$$

$$S_{ij;5} = I_{i-1,j-1} + I_{i+1,j+1} \text{ (1.5)}$$

$$S_{ij;6} = I_{i-1,j+1} + I_{i+1,j-1} \text{ (1.6)}$$

Los parámetros *α* y *σ* se estiman utilizando el método de estimación de errores mínimos cuadrados, de la siguiente manera:

Ecuación 2. Estimación de errores mínimos cuadrados

$$\begin{pmatrix}\alpha_1\\ \vdots\\ \alpha_n\end{pmatrix} = \left\{\sum_{ij}\begin{bmatrix} s_{ij};1 s_{ij};1 & \dots & s_{ij};1 s_{ij};n \\ \vdots & \ddots & \vdots \\ s_{ij};n s_{ij};1 & \dots & s_{ij};n s_{ij};n \end{bmatrix}\right\}^{-1} \sum_{ij} I_{ij}\begin{pmatrix} S_{ij;1}\\ \vdots \\ S_{ij;n}\end{pmatrix}$$

$$\sigma^2 = \frac{\sum_{ij}\left[I_{ij} - \sum_{l=1}^{n}\alpha_l s_{ij;l}\right]}{(M-2)(N-2)} \quad (2.1)$$

Donde M y N son las dimensiones de la imagen (17).

Filtro de Gabor

Es un filtro lineal cuya respuesta al impulso se define mediante una función armónica multiplicada por una función gaussiana. Los filtros de Gabor se pueden aplicar a muchas aplicaciones de procesamiento de imágenes, como segmentación de texturas, análisis de documentos, detección de bordes, identificación de retina y representación de imágenes. La ventaja de estos filtros es que satisfacen el mínimo ancho de banda espacial según el principio de incertidumbre. Por lo tanto, proporcionan una resolución óptima simultánea en los dominios espaciales y de frecuencia espacial. Los filtros Gabor se usan para resolver problemas que involucran imágenes complicadas compuestas por regiones texturizadas. (31)

Una función de Gabor se define como:

Ecuación 3. Función de Gabor

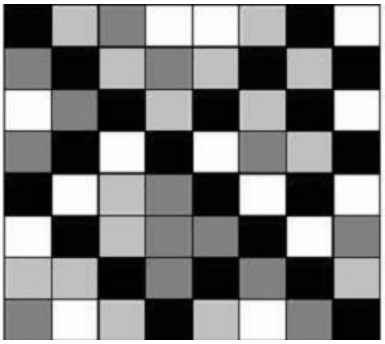
$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} exp\left[-\frac{1}{2}\left\{\frac{x^2}{\sigma^2 x} + \frac{y^2}{\sigma^2 y}\right\} + 2\pi j W x\right]$$

En la investigación no se utilizan ninguno de los métodos de segmentación anteriormente mencionados, por lo que la segmentación de las texturas relacionadas con la opacidad en la cápsula posterior se realizará de forma manual. Para la obtención de las variables de texturas necesarias para la realizar la segmentación se utilizó la matriz de coocurrencia.

1.4 Matriz de coocurrencia

La matriz de coocurrencia o GLCM (del inglés grey level coocurrence matrix) describe la frecuencia de un nivel de gris que aparece en una relación espacial específica con otro valor de gris, dentro de un área de una ventana predeterminada. La matriz de coocurrencia es un resumen de la forma en que los valores de los píxeles ocurren uno al lado de otro en una pequeña ventana. (32, 33)

La matriz de coocurrencia es una tabla que indica cómo están relacionadas las distintas combinaciones de los valores de brillo de los píxeles (niveles de grises) que ocurren en la imagen por lo que la matriz de coocurrencia tendrá dimensiones de NxN si existen N niveles de grises. Para una imagen de 256 niveles de grises se tendrá una matriz de coocurrencia de 256x256 para cada dirección y ventana. La figura 2 se considera una imagen de prueba donde los valores corresponden a los valores de grises. La imagen tiene cuatro niveles de grises (0, 1, 2, 3) por lo tanto se obtendrá una matriz de coocurrencia de 4x4. (34–36)



0	2	1	3	3	2	0	3
1	0	2	1	2	0	2	0
3	1	0	2	0	2	0	3
1	0	3	0	3	1	2	0
0	3	2	1	0	3	0	3
3	0	2	1	1	0	3	1
2	2	0	1	0	1	0	2
1	3	2	0	2	3	1	0

Figura 2 Niveles de grises de una imagen. (17)

Cálculo de la matriz

La matriz de coocurrencia considera la relación espacial entre dos píxeles, llamados píxeles de referencia y píxel vecino. Por ejemplo, si se escoge el píxel vecino que está situado a la derecha de cada píxel de referencia, este se expresa como (1,0), donde 1 será el píxel en la dirección x y 0 será un píxel en la dirección y. Así, cada píxel en la ventana se convierte en el píxel de referencia empezando por el ubicado arriba a la izquierda y finalizando abajo a la derecha. Se observa que los píxeles ubicados en el margen derecho de la imagen original no tienen vecino a la derecha por lo tanto no son usados en el cómputo. (32, 37)

La relación espacial entre el píxel de referencia y su vecino puede ser en cualquiera de las ocho direcciones (N, S, E, O y las cuatro diagonales), pero solo se toman cuatro, ya que la N es opuesta a la S y en vez de contarlas separadamente hay formas más sencillas de medirlas (matriz simétrica, que más

adelante se detalla). Cuando se habla de una relación “espacialmente invariante” se eligen las cuatro direcciones N, NE, E y SE y se promedian. Esto también se expresa respectivamente como 0°, 45°, 90° y 135° y se ve en la figura 3. (38)

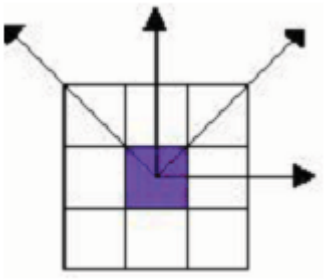


Figura 3 relación espacial de pixeles. (17)

Así, se pueden utilizar las diferentes relaciones entre píxeles:

- (1,0) ó 0º un píxel a la derecha.
- (1,1) ó 45º un píxel a la derecha y un píxel arriba.
- (0,1) ó 90º un píxel arriba.
- (-1,1) ó 135º un píxel a la izquierda y un píxel arriba.

Tomando en cuenta la dirección (1,0) mencionada anteriormente y que el píxel de referencia estará inmediatamente después de su píxel vecino, las posibles combinaciones entre los cuatro niveles de grises están mostrados en la Tabla 1. (17, 34)

Tabla 1 posibles combinaciones entre los cuatro niveles de grises. (17)

Pixel Vecino Pixel de Referencia	0	1	2	3
0	(0,0)	(0,1)	(0,2)	(0,3)
1	(1,0)	(1,1)	(1,2)	(1,3)
2	(2,0)	(2,1)	(2,2)	(2,3)
3	(3,0)	(3,1)	(3,2)	(3,3)

En la tabla anterior la primera celda debe ser llenada con la cantidad de veces que ocurre la combinación (0,0), es decir, cuántas veces en el área de la ventana un píxel con valor de gris igual a 0 (píxel vecino), está situado a la derecha de otro píxel con valor 0 (píxel de referencia). Por lo tanto, existen diferentes

matrices de coocurrencia para cada relación espacial según se considere el vecino de arriba, al lado o en diagonal. (17, 34)

De lo anterior resulta una matriz de la forma indicada en la figura 4:

0	2	8	8
8	1	2	2
8	4	1	1
3	4	3	1

Figura 4 Matriz resultante. (17)

Normalización de la matriz.

La matriz mostrada en la figura 2 tiene cada píxel vecino en su respectiva derecha. Si el cálculo se realiza solo de este modo, usando la dirección (1,0), entonces el número de veces que aparece la combinación (2,3) no es el mismo que la combinación (3,2), por lo tanto, la matriz no es simétrica. La simetría es necesaria para el cálculo de texturas. Para la conformación de una nueva matriz simétrica, se procede a sumar las veces que aparecen las combinaciones que tienen valores opuestos y se sustituye el valor de las dos por el nuevo resultado (*e.g.*, (2,3) y (3,2)). De la matriz anterior se obtiene como matriz simétrica la mostrada en la figura 5. (17, 34)

0	10	16	11
10	2	6	6
16	6	2	4
11	6	4	2

Figura 5 Matriz simétrica. (17)

Una vez obtenida la matriz simétrica, el paso siguiente es expresar esta matriz como probabilidad. Se asume que toda la información está contenida en la matriz de dependencia espacial desarrollada para las cuatro direcciones mostradas en la figura 3. En general, cuanto mayor es el número de la diagonal en la matriz de coocurrencia, más homogénea es la textura en esa parte de la imagen que está siendo analizada. Entonces, los elementos de la diagonal representan pares de píxeles que no tienen diferencias en su nivel de gris. Si estos elementos tienen altas probabilidades, entonces la imagen no muestra mucho contraste, la mayoría de los píxeles son idénticos a sus vecinos. Sumando los valores

de la diagonal se obtiene la probabilidad de que un píxel tenga el mismo nivel de gris que su vecino. Además, las líneas paralelas a la diagonal representan los pares de píxeles con una diferencia de un nivel de gris, es por eso que, a medida que se aleja de la diagonal la diferencia entre los niveles de grises es mayor. Sumando los valores de estas diagonales paralelas se obtiene la probabilidad de que un píxel tenga 1, 2, 3, etc. niveles de grises de diferencia con su vecino como se observa en la figura 6. (17, 34)

0	0.089	0.142	0.098
0.089	0.017	0.053	0.053
0.142	0.053	0.017	0.035
0.098	0.053	0.035	0.017

Figura 6 Matriz de probabilidades. (17)

A partir de la matriz de probabilidades se pueden extraer un grupo de variables de textura que caracterizan la textura que presenta cada pixel en la imagen, dichas texturas son utilizadas para realizar la segmentación de las regiones con OCP.

Medidas de Textura

Hasta este punto se ha detallado como se crea una matriz normalizada, expresada como probabilidad, para una determinada relación espacial entre dos píxeles vecinos. Una vez construida, de esta matriz pueden derivarse diferentes medidas, en esta sección se definen algunas de ellas, y se desarrollan con mayor profundidad las medidas cuyos cálculos pueden ser realizados manualmente por su sencillez. (39)

Las medidas de texturas más importantes abordadas en la bibliografía son:

- **Homogeneidad**

  Se calcula mediante la ecuación

Ecuación 4. Ecuación de la variable de textura de homogeneidad

$$\sum_{j,j=0}^{N-1} P_{i,j} \Big/ 1 + (i-j)^2$$

Siendo P i, j la probabilidad de coocurrencia de los valores de gris i y j, para una distancia dada. (39)

- **Contraste**

  Es lo opuesto a la homogeneidad, es decir es una medida de la variación local en una imagen. Tiene un valor alto cuando la región dentro de la escala de la ventana tiene un alto contraste.

Ecuación 5. Ecuación de la variable de textura contraste

$$\sum_{j,j=0}^{N-1} P_{i,j}\,(i-j)^2$$

La matriz de pesos, toma valores que crecen exponencialmente a medida que se aleja de la diagonal (0,1,4,9,(39).

- **Disimilaridad**

  Similar al Contraste, es alta cuando la región tiene un contraste alto.

Ecuación 6. Ecuación de la variable de textura disimilaridad

$$\sum_{j,j=0}^{N-1} P_{i,j}\,|i-j|$$

Para construir la matriz de pesos, estos pesos crecen linealmente a medida que nos alejamos de la diagonal (0,1,2,3).

- **GLCM Media**

  La ecuación para su cálculo es la siguiente:

Ecuación 7. Ecuación de matriz de niveles de grises

$$\sum_{j,j=0}^{N-1} iP_{i,j}$$

Se hace notar la diferencia que existe entre esta GLCM media de la media aritmética de los valores de grises de los píxeles de la ventana. La media en la matriz de coocurrencia no es simplemente el promedio de los valores originales de los niveles de gris en la ventana. El valor del pixel no es ponderado por su frecuencia por sí mismo, sino por la frecuencia de su coocurrencia en combinación de un determinado valor del pixel vecino.

- **Desviación standard**

  Corresponde con la desviación standard de los niveles de gris en la matriz, esta presenta valores altos cuando la desviación estándar en los niveles de grises dentro de la ventana es también alta.

Las ecuaciones para el cálculo de la varianza que a continuación de muestran, dan el mismo resultado tanto para i como para j, pues la matriz es simétrica.

Ecuación 8. Ecuaciónes para calcular la varianza

$$\sigma_i^2 = \sum_{j,j=0}^{N-1} P_{i,j} (i - \mu_i)^2$$

$$\sigma_j^2 = \sum_{j,j=0}^{N-1} P_{i,j} (i - \mu_j)^2$$

Mientras que las ecuaciones para el cálculo de la desviación Standard son las siguientes:

Ecuación 9 Ecuación para el cálculo de desviación standard

$$\sigma_i = \sqrt{\sigma_i^2}\ \sigma_j = \sqrt{\sigma_j^2}$$

Esta medida se basa en la media y la dispersión alrededor de la media de los valores de las celdas de la matriz de coocurrencia. Como la varianza utiliza específicamente la combinación entre los píxeles de referencia y vecino, no es la misma que la varianza de los niveles de grises de la imagen original.

- **Entropía**

  Es alta cuando los elementos de la matriz de coocurrencia tienen relativamente valores iguales. Es baja cuando los elementos son cercanos a 0 o 1 (por ejemplo, cuando la imagen es uniforme dentro de la ventana)

Ecuación 10. Ecuación de cálculo de entropía.

$$\sum_{j,j}^{N-1} -P_{i,j}\ ln(P_{i,j})$$

Se asume que $0 * ln(0) = 0$

Si, $P\ i,j$ es una probabilidad y toma valores entre 0 y 1, entonces el $ln\ (P\ i,j\ )$ siempre tomará valores de 0 o negativos. Cuanto más pequeño sea el valor de $P\ i,j$, es decir que la ocurrencia de esa combinación de píxeles es poco común, el valor absoluto de $ln\ (P\ i,j\ )$ será mayor

- **Correlación**

Ecuación 11 Ecuación de la variable de textura correlación

$$\sum_{j,j=0}^{N-1} P_{i,j} \left[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right]$$

Esta medida se obtiene a partir de la fórmula anterior. Se calcula de una forma diferente a las anteriores medidas, por lo cual la información que suministra es esencialmente distinta, es independiente de las otras medidas. Por lo tanto, es de esperar que pueda ser usada en combinación con otra medida textural.

1.5 Algoritmos de agrupamiento

El problema de formar grupos en un conjunto de datos es muy importante para el conocimiento del comportamiento de una población de la cual solo se tiene una pequeña cantidad de información. La solución de estos problemas se realiza mediante la creación de algoritmos de agrupamiento. Entre los métodos de agrupamiento paramétricos se encuentran las mixturas finitas, éstas son una poderosa herramienta para modelar densidades de probabilidad de conjuntos de datos univariados y multivariados, modelan observaciones las cuales se asume que han sido producidas por un conjunto de fuentes aleatorias alternativas e infieren los parámetros de estas fuentes para identificar qué fuente produjo cada observación, lo que lleva a un agrupamiento del conjunto de observaciones. Los métodos de agrupamiento no paramétricos pueden dividirse en tres grupos fundamentales: jerárquicos, particionales y basados en densidad.

Los algoritmos jerárquicos son aquellos en los que se va particionado el conjunto de datos por niveles, de modo tal que en cada nivel generalmente, se unen o se dividen dos grupos del nivel anterior, según si es un algoritmo aglomerativo o divisivo.

Los algoritmos particionales son los que realizan una división inicial de los datos en grupos y luego mueven los objetos de un grupo a otro según se optimice alguna función objetivo.

Los algoritmos basados en densidad enfocan el problema de la división de una base de datos en grupos teniendo en cuenta la distribución de densidad de los puntos, de modo tal que los grupos que se forman tienen una alta densidad de puntos en su interior mientras que entre ellos aparecen zonas de baja densidad.

Algoritmo Chameleon

Es un algoritmo jerárquico que cuenta de dos fases fundamentales. Durante la primera fase construye el grafo de los k vecinos más cercanos y usa un algoritmo de particionamiento de grafo para agrupar los puntos en subgrupos. Durante la segunda fase, usa un algoritmo jerárquico aglomerativo para encontrar los clústeres genuinos combinando repetidamente estos subgrupos. En esta segunda fase determina el par de subgrupos más similares tomando en cuenta su interconectividad y cercanía, éstas expresan las características internas de los subgrupos, el modelo no es estático, sino que es capaz de adaptarse a las características internas de los subgrupos según estos van cambiando. (40)

Algoritmo K-Means

Es uno de los más simples y conocidos algoritmos de agrupamiento, sigue una forma fácil y simple para dividir una base de datos dada en k grupos (fijados a priori). La idea principal es definir k centroides (uno para cada grupo) y luego tomar cada punto de la base de datos y situarlo en la clase de su centroide más cercano. El próximo paso es recalcular el centroide de cada grupo y volver a distribuir todos los objetos según el centroide más cercano. El proceso se repite hasta que ya no hay cambio en los grupos de un paso al siguiente. Este algoritmo cuenta de dos fases fundamentales:

- La **fase 1** utiliza actualizaciones por lotes, donde cada iteración consiste en la reasignación de puntos al centroide del clúster más cercano, todo a la vez, seguido del recálculo de los centroides del clúster.
- La **fase 2** usa actualizaciones en línea, donde los puntos se reasignan individualmente si al hacerlo se reduce la suma de distancias, y los centroides del grupo se vuelven a calcular después de cada reasignación.

El problema del empleo de este algoritmo es que fallan cuando los puntos de un grupo están muy cerca del centroide de otro grupo, también cuando los grupos tienen diferentes tamaños y formas (40, 41).

Algoritmo DBSCAN

DBSCAN es el primer algoritmo basado en densidad, se definen los conceptos de punto central (puntos que tienen en su vecindad una cantidad de puntos mayor o igual que un umbral especificado), borde y ruido. El algoritmo comienza seleccionando un punto *p* arbitrario, si *p* es un punto central, se comienza a construir un grupo y se ubican en su grupo todos los objetos denso-alcanzables desde p. Si p no es un punto central se visita otro objeto del conjunto de datos. El proceso continúa hasta que todos los

objetos han sido procesados. Los puntos que quedan fuera de los grupos formados se llaman puntos ruido, los puntos que no son ni ruido ni centrales se llaman puntos bordes (42).

La literatura abordada (36, 38) recomienda la utilización del algoritmo K-Means como mejor apuesta para el caso es cuestión, pues a pesar de presentar sensibilidad ante el ruido en los datos, ofrece buenos resultados en un espacio de tiempo pequeño en comparación con otros algoritmos.

1.6 Ambiente de desarrollo

A continuación se describen las herramientas y tecnologías a utilizar durante el desarrollo del sistema, así como una explicación de las principales características de estas herramientas, para arribar a conclusiones apropiadas sobre cuáles son las idóneas para resolver el problema en cuestión. Se explican los principales elementos de las metodologías de software, para seleccionar la más adecuada, donde se ofrecen las ventajas que tendría utilizar la escogida. Se argumenta acerca del lenguaje de programación a utilizar y el IDE de desarrollo, donde se hace una breve explicación de cada una.

1.6.1 Metodología de desarrollo

Las metodologías de desarrollo de software se emplean para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. Entre las metodologías que existen actualmente se encuentran las metodologías tradicionales, iterativas/evolutivas, las metodologías basadas en tecnologías web y las metodologías ágiles. Las Tendencias modernas en el desarrollo de software apuntan hacia el uso de metodologías más flexibles con un enfoque simple, donde el cliente está presente en todo el proceso de avance, estas son las metodologías ágiles mencionadas anteriormente.

Una de las principales es la metodología de Programación Extrema XP, la cual garantiza constar con una herramienta accesible al usuario, sencilla y a la misma vez dinámica. XP constituye un modelo de trabajo compartido, donde existe la conexión entre el cliente y el desarrollador, lo que permite la construcción de un sistema de acuerdo a los requerimientos establecidos por el cliente al principio de llevar a cabo el proyecto.

El grupo de investigación AIRI utilizó la metodología programación extrema (XP) durante desarrollo del software PANDOC por los beneficios que brinda para un grupo de desarrolladores pequeños. Además, XP es la metodología definida por dicho grupo de investigación, al cual pertenece la investigación.

Programación Extrema (XP)

XP es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y a realimentación o reutilización del código desarrollado. (51) Es una de las metodologías de desarrollo

de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo, equipos pequeños y cuyo plazo de entrega es inminente. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo de desarrollo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. (43)(44)

La metodología XP se basa en:

- Pruebas Unitarias: pruebas realizadas a los principales procesos, para detectar errores futuros.
- Refabricación: reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

¿Qué nos brinda XP?

- 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.

1.6.2 Herramientas

Entorno de desarrollo

Matlab (abreviatura de MATrix LABoratory, laboratorio de matrices), es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X. Es una poderosa herramienta para la resolución numérica de problemas. MATLAB ofrece un entorno interactivo sencillo mediante una ventana en la que se pueden introducir órdenes en modo texto y en la que aparecen los resultados. Los gráficos se muestran en ventanas independientes. Cada ventana dispone de una barra de menús que controla su funcionalidad. Lo que distingue a MATLAB de otros sistemas de cálculo es su facilidad para trabajar con vectores y matrices. Las operaciones ordinarias, suma, producto, potencia, operan por defecto sobre matrices, sin más restricción que la compatibilidad de tamaños en cada caso. (45)

El software cuenta con un amplio abanico de herramientas especializadas denominadas t*oolboxes* que extienden significativamente la funcionalidad del programa principal. Estas herramientas cubren en la actualidad prácticamente todas las aéreas principales del mundo de la ingeniería y la simulación. La *toolbox* para el procesado digital de imágenes es una de sus herramientas más famosas y utilizadas, y está formada por un conjunto de funciones que aplican las capacidades de MATLAB para el desarrollo de aplicaciones y algoritmos en el campo del procesado y análisis digital de imágenes. Entre sus prestaciones básicas se hallan: la manipulación de matrices, el trabajo con imágenes, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes. Matlab dispone además de las herramientas: GUIDE (editor de interfaces de usuario GUI por sus siglas en inglés) y *Simulink* (plataforma de simulación multidominio). También se pueden ampliar las capacidades de Matlab con las cajas de herramientas (t*oolboxes*); y las de *Simulink* con los paquetes de bloques (*blocksets*). Es un software muy usado en universidades y centros de investigación y desarrollo. (45)

Características principales:

- Lenguaje de alto nivel para cálculo técnico.
- Entorno de desarrollo para la gestión de código, archivos y datos.
- Funciones matemáticas para álgebra lineal, estadística, análisis de Fourier, filtraje, optimización e integración numérica.
- Herramientas para crear interfaces gráficas.
- Funciones para integrar los algoritmos basados en Matlab con aplicaciones y lenguajes externos, tales como: C/C++, FORTRAN, Java, COM y Microsoft Excel.

Ventajas de Matlab:

- Posee una excelente ayuda incorporada.
- Es multiplataforma.
- Alta precisión en el cálculo y procesamiento de imágenes.
- Colección útil de funciones predefinidas.
- Comunidad de usuarios muy extendida
- Potente herramienta matemática

La principal desventaja de Matlab es que es un software propietario pero a pesar de este inconveniente se decidió el uso de esta herramienta principalmente por las ventajas que ofrece, ya que es una

herramienta multiplataforma, con alta precisión en el cálculo y en el procesamiento de imágenes, y tiene incorporado una excelente ayuda.

Herramienta para el modelado.

Como herramienta para el modelado de la solución se ha seleccionado Visual *Paradigm for UML* (VPUML 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 captura de requisitos, planificación de software, planificación de controles, modelado de clases y modelado de datos.

Herramienta utilizada para la realización de las pruebas unitarias

Las pruebas de algoritmos escritos en MATLAB se realizan mediante *frameworks* externos, sobre todo se emplean marcos del tipo xUnit. Con la versión R2008a se desarrolló un primer intento de marco de pruebas propio conocido como MATLAB xUnit Test Framework, pero resultó muy limitado y su uso fue escaso ya que no formaba parte del paquete de Matlab. Con la versión R2013a, se desarrolló y distribuyó como parte integrante de MATLAB un nuevo *framework*, MATLAB *UnitTesting*, su código es abierto y puede encontrarse en el paquete “matlab.unittest”. MATLAB UnitTesting permite a los usuarios escribir pruebas unitarias de forma automatizadas, administrarlas y verificar los resultados esperados de las pruebas contra los resultados reales. El *framework* permite escribir pruebas unitarias en tres formas distintas scripts, funciones y clases, donde cada tipo de prueba permite niveles de complejidad distintos. Las pruebas en forma de scripts se emplean para comprobar que los resultados producidos por scripts, funciones y clases son los esperados. Sólo dos funciones están permitidas en estas pruebas *assert* para comprobar el valor, tipo, tamaño de los resultados, etc y *runtests*, para ejecutar las pruebas. Las pruebas en forma de funciones también se emplean para comprobar que los resultados producidos por scripts, funciones y clases son los esperados. Las pruebas en forma de funciones se basan en una función principal acompañada de tantas funciones locales como sea necesario. Por claridad, cada función suele asociarse con un caso de prueba. Las pruebas en forma de clases permiten test paramétricos, etiquetar las pruebas o emplear bloques. (45)

Herramienta para el análisis estadístico utilizando SPSS

El programa estadístico SPSS (del inglés *Statistical Package for the Social Sciences*) es uno de los programas más conocidos teniendo en cuenta su capacidad para trabajar con grandes bases de datos y una sencilla interfaz para la mayoría de los análisis. Familiarizarse con las diversas opciones y procedimientos estadísticos de un programa como SPSS permite la realización de varios procedimientos estadísticos, brindando gráficos, ejemplos de análisis de hipótesis en una interfaz sencilla y con una curva de aprendizaje pequeña.(46)

1.6.3 Lenguaje de Programación

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

Conclusiones parciales

Al término del desarrollo del marco teórico se pudieron arribar a las siguientes conclusiones:

- El algortimo K-means es recomendado para trabajar con grandes cúmulos de datos, por ello se decidió utilizar para realizar el agrupamiento de las regiones con texturas similares.
- Se profundizó en los principales métodos de segmentación de imágenes por texturas, específicamente en la matriz de coocurrencia; la cual destaca por sus características como una buena opción en la detección de opacidad en la cápsula posterior.
- Se decidió utilizar la metodología XP por las bondades que brinda al proceso de desarrollo y por ser la definida por el grupo de investigación (AIRI).
- Se seleccionaron y describieron las herramientas y tecnologías a ser usadas seleccionando Matlab como lenguaje de programación e IDE de desarrollo, Visual Paradigm como herramienta de modelado y SPSS para la validación estadística.

Capítulo 2. Herramienta para la segmentación de imágenes en retroiluminación con opacidad en la cápsula posterior

En el presente capítulo se presentan los procedimientos llevados a cabo en la propuesta de solución para identificar la OCP en imágenes en retroiluminación provenientes de la lámpara de hendidura, se describe las fases de la metodología XP: planificación, diseño y codificación, obteniéndose artefactos importantes como las Historias de Usuarios, Plan de Iteraciones, Plan de Duración de Iteraciones y Plan de Entregas. También durante el desarrollo del capítulo se describe la fase de Diseño basándose fundamentalmente en el desarrollo de las tarjetas Clase-Responsabilidad-Colaboración (CRC).

2.1 Características de la imagen

Las imágenes médicas que proceden de la lámpara de hendidura se encuentran en el espacio de color RGB, por lo que están compuestas por tres canales de color (rojo, verde, azul). Las imágenes obtenidas tienen una profundidad de 24 bits (8 bits por cada uno de los canales del espacio de color), por lo que presenta variaciones de intensidad entre 0 y 255 niveles de color en cada uno de los canales. Estas fueron comprimidas mediante formato JPG, dicho formato es un algoritmo diseñado para comprimir imágenes con 24 bits de profundidad o en escala de grises. Como resultado de la compresión, la imagen obtenida no es la misma que la deseada, esto es provocado por el algoritmo que utiliza el formato JPG, que es un algoritmo de reducción con pérdida para disminuir el tamaño del archivo.

Estas imágenes presentan una resolución espacial de 483 x 333 píxeles. Para explorar los elementos profundos del globo ocular es necesario aplicar una luz de alta intensidad en el ojo, lo que provoca que se refleje en partes de las estructuras a analizar (como se puede apreciar en la figura 7). Puede ser directa, reflejando la luz directamente sobre la estructura a examinar, o indirecta, donde la estructura a observar se ve contra un fondo oscuro. 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, pérdida de información en zona que ocupa el haz de luz y que existan bordes que no estén bien definidos.

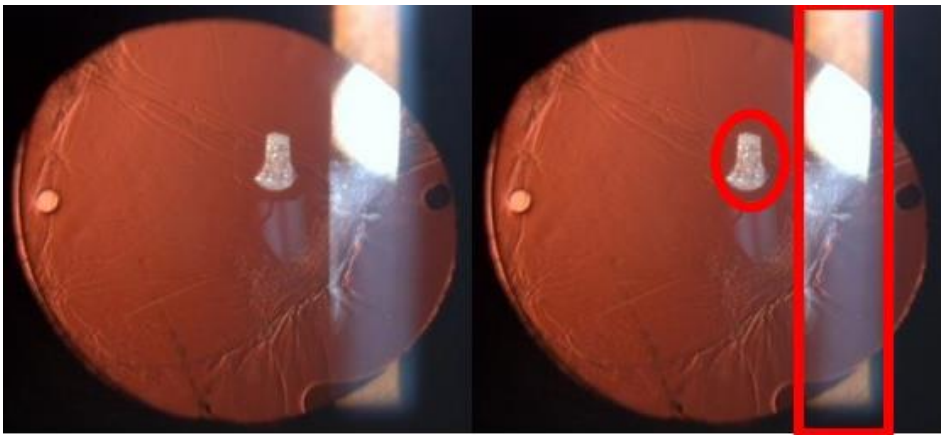


Figura 7. Imagen retroiluminación (en rojo las regiones con alta incidencia de la luz)

2.2 Propuesta de solución

Para el diseño de la propuesta de solución se seleccionaron algunos de los métodos de procesamiento de imágenes expuestos en el capítulo anterior. Se utilizó la matriz de coocurrencias, esta permite obtener las variables de texturas necesarias para realizar la segmentación, con el objetivo de obtener una solución adecuada que permita identificar la OCP en pacientes operados de catarata. Teniendo en cuenta las características esenciales de las imágenes y haciendo uso del entorno de computación Matlab 2014, se seleccionaron una secuencia de pasos modelados en un diagrama de actividades (como se muestra en la figura 8), en aras de alcanzar una alta precisión en la segmentación imágenes médicas.

La herramienta cuenta con una interfaz donde el usuario puede configurar algunos parámetros con el objetivo de personalizar la segmentación en caso de que la imagen necesite tratamiento especial. Además, se muestran los resultados obtenidos durante el proceso de segmentación (imagen en escala de grises, imagen binaria de regiones con opacidad e histogramas de las variables de texturas), propiciándose varios niveles de información para los usuarios. Una vez realizada la segmentación y obtenido todos resultados esperados, el usuario puede guardar la imagen resultante de la segmentación en formato jpg para su posterior revisión.

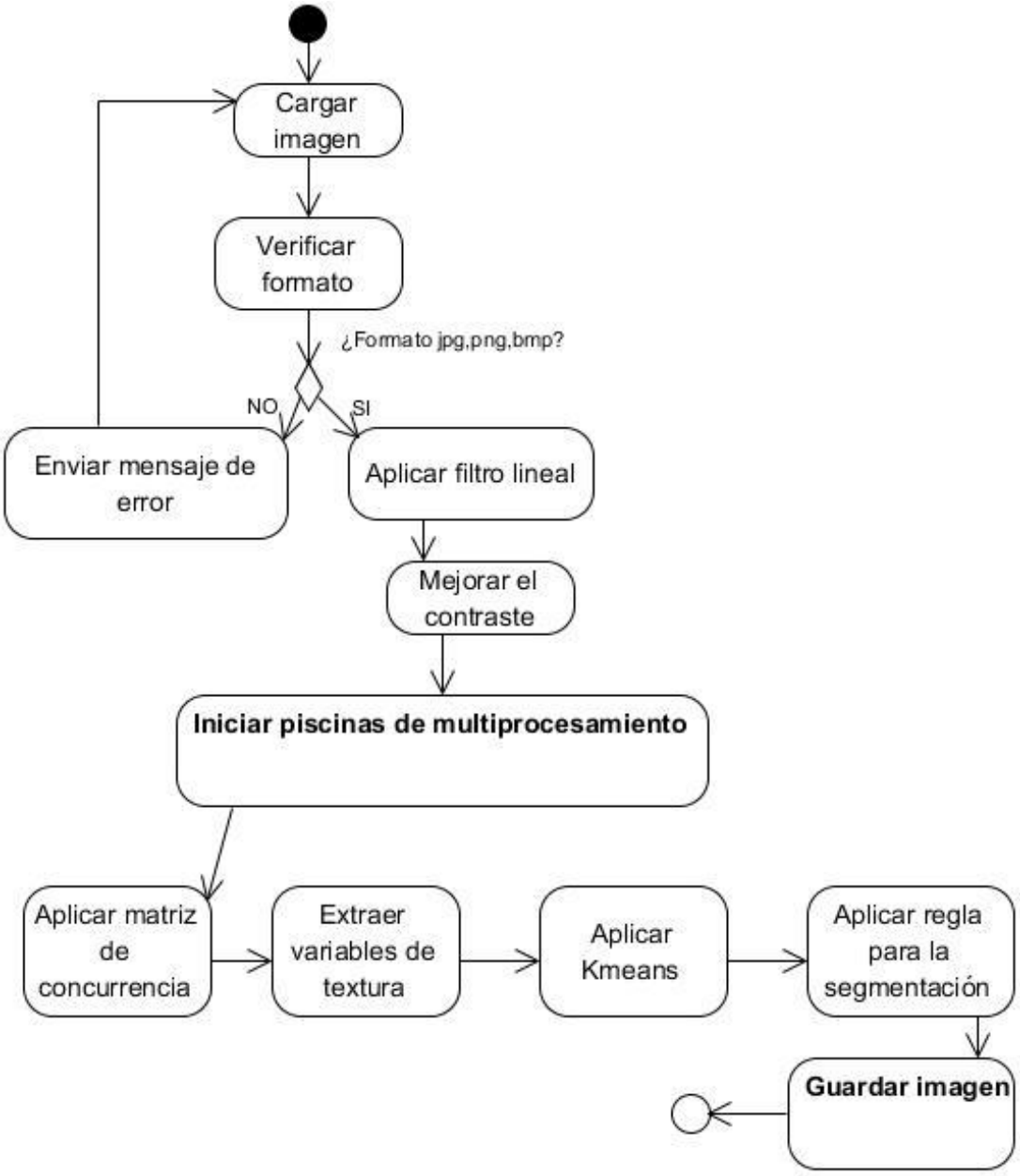


Figura 8. Diagrama de actividades para identificación de opacidad capsular en imágenes en retroiluminación mediante detección de texturas. Fuente: Elaboración propia

Las diferentes acciones que se pueden llevar a cabo en la propuesta de solución se explican a continuación:

2.2.1 Explorar los archivos y cargar la imagen médica.

En el primer paso se procede a explorar los archivos y cargar la imagen oblicua proveniente de la lámpara de hendidura en los formatos jpg, png o bmp para su posterior análisis y procesamiento, como se muestra en la Figura 9:

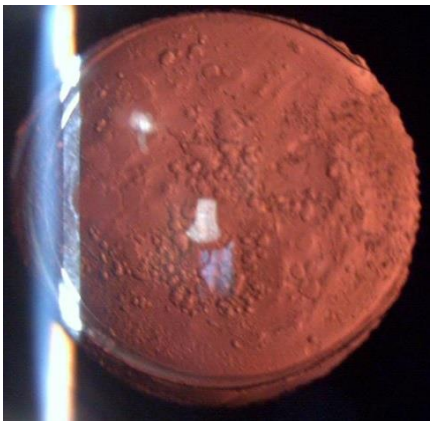


Figura 9 Imagen en retroiluminación.

2.2.2 Aplicar filtro lineal para la reducción de ruido.

Para eliminar el ruido introducido en el proceso de captura de las imágenes se selecciona una técnica de filtrado lineal empleando un algoritmo de análisis y mejora. Para ello se aplica la función *imadjust,* que permite ajustar los niveles de intensidad de la imagen, dicha función requiere como parámetros la imagen a tratar y el kernel o filtro (creado previamente) que será utilizado para tratar la imagen. El filtro es generado utilizando la función fspecial, esta necesita un grupo de configuraciones para lograr su objetivo:

- **Tipo de filtro:** gaussiano de paso bajo
- **Tamaño del kernel:** [15 15]
- **Intensidad:** 0.8

Cada filtro tiene sus propias características y ofrece resultados específicos, en este caso el filtro gaussiano disminuye considerablemente el ruido, pero crea un poco de distorsión en la imagen. La distorsión generada es tratada con la aplicación de un nuevo filtro de realce que se lleva a cabo utilizando la función *imsharpen*. Dicha función resalta los detalles de la imagen de acuerdo con los parámetros establecidos; para una correcta utilización de la misma y la obtención de los mejores resultados, se definieron como parámetros:

- **Fortaleza del efecto de realce:** 5
- **Desviación estándar del filtro** :5

Luego de la aplicación de los filtros anteriormente mencionados, se obtiene una imagen mejorada como se muestra en la figura 10, donde se puede observar una disminución considerable de ruido y un aumento de los detalles de la misma.

Figura 10. Imagen filtrada

2.2.3 Mejorar contraste de la imagen

Para mejorar el contraste de la imagen cargada, se propone expandir el histograma o realizar una ecualización del mismo buscando una distribución uniforme con el objetivo de potenciar detalles y características de la imagen. Para ello se utilizan las funciones *adaphisteq,* stretchlim, definidas en la caja de herramienta de Matlab 2014 llamada “Procesamiento de imágenes”, que mejoran el contraste de la imagen utilizando una ecualización adaptativa con contraste limitado y amplía los límites de intensidad de la imagen. La función trabaja con imágenes en escala de grises y puede ser del tipo *uint8, uint16, int16, single* o *double* que son los tipos de datos que contendrá una imagen en Matlab.

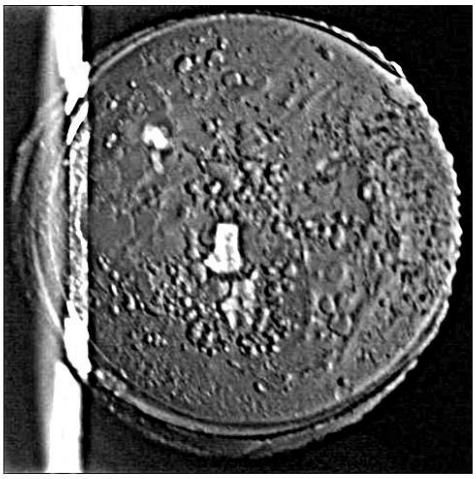


Figura 11. Imagen mejorada

2.2.4 Establecimiento de procesamiento en paralelo

El Multiprocesamiento o procesamiento en paralelo es el uso de dos o más procesadores (CPU) en una computadora para la ejecución de uno o varios procesos (programas corriendo). El procesamiento en paralelo permite a múltiples procesos compartir una única CPU, múltiples CPU pueden ser utilizados para ejecutar múltiples procesos o múltiples hilos de ejecución dentro de un único proceso. Se utiliza el procesamiento en paralelo que presenta la herramienta Matlab para aumentar la eficiencia del algoritmo y disminuir el tiempo de procesamiento de las imágenes digitales.

La herramienta Matlab en su configuración por defecto tiene creada una piscina para el procesamiento en paralelo, para su utilización es necesario definir sobre que partes del algoritmo se desea realizar dicho el procesamiento. El multiprocesamiento en el algoritmo se utiliza en el ciclo donde se aplica la matriz de coocurrencia, que es el que mayor complejidad temporal presenta, para ello se define el ciclo más externo como ciclo en paralelo utilizando el código "parfor".

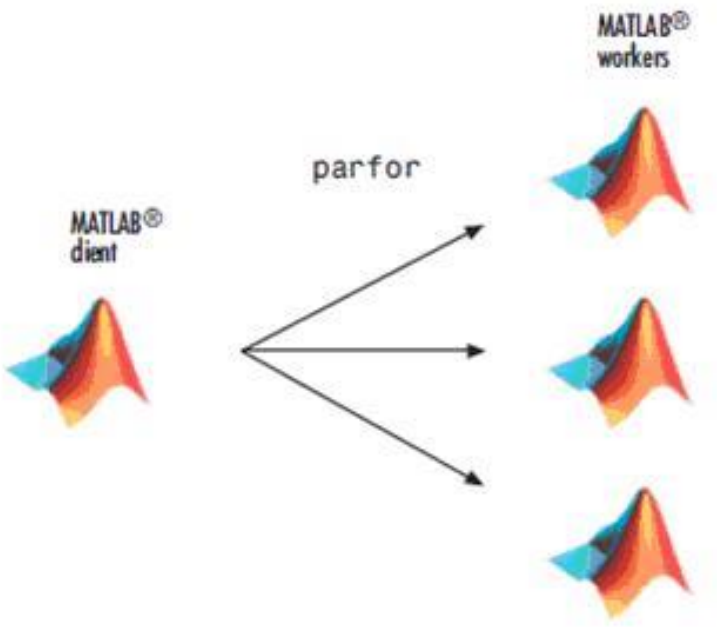


Figura 12. Procesamiento en paralelo en MATLAB

2.2.5 Aplicar matriz de coocurrencia.

En esta etapa se aplica la matriz de coocurrencia a la imagen con el objetivo de obtener las variables de texturas necesarias para la segmentación. Para ello primeramente se subdivide la imagen en matrices de NxN píxeles (N es un número impar), donde el pixel del medio se va corriendo en una posición y se obtiene una submatriz por cada píxel de la imagen original (ver sección 1.3). Luego se utiliza la función

graycomatrix para la obtención de la matriz de coocurrencia relacionada a cada submatriz. Esta función presenta un grupo de configuraciones que permiten la obtención de distintos resultados, estas son:

- **Límites de gris:** que permite establecer el nivel mínimo y el máximo de gris.
- **Niveles de gris:** define los niveles de gris.
- **Direcciones:** Define las direcciones que se van analizar con respecto al pixel analizado y simetría en caso que la matriz sea simétrica.

Con el objetivo de disminuir la complejidad temporal y el número de configuraciones de la función se decide utilizar el parámetro direcciones (como se observa en la figura 13). Los restantes parámetros en su configuración por defecto ofrecen los resultados correctos.

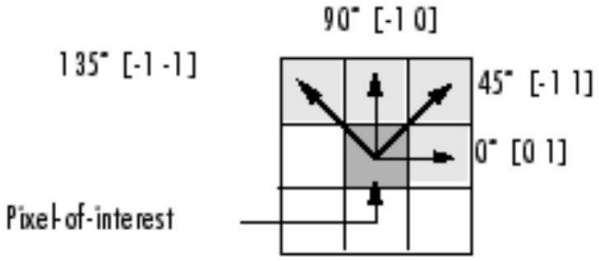


Figura 13.Direcciones utilizadas

2.2.6 Obtención de variables de textura

Para extraer las variables de textura de cada matriz de coocurrencia se utiliza la función *graycoprops*, que necesita los parámetros: matriz de coocurrencia y los nombres de las variables de textura a extraer de dicha matriz. Luego de la aplicar la función se obtiene una matriz por cada variable de textura seleccionada, que consta de las mismas dimensiones de la imagen inicial.

Las variables de textura utilizadas son:

- **Contraste:** Devuelve una medida del contraste de intensidad entre un píxel y su vecino sobre la imagen completa. En la figura 14 como puede observar el resultado de la variable contraste.

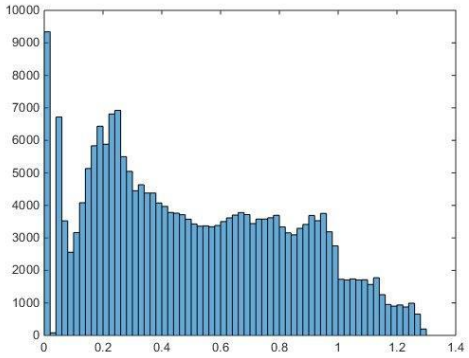


Figura 14. Histograma de contraste. Fuente: herramienta para detección OCP utilizando matriz de coocurrencia.

- **Homogeneidad:** Devuelve un valor que mide la proximidad de la distribución de elementos en el GLCM a la diagonal GLCM. En la figura 15 se puede observar el resultado de la variable homogeneidad

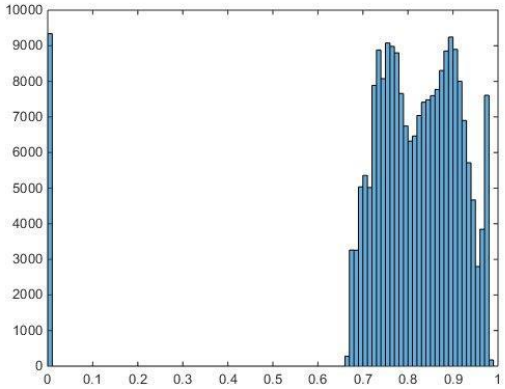


Figura 15 Histograma de homogeneidad. Fuente: herramienta para detección OCP utilizando matriz de coocurrencia.

- **Correlación:** Devuelve una medida de qué tan correlacionado está un píxel con su vecino sobre toda la imagen. En la figura 16 se puede observar el resultado de la variable correlación.

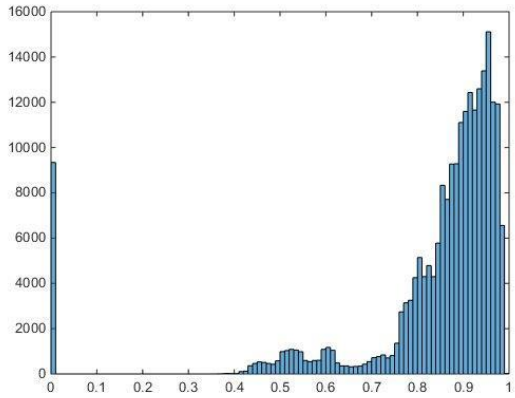


Figura 16 Histograma de correlación. Fuente: herramienta para detección OCP utilizando matriz de coocurrencia.

2.2.7 Aplicar algoritmo de agrupamiento.

K-Means usa un algoritmo iterativo de dos fases para minimizar la suma de las distancias punto a centroide y determinar los agrupamientos finales.

La primera fase utiliza actualizaciones por lotes, donde cada iteración consiste en la reasignación de puntos al centroide del clúster más cercano, todo a la vez, seguido del recalculo de los centroides del clúster. Esta fase ocasionalmente no converge a la solución que es un mínimo local. Es decir, una partición de los datos donde mover un solo punto a un grupo diferente aumenta la suma total de distancias. Esto es más probable para pequeños conjuntos de datos. La fase del lote es rápida, pero potencialmente solo se aproxima a una solución como punto de partida para la segunda fase.

La segunda fase usa actualizaciones en línea, donde los puntos se reasignan individualmente si al hacerlo se reduce la suma de distancias, y los centroides del grupo se vuelven a calcular después de cada reasignación. Cada iteración durante esta fase consiste en un pase a través de todos los puntos. Esta fase converge a un mínimo local, aunque podría haber otros mínimos locales con menor suma total de distancias. En general, encontrar el mínimo global se resuelve mediante una selección exhaustiva de puntos de partida, pero el uso de varias repeticiones con puntos de partida aleatorios generalmente da como resultado una solución que es un mínimo global.

Para realizar la agrupación de texturas se utiliza el método K-Means a través de la función *kmeans* que necesita de los parámetros:

- **Matriz:** una matriz donde cada columna corresponde a un vector con los valores de las variables de texturas.

- **Número de clústeres:** cantidad de clústeres en los que se van a agrupar las texturas de la imagen.

Luego de la aplicación del algoritmo K-Means se obtiene un conjunto de valores relacionados a la posición de los clústeres y las distancias de los puntos con respecto a estos, dichos valores son obtenidos como:

- **Idx**. Índices de clúster, devueltos como un vector de columna numérica. idx tiene tantas filas como X, y cada fila indica la asignación de clúster de la observación correspondiente.
- **C.** Ubicaciones del centroide de los clústeres, devueltas como una matriz numérica. C es una matriz $k\ x\ p$.
- **D.** Distancias desde cada punto a cada centroide, devuelto como una matriz numérica. D es una matriz n-by-k, donde el elemento (j, m) es la distancia desde la observación j al centroide m.
- **sumd.** Sumas dentro del conglomerado de distancias punto a centroide, devueltas como un vector de columna numérica. **sumd** es un vector $k\ x\ 1$, donde el elemento j es la suma de las distancias punto a centroide.

2.2.8 Definir regla para la segmentación

Para la creación de la regla se realizaron pruebas sobre un grupo de 12 imágenes provenientes de lámparas de hendiduras donde seleccionaron los valores (posición de centroides) que mejores resultados mostraron (ver figura 17). La selección realizada anteriormente se utiliza como punto de partida para la detección de zonas de opacidad en imágenes retroiluminación y su posterior segmentación. A continuación, se muestra los mejores resultados obtenidos junto con, la ubicación de sus clústeres:

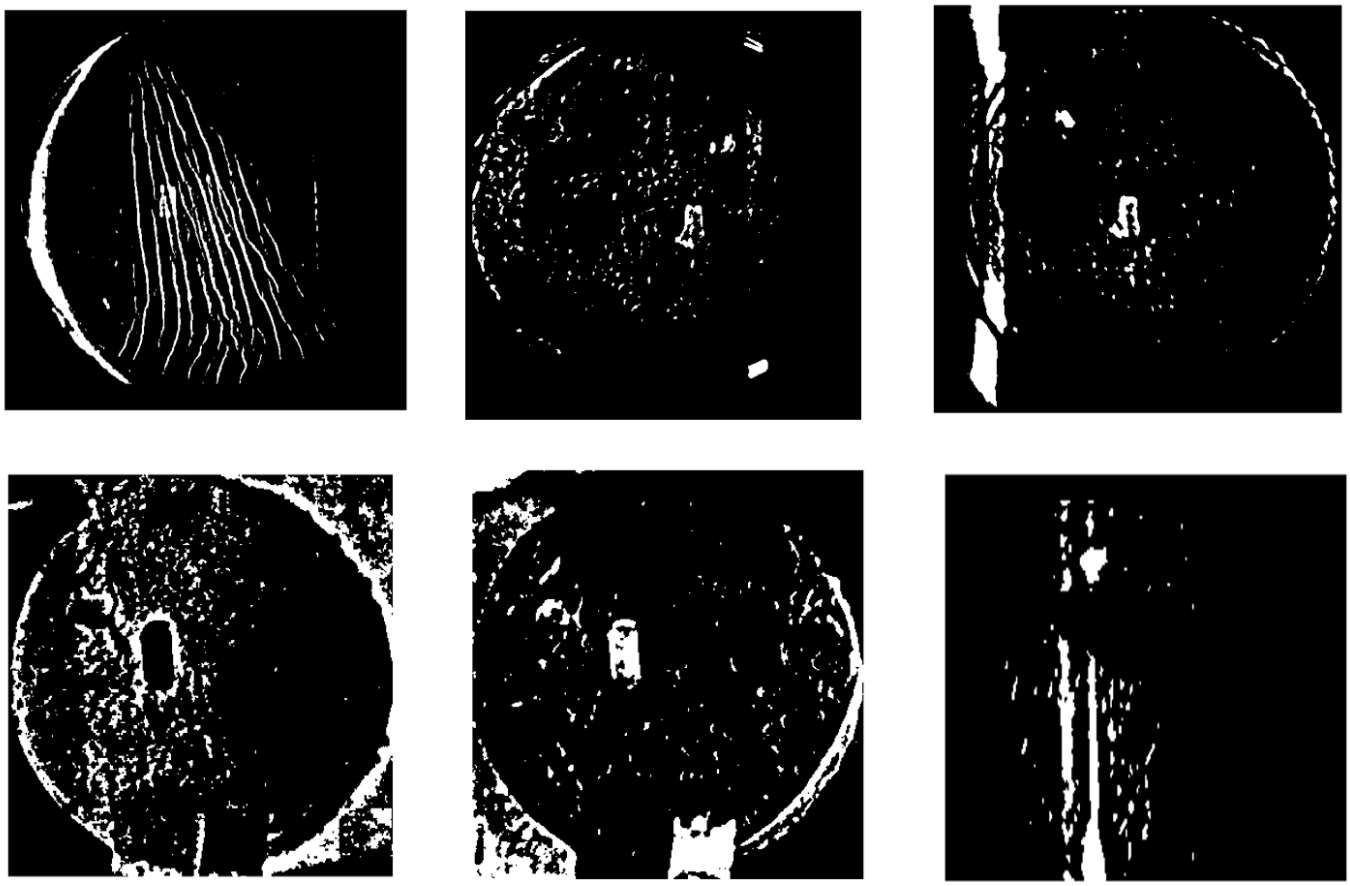


Figura 17. Mejores resultados luego de aplicar K-Means

La tabla 2 muestra los valores de los centroides que ofrecieron los mejores resultados.

Tabla 2. Posiciones de centroides

Posiciones de centroides	
0.942699973880266	0.892549355073405
0.940272506852241	0.892462292734488
0.952459413738767	0.892855783727682
0.957811321644064	0.937925779603065
0.963585434173690	0.854672814158781
0.977009990781586	0.931967537701336
0.973790217841534	0.846816088116201

Partiendo de la información anterior se crea un nuevo centroide que funciona como punto de selección de zonas con opacidad, para ello se calcula el promedio de cada columna obteniéndose el punto A = [0.9582, 0.8927].

Los pasos que definen la regla son:

- Se crea un centroide (A) como valor representativo de la región de opacidad a partir de los resultados anteriores.
- Se calcula la distancia entre cada centroide y el centroide A.
- Se seleccionan los centroides que presenten una distancia menor que 0,1.
- Se crea una nueva matriz binaria con las mismas dimensiones de la imagen donde los valores que pertenezcan a los centroide seleccionados van a tener valor 1.
- Se detectan los bordes de las regiones de opacidad que se muestran en la matriz binaria.
- Se pintan de color blanco los bordes de las regiones de opacidad en la imagen original.

2.3 Fase de planificación

La Metodología XP define como fase inicial del desarrollo de software la planificación. Además, plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores. El proyecto comienza recopilando las historias de usuarios, donde una vez obtenidas los programadores evalúan rápidamente el tiempo de desarrollo de cada una.

2.3.1 Historias de usuarios

Durante el transcurso de esta etapa se realiza el proceso de identificación y elaboración de las historias de usuario (HU), además el equipo de trabajo se familiariza con las tecnologías y herramientas seleccionadas para el desarrollo. Según Kent Beck cada HU recoge al menos los siguientes aspectos.(47)

- **Número:** Posee el número asignado a la HU.
- **Nombre de HU:** Atributo que contiene el nombre de la HU.
- **Usuario:** El usuario del sistema que utiliza o protagoniza la HU.
- **Prioridad en el negocio**: Evidencia el nivel de prioridad de la HU en el negocio.
- **Riesgo de desarrollo**: Evidencia el nivel de riesgo en caso de no realizarse la HU.
- **Puntos estimados:** Este atributo no es más que una estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal

de trabajo. En la metodología XP está definida una semana ideal como 5 días hábiles trabajando 40 horas, es decir, 8 horas diarias, por lo que cuando el valor de dicho atributo es 0.5 equivale a 2 días y medio de trabajo, lo que se traduce en 20 horas.

- **Puntos reales:** Igual que el parámetro anterior, pero en este caso será el tiempo real en el que se realizó la HU.
- **Descripción:** Posee una breve descripción de lo que realizará la HU.

Durante esta etapa se identificaron un total de diez Historias de Usuarios, en las tablas 3 y 4 se muestran dos ejemplos de las HU obtenidas.

Tabla 3. Historia de usuario 4

Historia de usuario	
Número:4	**Nombre:** Obtención de la matriz simétrica de los niveles de grises
Usuario: Especialista	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 1	**Iteración asignada:** 1
Programador responsable: Jessie Romero Pérez	
Descripción: Se subdivide la imagen en matrices de 21x21 pixeles calculando las relaciones entre pixel en las 4 direcciones espaciales N, NE, E y SE. Donde se obtendrá una matriz con la cantidad de veces que se repiten las posibles combinaciones de los niveles de grises.	
Observaciones:	

Tabla 4. Historia de usuario 6

Historia de usuario	
Número:6	**Nombre:** Obtención de las variables de textura
Usuario: Especialista	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 2	**Iteración asignada:** 1
Programador responsable: Jessie Romero Pérez	
Descripción: Se normaliza la matriz de coocurrencia de nivel de gris (GLCM) de modo que la suma de sus elementos es igual a 1. Cada elemento (r, c) en el GLCM normalizado es la ocurrencia conjunta de probabilidad de pares de píxeles con una relación espacial definida que tiene valores de nivel de gris r y c en la imagen. Luego se calculan las estadísticas especificadas en las propiedades de la matriz de coocurrencia de nivel de grises normalizada.	

Observaciones:

2.3.2 Plan de entrega del proyecto.

A partir de las historias de usuario definidas para el desarrollo del sistema web se elaboró el siguiente plan de entrega que muestra las historias de usuario que se llevarán a cabo en cada iteración. Para este plan de entrega se ha tomado en cuenta la prioridad y el esfuerzo de cada historia de usuario.

Tabla 5. Estimación de esfuerzo por Historia de Usuario

No	Historias de usuarios	Puntos estimados (semanas)
1	Cargar Imagen	0.5
2	Aplicar filtro lineal	2,5
3	Mejorar contraste de la imagen	2,5
4	Procesamiento en paralelo	2
5	Obtención de la matriz simétrica de los niveles de grises	2,5
6	Obtención de matriz de probabilidad	3
7	Obtención de las variables de textura	2,5
8	Cálculo de distancia y clúster	3
9	Agrupamiento	2,5
10	Selección de los clúster más representativo.	2,5
11	Segmentación utilizando los clúster más representativo	2,5
12	Guardar la imagen	0,5

2.3.3 Plan de iteraciones

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración donde cada historia de usuario se traduce en tareas específicas de programación. Asimismo, para cada historia de usuario se establecen las pruebas de aceptación que se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, 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. (48)

En esta fase se definieron tres iteraciones para la realización del sistema:

Iteración 1

En esta iteración se realizan las HU que obtuvieron los primeros resultados en el sistema, relacionadas con los filtros para disminuir el ruido y mejora del contraste en las imágenes. Se realizan las HU Cargar Imagen, Aplicar Filtros Lineales y Mejorar contraste de la imagen.

Iteración 2

En esta iteración se realiza la HU relacionada con el proceso para obtener las variables de textura y se realizan las HU: Procesamiento en paralelo, Obtención de la matriz de los niveles de grises, Obtención de matriz de probabilidad y Obtención de las variables de textura.

Iteración3

En esta iteración se realizan las HU relacionadas con el agrupamiento de las variables de textura y la segmentación de las imágenes. Se realizan las HU: Cálculo de distancia y clúster, Agrupamiento, Selección de los clúster más representativo y Segmentación utilizando los clústeres más representativos.

Tabla 6. Plan de duración de las iteraciones

Iteración	Historias de usuarios	Duración (semanas)
1	Cargar Imagen	5.5
	Aplicar filtro lineal	
	Mejorar contraste de la imagen	
2	Procesamiento en paralelo	10
	Obtención de la matriz simétrica de los niveles de grises	
	Obtención de matriz de probabilidad	
	Obtención de las variables de textura	
3	Cálculo de distancia y clúster	11
	Agrupamiento	
	Selección de los clúster más representativo.	
	Segmentación utilizando los clúster más representativo	
	Guardar la imagen	

2.4 Fase de diseño

El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias. Si hay fallos en el diseño o malos diseños, estos deben de ser corregidos cuanto antes. El objetivo de esta fase de la metodología es conseguir diseños simples y sencillos con el objetivo de facilitar la comprensión por parte del usuario o cliente. Además, se persigue conseguir un diseño fácilmente entendible e impleméntale que a la larga costará menos tiempo y esfuerzo para desarrollarlo.

2.4.1 Tarjetas CRC

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) son utilizadas para representar las responsabilidades de las clases y sus interacciones. 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. Las tarjetas determinan el comportamiento de cada actividad. Como resultado del trabajo realizado durante esta fase se generan como artefactos 3 tarjetas CRC. A continuación, se muestran algunas de ellas.

Tabla 7. Tarjeta CRC#3

Tarjeta CRC	
Clase: **DetecciónNivelesGris**	
Responsabilidad	**Colaboración**
• Permite obtener las posibles combinaciones de los niveles de gris • Permite obtener la matriz de probabilidades de los niveles de grises • Permite obtener las variables de textura necesarias para el agrupamiento y la segmentación.	• Preprocesamiento de imágenes • Contraste

Tabla 8 Tarjeta CRC#4

Tarjeta CRC	
Clase: **Agrupamiento**	
Responsabilidad	**Colaboración**

• Permite obtener la pertenencia de cada píxel a un clúster específico. • Permite agrupar cada pixeles en el clúster más cercano a él.	• Detección Niveles Gris

2.5 Fase de codificación

En esta fase se establecen los estándares de codificación utilizados en la investigación para el desarrollo de la herramienta propuesta. Además, se presenta el pseudocódigo del algoritmo creado para la segmentación de imágenes en retroiluminación para la detección de opacidad en la cápsula posterior.

2.5.1 Estándares de codificación

La adopción de estándares de estilo y codificación son de vital importancia para asegurar la calidad del software. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Si bien los programadores deben implementar un estándar de forma prudente, este debe estar bien definido a nivel departamental, por tanto, al comenzar un proyecto de software es necesario establecer un estándar de codificación único para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada.

Las convenciones de código o estándares de codificación son importantes para los programadores por un gran número de razones.

- El 80% del costo del código de un programa va a su mantenimiento.
- Casi ningún software es mantenido toda su vida por el autor original.
- Las convenciones de código mejoran la lectura del software lo que permite entender código nuevo de manera más óptima y rápida.
- Si distribuyes tu código fuente como un producto, necesitas asegurarte de que está bien hecho y presentado como cualquier otro producto.

A continuación, se presentan algunos de los estándares de codificación definidos y aplicados al sistema:

- Se debe utilizar como idioma el español, las palabras no se acentuarán.
- Todos los ficheros fuentes deben comenzar con un comentario en el que se lista el nombre de la clase, información de la versión, fecha y copyright.
- Las líneas en blanco mejoran la facilidad de lectura separando secciones de código que están lógicamente relacionadas. Se deben usar siempre dos líneas en blanco en las siguientes circunstancias:

- Entre las secciones de un fichero fuente.
- Entre las definiciones de clases e interfaces.

- Se debe usar siempre una línea en blanco en las siguientes circunstancias:
  - Entre métodos.
  - Entre las variables locales de un método y su primera sentencia.
  - Antes de un comentario de bloque o de un comentario de una línea.
  - Entre las distintas secciones lógicas de un método para facilitar la lectura.
- Se debe dar un espacio en blanco en la siguiente situación:
  - Entre una palabra clave del lenguaje y un paréntesis.
- Respecto a las normas de inicialización, declaración y colocación de variables, constantes, clases y métodos:
  - Todas las instancias y variables de clases o métodos empezarán con minúscula. Las palabras internas que lo forman, si son compuestas, empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres guion bajo "_" o signo de peso "$", aunque ambos están permitidos por el lenguaje.
  - Los nombres de variables de un solo carácter se deben evitar, excepto para variables índices temporales.
  - Los nombres de las variables declaradas como constantes deben aparecer totalmente en mayúscula separando las palabras con un guion bajo ("_").
  - Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúscula. Mantener los nombres de las clases simples y descriptivas. Usar palabras completas, evitar acrónimos y abreviaturas.
  - Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula y la primera letra de las siguientes palabras que lo forman en mayúscula.
- Respecto a la identación y longitud de la línea:
  - Se deben emplear cuatro espacios como unidad de identación. La construcción exacta de la identación (espacios en blanco contra tabuladores) no se especifica. Los tabuladores deben ser exactamente cada ocho espacios.
  - Evitar las líneas de más de ochenta caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

2.5.2 Pseudocódigo

La naturaleza no específica de pseudocódigo simplifica en gran medida la fase de desarrollo del producto, ya que elimina muchas de las distracciones que podrían descarrilar fácilmente la parte inicial del proceso. Los desarrolladores pueden ver el cuadro completo, en lugar de los elementos específicos que componen ese marco. El pseudocódigo se puede entender incluso para los no programadores, permitiendo a las personas que no tienen conocimientos de informática, que se beneficien de sus aportes.

Tomados en conjunto, todas estas ventajas resultan en un proceso de desarrollo mucho más eficiente. El proyecto puede desarrollarse más rápidamente, ya que la planificación es más precisa. Además, se presentan menos problemas durante la codificación, por lo que se pierde menos tiempo solucionar el mismo código varias veces.

Para una mejor comprensión del algoritmo propuesto se diseñó el pseudocódigo el cual describe la propuesta de solución:

```
imagen ← Cargar imagen

imGris ← Convertir a escala de grises

x, y ← Dimensiones de la imagen

filtro1 ← Crear filtro gaussiano

Aplicar filtro gaussiano por convolución(filtro1)

Aplicar filtro de realce(imGris, fuerza ← 5 ,radio ← 5 )

Ecualizar histograma

direcciones ← establecer direcciones (N, NE, E, SE)

ancho ← 11

Iniciar piscinas de multiprocesamiento

Para i ← ancho Hasta x-ancho Con Paso 1 Hacer
```

```
Para j ← ancho Hasta y-ancho Con Paso 1 Hacer
        submatriz ← Seleccionar submatriz(limites)
        matrizCoocurrencia ← Aplicar matriz de coocurrencia(
            submatriz, direcciones)
        stats ← Extraer variables de textura(matrizCoocurrencia ,
            "contraste" , "homogeneidad", "correlación" )
        contr(i,j) ← Extraer contraste(stats)
        corr(i,j) ← Extraer correlación(stats)
        homo(i,j) ← Extraer homgeneidad(stats)
 Fin Para
Fin Para
vectorContr ← Convertir matriz en vector columna(contr)
vectorCorr ← Convertir matriz en vector columna(corr)
vectorHomo ← Convertir matriz en vector columna(homo)
Normalizar imagen
imgRec ← Recortar 6 pixeles de imGris
vectorImg ← Convertir matriz en vector columna(imgRec)

vimg ← Concatenar vectores(vectorCont, vectorCorr, vectorHomo,
    vectorImg)
cantClústeres ← 30
```

```
//km: asignación de clúster por pixel, clústeres: posiciones de los clústeres

km, clústeres ← Aplicar K-Means(vimg, cantClústeres)

matriz ← Convertir vector a matriz(km)

matriz ← Girar matriz izquierda a derecha

matriz ← Rotar matriz(90)

puntoInteres ← [0.9582, 08927]

selected ← SeleccionCluster(clústeres, puntoInteres)

Para textura ← 1 Hasta cantClústeres Con Paso 1 Hacer

 Para j ← 1 Hasta y Con Paso 1 Hacer

      Para i ← 1 Hasta x Con Paso 1 Hacer

            Si matriz(i,j)=textura Y selected(textura,1)=1 Entonces

            figuraOpacidad(i,j) ← 1

            Fin Si

      Fin Para

 Fin Para

Fin Para

bordes ← Detección de bordes(figuraOpacidad)

ultima ← imagen

Para i ← 1 Hasta x Con Paso 1 Hacer
```

```
Para j ← 1 Hasta y Con Paso 1 Hacer
    Si bordes(i,j)=1 Entonces
        ultima(i,j,1) ← 255
        ultima(i,j,2) ← 255
        ultima(i,j,3) ← 255
    Fin Si
 Fin Para
Fin Para
Pintar(imagen)
Pintar(ultima)
```

Conclusiones parciales:

- Se cargaron las imágenes en diversos formatos, se convirtió a escala de grises, se aplicaron filtros lineales y algoritmos de mejora de contraste para la reducción del ruido que presentan las imágenes en retroiluminación con el objetivo de tener comportamientos de textura más homogéneos.
- La aplicación de la matriz de coocurrencia permitió obtener las variables de textura utilizadas por al algoritmo k-means para realizar el agrupamiento de las regiones con texturas similares.
- EL procesamiento multihilo simplificó considerablemente el tiempo de ejecución del algoritmo lo que permitió el aprovechar al máximo los recursos de la máquina.
- La regla para la segmentación detectó correctamente las zonas de opacidad lo que permitió segmentar dichas zonas en la imagen original.

Capítulo 3. Resultados y validación

Durante el desarrollo del presente capítulo se estará transitando por las fases de implementación y prueba que propone la metodología XP, generando como artefacto en el caso de la fase de implementación las tareas de desarrollo que dan solución a cada una de las historias de usuarios identificadas en la fase de planificación. También se realizan un conjunto de pruebas al sistema, divididas en pruebas unitarias y de aceptación como propone la metodología y finalmente se realiza la validación a los resultados obtenidos utilizando la técnica de juicio de experto.

3.1 Fase de implementación

Se especifica en esta fase la implementación de las HU en su correspondiente iteración, obteniendo se en cada una de ellas una versión funcional del producto. Lo primero es hacer un chequeo de cada HU, en conjunto con el plan de iteraciones y se modifica en caso de ser necesario, para esto se crean tareas de desarrollo para de esta forma poder organizar la implementación. Estas tareas, al contrario de las HU escritas en el lenguaje del cliente, son escritas en un lenguaje técnico. Como parte de la planificación realizada en el capítulo anterior se detallan a continuación las iteraciones de desarrollo sobre el sistema.

3.1.1 Iteración 1

En esta iteración se implementan las funcionalidades que obtuvieron los primeros resultados en el sistema, relacionadas con los filtros para disminuir el ruido y mejorar el contraste en las imágenes:

Tabla 9. Puntos estimados de historias de usuarios de la iteración 1

No	Historia de Usuario	Puntos Estimados
1	Cargar Imagen	0.5
2	Aplicar filtro lineal	2,5
3	Mejorar contraste de la imagen	2,5

A continuación, se muestran las tareas de desarrollo realizadas para las funcionalidades de esta iteración:

Tabla 10. Tarea de desarrollo 1

Tarea	
Número de tarea:1	**Numero de Historia de usuarios :**1
Nombre de la tarea: Desarrollo de la funcionalidad para cargar imagen	
Tipo de tarea: configuración-desarrollo	**Puntos estimados:** 0.5
Fecha de Inicio:	**Fecha de fin:**
Programador responsable : Jessie Romero Pérez	
Descripción: Para cargar la imagen se utiliza la función ***imread*** la cual permite cargar la imagen en formato jpg,png y bmp.	

3.1.2 Iteración 2

En esta iteración se implementan las funcionalidades para obtener las variables de textura:

Tabla 11. Puntos estimados por historias de usuarios iteración 2

No	Historia de Usuario	Puntos Estimados
1	Procesamiento en paralelo	2
2	Obtención de la matriz simétrica de los niveles de grises	2,5
3	Obtención de matriz de probabilidad	3
4	Obtención de las variables de textura	2,5

A continuación, se muestran las tareas de desarrollo realizadas para las funcionalidades de esta iteración:

Tabla 12. Tarea de desarrollo 4

Tarea	
Número de tarea: 4	**Numero de Historia de usuarios :**4
Nombre de la tarea: Desarrollar funcionalidad para el procesamiento en paralelo	
Tipo de tarea: configuración-desarrollo	**Puntos estimados:**
Fecha de Inicio:	**Fecha de fin:**
Programador responsable : Jessie Romero Pérez	
Descripción: Para agilizar procesamiento de la herramienta se utilizó el multiprocesamiento, haciendo uso de las piscinas de procesamiento paralelo que brinda la herramienta MATLAB. Para ello se utilizó la estructura *parfor* donde se creó un ciclo de procesamiento en paralelo.	

3.1.3 Iteración 3

En esta iteración se realizan las HU relacionadas con el agrupamiento de las variables de textura y la segmentación de las imágenes. Se realizan las HU Cálculo de distancia y clúster, Agrupamiento, Selección de los clúster más representativo y Segmentación utilizando los clústeres más representativos.

Tabla 13. Asignación de puntos por historias de usuario iteración 3

No	Historia de Usuario	Puntos Estimados
1	Cálculo de distancia y clúster	2
2	Agrupamiento	2,5
3	Selección de los clúster más representativo.	3
4	Segmentación utilizando los clúster más representativo	2,5
5	Guardar la imagen	0,5

A continuación, se muestran las tareas de desarrollo realizadas para las funcionalidades de esta iteración:

Tabla 14. Tarea de desarrollo 8

Tarea	
Número de tarea:	**Numero de Historia de usuarios :**
Nombre de la tarea: Selección del clúster más representativo	
Tipo de tarea: configuración-desarrollo	**Puntos estimados:**
Fecha de Inicio:	**Fecha de fin:**
Programador responsable : Jessie Romero Pérez	
Descripción: Se analizan un grupo de imágenes para determinar los mejores resultados y determinar los valores representativos de las zonas con opacidad. A partir de los resultados se obtuvieron las posiciones de los clústeres que mejor agrupaban las texturas con opacidad y para determinar la posición más representativa se promediaron los valores de los clústeres.	

3.2 Interfaz del sistema

A continuación, se muestra la interfaz que permite visualizar los resultados de aplicar el algoritmo y establecer algunas configuraciones necesarias para la segmentación de las regiones con opacidad.

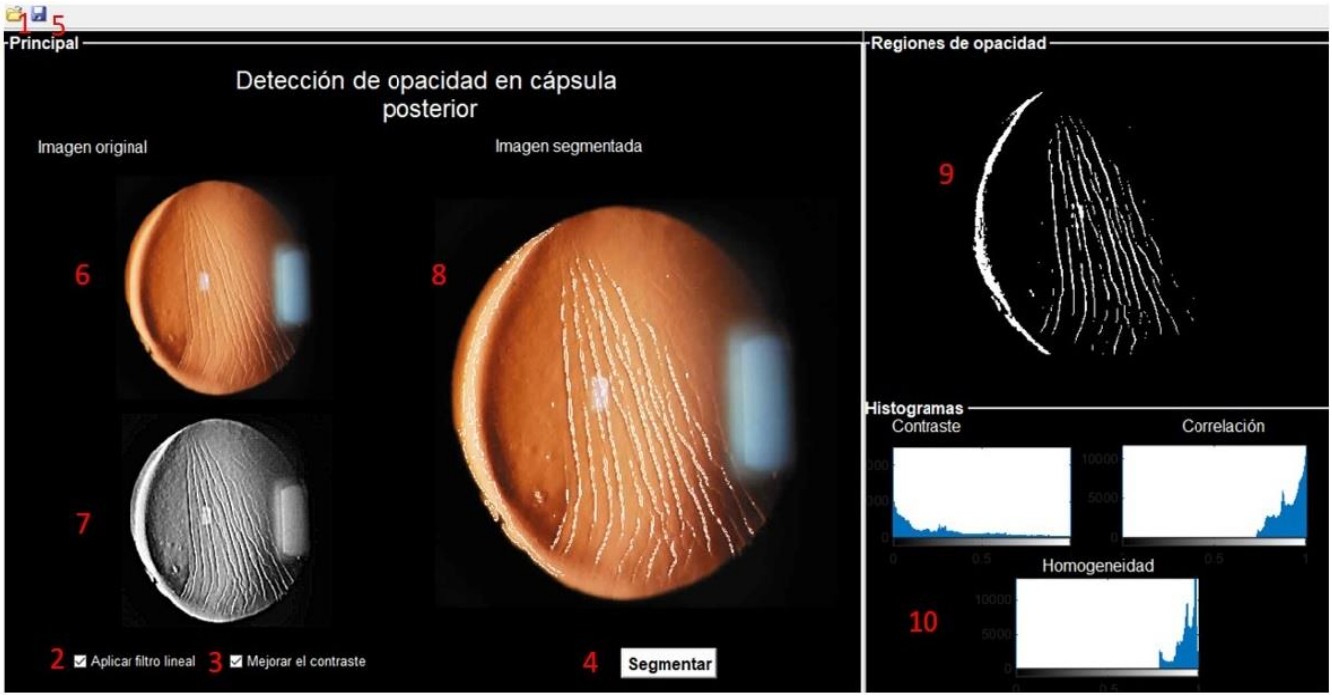


Figura 18. Interfaz del sistema

1. Permite cargar la imagen en formato jpg, png o bmp.
2. Permite aplicar filtro lineal.
3. Permite mejorar el contraste de la imagen.
4. Permite segmentar por textura.

5. Guarda los cambios realizados en la imagen.
6. Muestra la imagen cargada.
7. Muestra la imagen preprocesada.
8. Muestra la imagen segmentada.
9. Muestra las zonas de interés en una imagen binaria.
10. Muestra los histogramas de las variables de textura de la imagen.

3.3 Resultados de aplicar el algoritmo

El algoritmo fue aplicado a 12 imágenes en retroiluminación provenientes de la lámpara de hendidura. A continuación, se muestran los resultados obtenidos de aplicar este algoritmo a tres de estas imágenes. La figura 19 muestra dos columnas de imágenes donde la columna izquierda muestra las imágenes originales y la columna derecha las imágenes segmentadas.

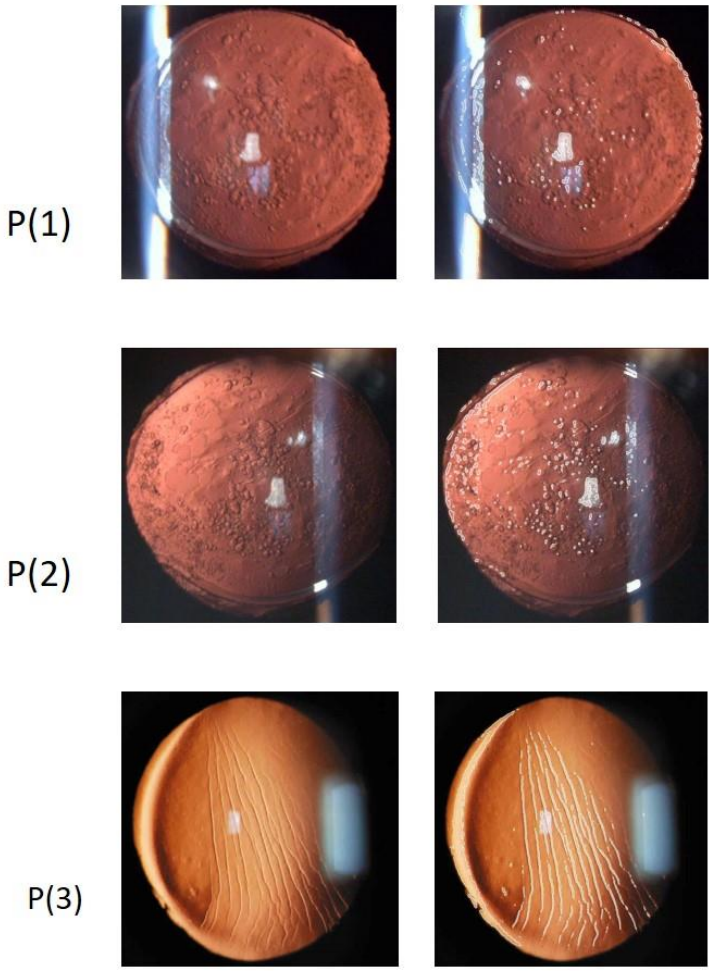


Figura 19. Resultados de aplicar el algoritmo

3.4 Pruebas

Uno de los pilares de la Programación Extrema (XP) es el proceso de pruebas. XP anima a probar constantemente tanto como sea posible, esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. XP divide las pruebas del sistema en dos grupos (47):

- Pruebas unitarias las cuales se concentran en los componentes individuales asegurándose que funcionen de manera apropiada como unidad
- Pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

3.4.1 Pruebas unitarias

Una prueba unitaria es la verificación de un módulo (unidad de código) determinado dentro de un sistema. Los programadores realizan estas pruebas cuando: la interfaz de un método no es clara, la implementación es complicada, para probar entradas y condiciones inusuales luego de modificar algo. Para poder integrar el código realizado al ya existente, el mismo debe aprobar satisfactoriamente todos los casos de prueba definidos. En XP los programadores deben escribir las pruebas unitarias para cada módulo antes de escribir el código. No es necesario escribir casos de prueba para todos los módulos, sólo para aquellos en que exista la posibilidad de que puedan fallar. Las pruebas se realizaron automatizadas utilizando el *framework* MATLAB Unit Testing, que permite escribir pruebas unitarias en tres formas distintas, Scripts, Funciones y Clases. Las pruebas en forma de scripts se emplean para comprobar que los resultados producidos por scripts, funciones y clases son los esperados. Las funciones que permiten realizar las pruebas son:

- Assert: para comprobar el valor, tipo, tamaño, etc, de los resultados.
- Runtests: para ejecutar las pruebas. Ejecutando el script de pruebas mediante la función runtests se asegura que se ejecutan todas las pruebas.

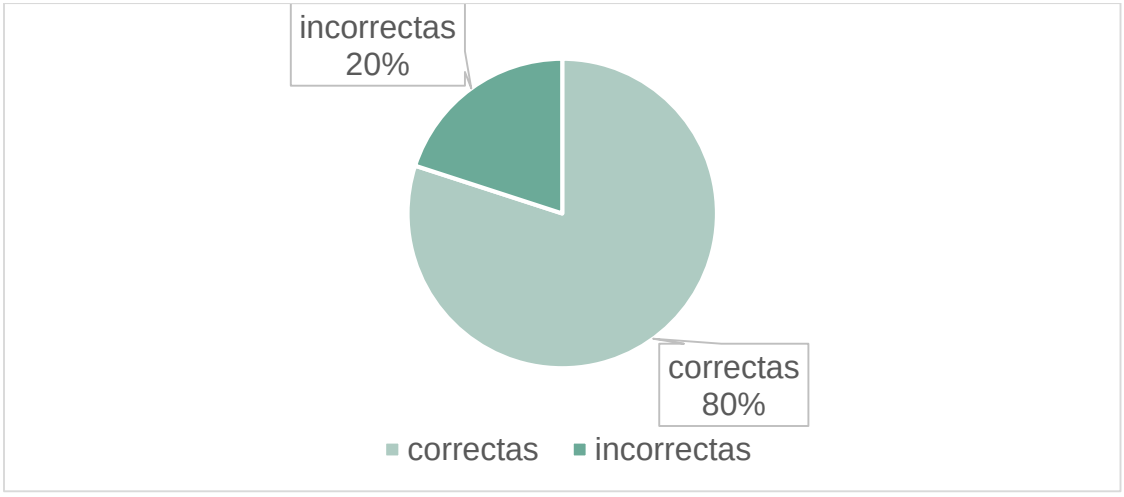


Figura 20. Resultados de aplicar el framework MATLAB Unit Testing.

3.4.2 Pruebas de caja blanca

Las pruebas de caja blanca son pruebas funcionales, estas son un método de diseño que usa la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba. Al emplear sus métodos, se puede garantizar que, al menos una vez, se ejecuten todas las rutas independientes del módulo. La técnica del camino básico permite derivar casos de prueba a partir de un conjunto de caminos independientes por los cuales puede circular el flujo de control (Pressman 2009)

Para obtener el conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática. Para poder elaborar el grafo de flujo, primero se deben enumerar las sentencias del código

A continuación, se muestra un fragmento de código perteneciente al algoritmo desarrollado:

```
parfor i=value:x-value 1
      for j=value:y-value 2
        img=imcrop(FR,[i-value j-value i+value j+value]);3
        mc=graycomatrix(img,'Offset',offsets);
        stats = graycoprops(mc,{'contrast','homogeneity','correlation'});

         Cont(i,j)=stats.Contrast(1);
         Corr(i,j)=stats.Correlation(1);
         Homo(i,j)=stats.Homogeneity(1);

      end
end 4
```

Figura 21. Fragmento de código. Fuente: elaboración propia

Posteriormente se procede a la elaboración del grafo de flujo teniendo en cuenta dicha enumeración.

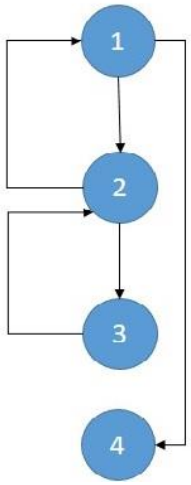


Figura 22. Representación del grafo de flujo de camino básico de aplicar matriz de coocurrencia. Fuente: elaboración propia

La complejidad ciclomática es la métrica de software con que se define la cantidad de caminos independientes de cada una de las funcionalidades del programa y provee el límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez (Pressman, 2009)

La complejidad ciclomática se basa en la teoría gráfica y se calcula de dos maneras distintas, donde, para que el cálculo sea correcto, todas deben arrojar el mismo resultado:

El número de regiones corresponde a la complejidad ciclomática “V (G)”.

V (G)= R Donde R es la cantidad total de regiones.

V (G)= 3

V (G)= E –N + 2 Donde E es el número de aristas y N número de nodos.

V (G)=5-4+ 2

V (G)= 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- 4

Camino básico 2: 1-2-4

Camino básico 3: 1-2-3-4

Cada camino independiente es un caso de prueba a realizar, de forma que se garantiza que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. En el caso anterior se calcularon cuatro caminos básicos, por tanto, surge la necesidad de hacer igual número de casos de prueba. A continuación, se muestra uno de los casos de pruebas realizados.

Tabla 15.Caso de prueba de caja blanca para el camino básico 3

Entrada	Imagen escala de grises y direcciones analizar.
Resultados Esperados	Se obtienen las variables de textura homogeneidad, contraste y correlación.
Condiciones	La imagen debe estar en escala de grises. Los valores de los pixeles de la imagen deben estar normalizados. La imagen debe pasar por un proceso de preprocesamiento.

3.4.3 Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las Historias de Usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son de gran importancia, dado que miden el grado de satisfacción del cliente con el producto desarrollado. Por lo tanto, son los clientes los responsables de verificar que los datos de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, son ellos los encargados de indicar el orden de resolución de los fallos [61]. Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información. La tabla 16 muestra un caso de prueba de aceptación utilizado durante el desarrollo de la investigación.

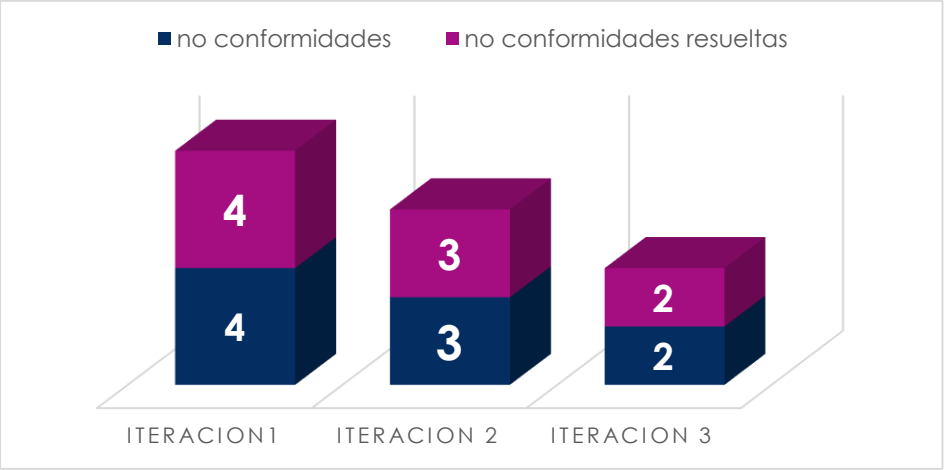
Tabla 16. Caso de prueba de aceptación

Caso de prueba de aceptación
Código: HU_P2 | Historia de Usuario: 11
Nombre: Segmentación utilizando los clúster más representativo
Descripción: Comprueba la funcionalidad que permite segmentar la opacidad de la cápsula posterior en las imágenes en retroiluminación.
Condiciones de Ejecución: Se debe haber cargado una imagen resultante de la lámpara de hendidura.
Entrada/Pasos de Ejecución: **• El usuario selecciona la imagen en formato jpg, bmp o png.** **• El usuario selecciona la opción de segmentar.** **• El sistema muestra la imagen segmentada donde la región con opacid**ad se muestra en color blanco
Resultado Esperado: **El sistema muestra la imagen segmentada, obteniendo las regione**s opacidad señalada en blanco
Evaluación de la Prueba: Prueba satisfactoria.

Resultados de las pruebas de aceptación

Por cada HU se realizó una prueba de aceptación. Fueron detectadas un total de 9 no conformidades en las 3 iteraciones realizadas, las cuales ya han sido solucionadas. Las no conformidades detectadas fueron:

1. Poco entendimiento de la herramienta.
2. La herramienta demora más de 1 hora en presentar resultados.
3. No se observan claramente las funcionalidades de la herramienta.
4. No se muestra la imagen en escala de grises.
5. No se puede observar claramente las regiones de opacidad en la imagen original
6. Faltas de ortografía.
7. La herramienta permite cargar cualquier tipo de fichero.
8. El color de la herramienta no es agradable.
9. No se muestra la imagen binaria con las regiones de opacidad segmentadas.



Como parte de estas pruebas se procedió a la creación de doce casos de estudio para verificar los resultados del software, en los mismos se tomaron imágenes de pacientes a los que fueron llamados (Paciente 1, Paciente 2..., Paciente 13). Estas pruebas arrojaron los siguientes resultados:

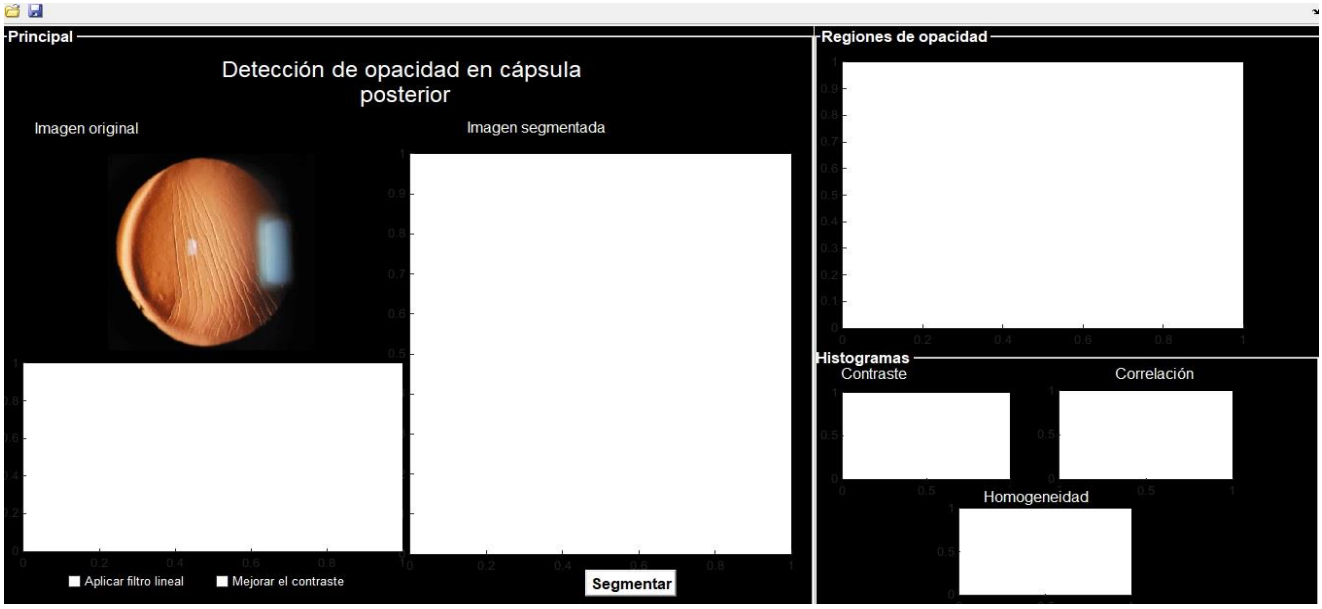


Figura 23. Imagen original cargada satisfactoriamente

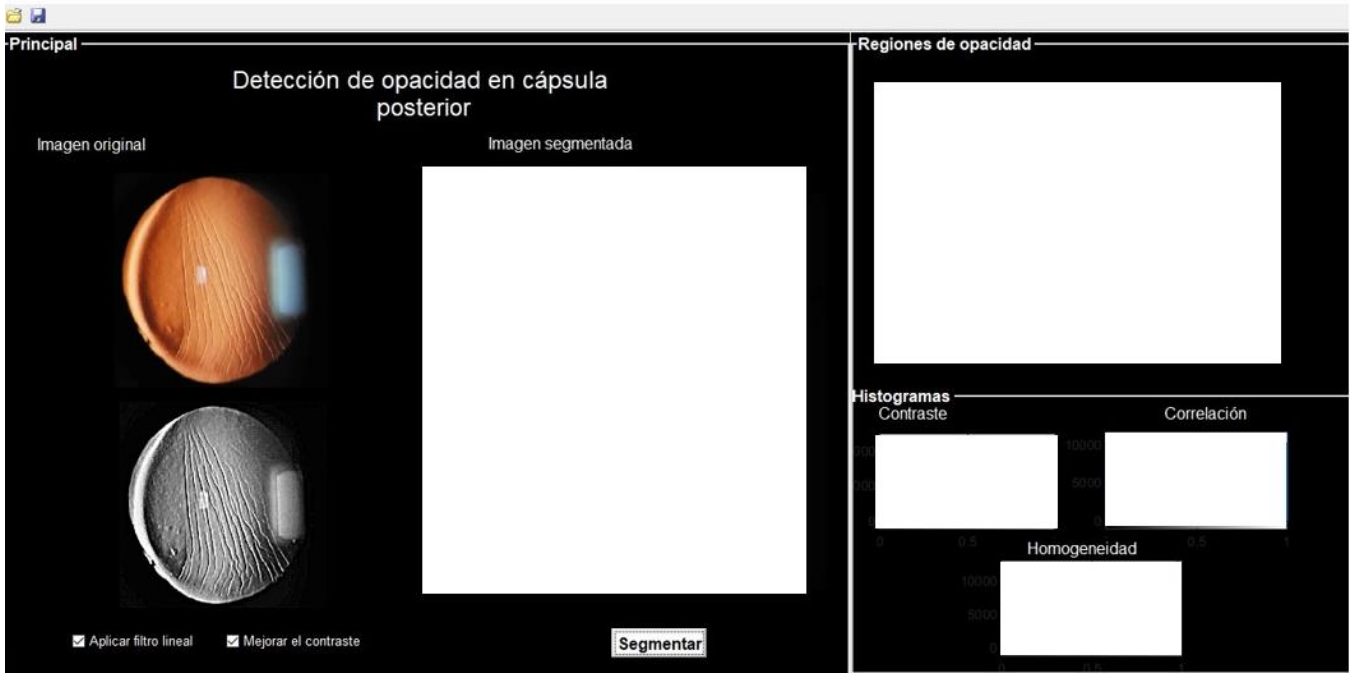


Figura 24. Muestra la imagen preprocesado

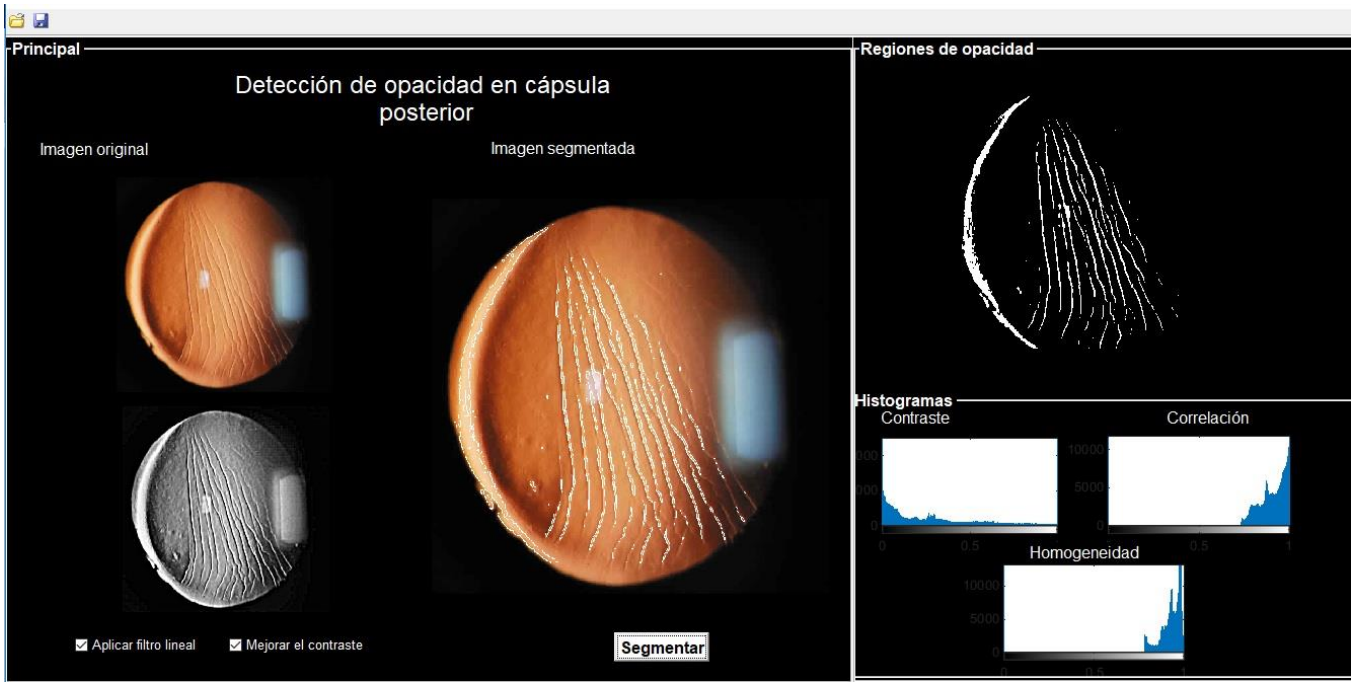


Figura 25. Muestra la imagen segmentada, las regiones de opacidad y el histograma de las variables de textura

3.5 Análisis de los resultados

Luego de efectuar las pruebas correspondientes para validar la fiabilidad del sistema, se procedió a realizar la comparación de las métricas de error y los tiempos de ejecución entre los algoritmos segmentación por texturas usando campo aleatorio de Markov en imágenes en retroiluminación y el algoritmo de segmentación por textura usando matriz de coocurrencia, con el fin de seleccionar el algoritmo más eficiente para la detección de la opacidad. Estas comparaciones fueron ejecutadas sobre un conjunto de doce imágenes de pacientes con esta complicación postoperatoria por cada uno de los algoritmos propuestos.

Para esta labor se confeccionó un equipo de tres especialistas del servicio de cataratas del ICO con amplia experiencia en el trabajo con la OCP:

- Dr. Iván Hernández López, Especialista de I grado en Oftalmología, Investigador Auxiliar.
- Dr. Eneida Pérez Candelaria, Especialista de II grado en Oftalmología, Profesor Auxiliar, Investigador Auxiliar.
- Dr. Zucell Veitía Rovirosa, Especialista de II grado en Oftalmología, Profesor Auxiliar, Investigador Auxiliar.

Estos se reunieron y marcaron en consenso en las doce imágenes las regiones que consideraron opacidad. Estas imágenes (*Ground Truth*) fueron usadas en la validación de la segmentación por los algoritmos propuestos como se muestra en la Figura 26.

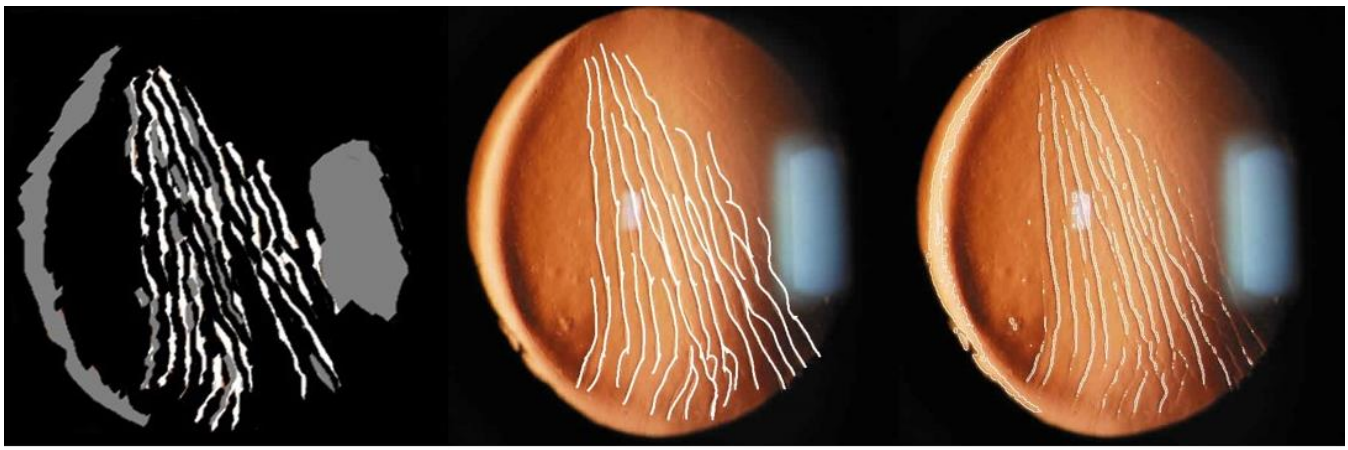


Figura 26.Imagen segmentada por los algoritmos propuestos y por el especialista.

3.5.1 Métricas de validación de la segmentación

En el campo de la ingeniería del software una métrica es cualquier medida o conjunto de medidas destinadas a conocer o estimar el tamaño u otra característica de un software o un sistema de información, generalmente para realizar comparativas o para la planificación de proyectos de desarrollo.

A continuación, se analizan los resultados (Mínimo Valor, Máximo Valor, Media y Desviación Típica) de las métricas (Tasa de Correctos Equilibrada, Índice de *Sokal-Sneath,* y Modificación de la Distancia de Hausdorff Normalizada) obtenidas a partir de la utilización del software PASI. Se analizaron las imágenes obtenidas luego de la segmentación por texturas mediante los algoritmos que utilizan matriz de coocurrencia y campo aleatorio de Markov, en comparación con las imágenes segmentadas por los especialistas.

Tasa de Correctos Equilibrada esta tomará valor 1 cuando lo segmentado por el algoritmo coincida completamente con la OCP marcada por los especialistas. (49) En la Figura 27 se pueden apreciar los resultados de esta medida sobre los algoritmos propuestos. El algoritmo que utiliza Matriz de Coocurrencia tiene una marcada superioridad en esta medida marcando una media de 0.59 y desviación 0.12 a diferencia del algoritmo que utiliza Campos de Markov que solo alcanzan un valor medio de 0.255.

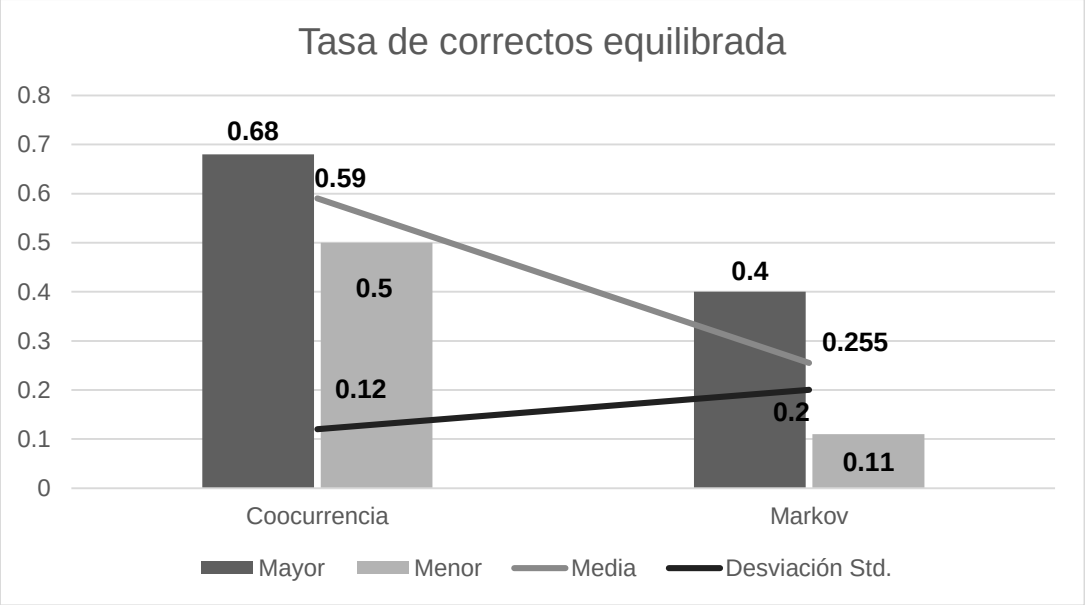


Figura 27. Resultados de la métrica Tasa de correctos equilibrada

Índice de Sokal-Sneath es una de las medidas estadística del error tomada en cuenta*. Esta* penaliza las regiones que fueron mal identificadas. Por lo que cuando toma valor 1, significa que todas las regiones con opacidad fueron correctamente segmentadas. (50) En la Figura 28 se pueden apreciar los resultados de esta medida sobre los algoritmos propuestos. En el caso del algoritmo que utiliza Matriz de Coocurrencia tiene una marcada superioridad en esta medida marcando una media de 0.39 a diferencia del algoritmo que utiliza Campos de Markov que solo alcanzan un valor medio de 0.32.

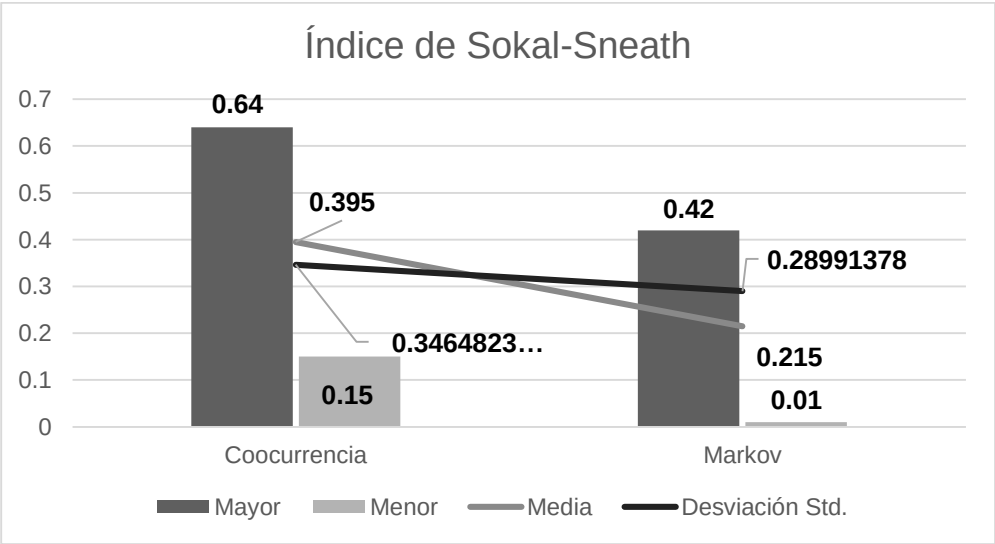


Figura 28. Resultados de la métrica Índice de Sokal-Sneath

Modificación de la Distancia de Hausdorff Normalizada esta métrica devuelve valores de distancia por las diferencias existentes entre la segmentación de los algoritmos y el *Ground Ttrusth* significando esto que el valor sea 0 en caso de una segmentación perfecta, y mientras mayor sea el valor de la misma, mayor la cantidad de diferencias encontradas. (51)

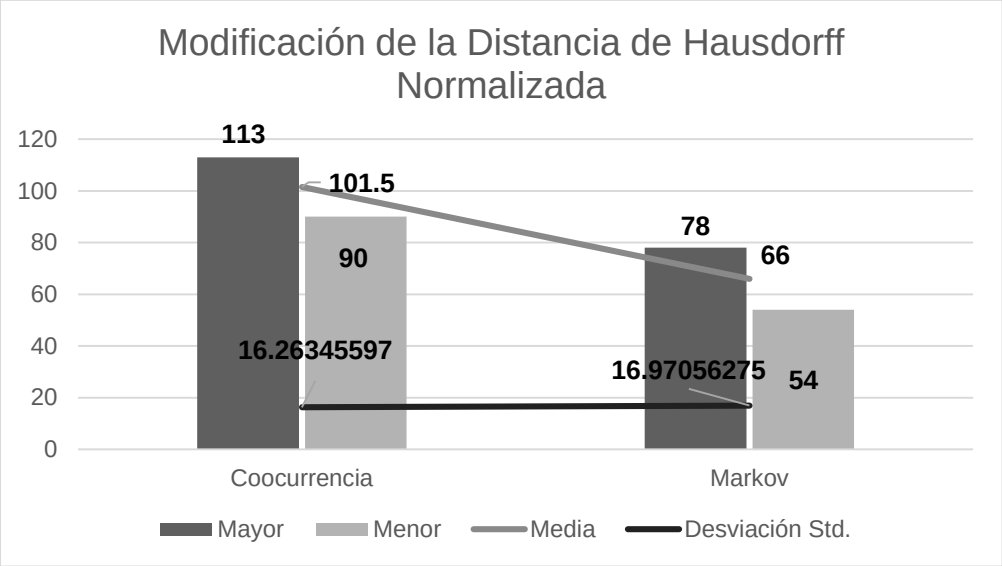


Figura 29. Resultados de la métrica Modificación de Distancia de Hausdorff

3.5.2 Análisis estadísticos.

Las pruebas estadísticas permiten contrastar la veracidad o falsedad de las hipótesis enunciadas desde el punto de vista estadístico. Las cuales se clasifican en dos grandes grupos de pruebas de significación estadística, el referido a las paramétricas y el relacionado con las no paramétricas, con rasgos distintivos que las caracterizan.

Con el objetivo de comparar el desempeño de los algoritmos se procede a realizar un análisis estadístico. Para esto se selecciona el uso de pruebas no paramétricas debido al tamaño de la muestra (solo 12 imágenes). Se escoge usar el análisis de dos vías de Friedman y como prueba post-hoc la prueba de rango de Wilcoxon por pares, ajustándose el nivel de significancia con una corrección de Bonferroni. Esto se aplicó sobre la variable Sokal-Sneath como medida estadística del error cometido en la segmentación, sobre la variable Tasa de Correctos Equilibrada como medida de localización y sobre Modificación de la Distancia de Hausdorff Normalizada como una variente de las métricas de Hausdorff. Para realizar todos los cálculos se empleó el software IBM SPSS Statistics 21 (52)(53).

Test de suma de rangos de Wilcoxon

Es un análisis estadístico de elementos de tamaño n1 de una población, y un segundo grupo de elementos de tamaño n2 de otra población. Hay n observaciones en total, donde n = n1 + n2. Se calcula el rango de las n observaciones. (54) El test estadístico será la suma W de los rangos del grupo con menor suma de rangos, este será el estadístico de suma de rangos de Wilcoxon. Si las dos poblaciones tienen la misma distribución continua, entonces W tiene media:

Ecuación 12. Ecuación de cálculo de la suma de rango de Wilcoxon para dos poblaciones.

$$\mu_w = \frac{(n_1(n+1))}{2}$$

Y desviación estándar:

Ecuación 13. Ecuación de cálculo de la desviación estándar para la suma de rango de Wilcoxon para dos poblaciones.

$$\sigma_w = \sqrt{\frac{n_1 n_2 (n+1)}{12}}$$

Donde n_1 será el tamaño muestral del grupo con menor suma de rangos.

El test de suma de rangos de Wilcoxon rechaza la hipótesis nula de que las dos poblaciones tienen la misma distribución cuando la suma de rangos W está lejos de su media.

Hipótesis para una prueba de signo o una prueba de rango con signo de Wilcoxon.

Las hipótesis que se prueban con la prueba de signo y la prueba de rango con signo de Wilcoxon son las siguientes:

- H_0: La probabilidad de una diferencia positiva es igual a la probabilidad de una diferencia negativa.
- H_a: La probabilidad de una diferencia positiva no es igual a la probabilidad de una diferencia negativa.

Como regla de decisión se rechaza la hipótesis nula cuando exista una significancia menor de 0.05 (p) < 0.05.

Test de Friedman

La prueba de Friedman es una alternativa no paramétrica a un análisis de varianza de medidas repetidas utilizado para comparar observaciones repetidas sobre los mismos sujetos. A diferencia de las medidas paramétricas ANOVA (52)(53). Además, si la investigación tiene k variables en columnas y n elementos en filas se trata de ordenar cada fila de menor a mayor según las diferentes columnas de 1 hasta k (esto es el rango que ocupa cada variable para ese caso). Si no hay diferencias estadísticamente significativas entre las variables se espera que los rangos estén repartidos en cada columna de manera uniforme y sólo se encontrarán entre las variables pequeñas diferencias debidas al azar.

Hipótesis para la prueba de Friedman

Las hipótesis que se prueban en la prueba de Friedman son las siguientes:

- H_0: Las distribuciones son las mismas a través de medidas repetidas.
- H_a: Las distribuciones a través de medidas repetidas son diferentes.

Como regla de decisión se rechaza la hipótesis nula cuando exista una significancia menor de 0.05.

Como se muestra en la Tabla 17 el valor de precisión (p) en todos los casos es menor que 0.05, presentando un nivel de significancia del 95% por lo que se rechaza la hipótesis nula y se puede confirmar que existen diferencias significativas entre los grupos de algoritmos analizados. Es decir, los resultados de los algoritmos en las variables analizadas cambiaron significativamente.

Tabla 17. Test de Friedman sobre los resultados de las variables Tasa de correctos equilibrados, Modificación de la Distancia de Hausdorff Normalizada y Sokal – Sneath.

No	Hipótesis nula	Test	Sig.	Decisión	Variables
1	Las distribuciones Coocurrencia y Markov son la misma.	Análisis de dos vías de Friedman de varianza por rangos de muestras relacionadas.	.003	Rechazar la hipótesis nula.	Tasa de correctos equilibrado
2	Las distribuciones Coocurrencia y Markov son la misma.		.005	Rechazar la hipótesis nula.	Modificación de la Distancia de Hausdorff Normalizada
3	Las distribuciones Coocurrencia y Markov son la misma.		.004	Rechazar la hipótesis nula.	Sokal-Sneath

Se desarrolló una prueba post-hoc de rangos con signos de Wilcoxon con el objetivo de comparar dos a dos las muestras relacionadas (algoritmos de segmentación) y determinar entre qué resultados se presentaron las diferencias significativas. Se usó la corrección de Bonferroni para evitar cometer un error de Tipo I (rechazar la hipótesis nula cuando debería aceptarse). El nuevo nivel de significación que se tomó por tanto fue de 0,016 (0.05/3=0.0166).

Tabla 18. Prueba de Wilcoxon sobre los resultados de las variables Tasa de correctos equilibrados, Modificación de la Distancia de Hausdorff Normalizada y Sokal – Sneath.

No	Hipótesis nula	Test	Sig.	Decisión	Variables
1	La mediana de las diferencias entre Coocurrencia y Markov es igual a 0.	Prueba de Wilcoxon de los rangos con signo de muestras relacionales.	.007	Rechazar la hipótesis nula.	Tasa de correctos equilibrado
	La mediana de las diferencias entre Coocurrencia y Markov es igual a 0.		.010	Rechazar la hipótesis nula.	Modificación de la Distancia de Hausdorff Normalizada
3	La mediana de las diferencias entre Coocurrencia y Markov es igual a 0.		.008	Rechazar la hipótesis nula.	Sokal-Sneath

Cuando se analizan los resultados luego de aplicar el test de Wilcoxon (ver tabla 18), se puede observar que en todos los casos se obtiene una significancia menor que 0.016, por esto se rechaza la hipótesis nula y se puede afirmar con un alto nivel de certeza que existen diferencias significativas entre los algoritmos. Dado este resultado se toma la media como medida para evaluar el rendimiento del algoritmo en las métricas aplicadas. Por tanto, se asume que el algoritmo de segmentación de OCP que mejores resultados brinda dado los valores de las medias es Matriz de Coocurrencia.

Conclusiones parciales

- Las pruebas de aceptación enfocadas a los usuarios se realizaron en cada ciclo de iteración, esto permitió un correcto desarrollo de la herramienta y un alto nivel de satisfacción de los usuarios con la misma.

- Las pruebas unitarias permitieron validar el correcto funcionamiento del algoritmo desarrollado, donde el código analizado ofreció resultados favorables en la mayor parte de las ocasiones.
- Los especialistas realizaron la segmentación manual de las imágenes en retroiluminación que presentaban opacidad lo que permitió realizar una correcta validación del algoritmo propuesto.
- La utilización de la validación estadística y las variables: Tasa de correctos equilibrados, Modificación de la Distancia de Hausdorff Normalizada y Sokal – Sneath, permitió comprobar que la Matriz de coocurrencia en la segmentación de regiones con opacidad ofrece mejores resultados que el algoritmo de campo aleatorio de Markov

Conclusiones

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

- La elaboración del marco teórico de la investigación, así como los principales problemas identificados en la bibliografía, permitieron seleccionar las técnicas del procesamiento digital de imágenes más adecuadas para mejorar la calidad de las imágenes en retroiluminación y seleccionar los mejores procedimientos y técnicas para la segmentación por texturas.
- La matriz de coocurrencia en conjunto con el algoritmo K-Means constituyen una estrategia efectiva para la segmentación por texturas en imágenes provenientes de la lámpara de hendidura, esta técnica permite obtener y agrupar las variables de texturas fundamentales para la detección de OCP.
- La validación de la herramienta mediante pruebas unitarias y de aceptación comprobó que el algoritmo desarrollado permite la identificación de las distintas regiones con opacidad dentro de las imágenes en retroiluminación, ayudando en el diagnóstico de los pacientes operados de catarata. La validación estadística realizada a la herramienta permitió aceptar la hipótesis de la investigación al comprobarse que, con la utilización de la matriz de coocurrencia para la detección de opacidad de la cápsula posterior en imágenes de retroiluminación, se obtienen mejores resultados que al utilizar algoritmos basados en campo aleatorio de Markov.

Recomendaciones

- Agregar un componente visual que permita establecer distintos niveles de sensibilidad para mejorar la segmentación por texturas en imágenes de menor calidad.
- Agregar la funcionalidad que permita seleccionar una zona de interés donde se desee realizar la segmentación por texturas y obtener resultados más específicos en un menor tiempo.
- Incorpora al algoritmo la utilización de más variables de texturas.

Bibliografía

1. MIQUELI, Maritza Rodríguez, LÓPEZ, Silvia M Hernández and MASÓ, Susana Rodríguez. Baja visión y envejecimiento de la población. *Rev Cubana Oftalmol* [online]. 2016. P. 492–501. Available from: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21762016000300011&lng=es

2. MARGARITA, Calonge. *Causas de Ceguera en el Mundo : distribución geográfica y relación con el medio socio-económico*. Valladolid, Facultad de Ciencias, 2014.

3. Ceguera y discapacidad visual. *Organización Mundial de la Salud* [online]. 2014. Available from: http://www.who.int/mediacentre/factsheets/fs282/es/

4. KIERSTAN, BOYD. ¿Qué Son las Cataratas? -. *American Academy of Ophthalmology* [online]. 2017. Available from: www.aao.org/eye-health/diseases/what-are-cataracts

5. CHRIS A, Knobbe. Complicaciones de la cirugía de cataratas. *All About Vision and AllAboutVision.com are registered trademarks of AAV Media, LLC. © 2000-2018 AAV Media, LLC.* [online]. 2016. Available from: http://www.allaboutvision.com/es/condiciones/complicaciones-cirugia-de-cataratas.htm

6. HERNÁNDEZ, Iván, HERNÁNDEZ, Juan Raúl and CASTRO, Yadira. Estrategias de prevención de la opacidad de la cápsula posterior. *Revista Cubana de Oftalmología.* 2010. Vol. 23, p. 608–623.

7. CANCIO, Michel Alvarez. PANDOC:Sistemas basados en casos para la cuantificación objetiva de la opacidad capsular de un paciente operado de catarata. *2014*.

8. ÁLVAREZ, Michel et al. *Aplicación de un sistema basado en casos para la identificación de opacidad en la cápsula posterior mediante imágenes del pentacam.* [online]. 2016. Available from: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-18592016000300007

9. SA, BALBERDI, TXOMIN, et al. Anterior capsule opacification after femtosecond laser–assisted cataract surgery: Clinical classification versus Scheimpflug device densitometry values. *Journal of Cataract & Refractive Surgery*. 2016. Vol. 42, no. 6, p. 826–832.

10. OLIVER FINDL, W. B. Intervenciones para la prevención de la opacificación de la cápsula posterior. . 2010.

11. ASLAN, Tariq M, PATTON, Niall and BALJEAN, Dhillon. Assessment of systems of analyzing PCO. Journal of Cataract and Refractive Surgery. *2005*. Vol. 31.

12. D. S FRIEDMAN, D. D. . Digital image capture and automated analysis of posterior capsular opacification. Investigative Ophthalmology & Visual Science. . 1999.

13. GARCÍA, YainetGarcía and RODRÍGUEZ, Reinier Guillén. Mapeo de imágenes digitales de fondo de ojo atendiendo a rasgos de textura. *Revista Cubana de Oftalmología.* 2017. Vol. 11, no. 1.

14. CANCIO, Michel Alvarez, HERNÁNDEZ, Adrián Barrios, RODRÍGUEZ, Rafael Puentes and HERNANDEZ, Iván López. PANDOC: software para la cuantificación objetiva de la opacidad de cápsula posterior mediante tomogramas scheimpflug del pentacam. . 2013.

15. PITA, Demetrio Salorio. *Diccionario terminológico de oftamología*. 2014.

16. CASTRO CAMACHO, KATIUSCA, Jennifer, PEÑA, Cerquera, ENRIQUE, Néstor and GUTIERREZ GUZMAN, Nelson. Determinación del color del exocarpio como indicador de desarrollo fisiológico y madurez en la guayaba pera (Psidium guajava cv. Guayaba pera), utilizando técnicas de procesamiento digital de imágenes. *Revista EIA*. 2013. Vol. 19, p. 68–79.

17. RODRÍGUEZ, José Luís Gil. Estado Actual de la Representación y Análisis de Textura en Imágenes. *Serie Azul*. 2008. No. 21812.

18. RAFAEL, Gonzales and RICHARD, Woods. Digital image processing. . 2002.

19. V. SRINIVASAN, R.D. Instrumentos y Equipos Oftalmológicos. . 2003.

20. RODRÍGUEZ, Odette Romero and LILIANA, Rodríguez Martínez. Algoritmo para la segmentación de la opacidad de la cápsula posterior en imágenes oblicuas provenientes de la lámpara de hendidura. . 2016.

21. OLIVA, Karla Rodriguez. Suavizado mediante técnicas de filtradomorfológico en imágenes de microscopía celular. . 2014.

22. CEPEDA NEGRETE, Jonathan and SANCHEZ YANEZ, Raul E. Algoritmos de Constancia de Color para el Mejoramiento de Im·genes Oscuras. V Congreso Internacional de la Ciencia de Sistemas. . 2010.

23. ÇESMELI, ERDOGAN;WANG, DeLiang. Texture Segmentation Using Gaussian–Markov Random Fields and Neural Oscillator Networks. *IEEE TRANSACTIONS ON NEURAL NETWORKS,*. 2001.

Vol. 12, no. 2, p. 394–404.

24. OYARZO, Weslly Ogier Jara. Visión Artificial: Análisis Teórico del Tratamiento Digital de Imágenes Para su aplicación en la identificación de objetos. . 2006.

25. ÁLVAREZ, YAIMARA MÁRQUEZ and REYES., Licel Salazar. Propuesta de un filtro para la eliminación de ruido causado por la iluminación en el sistema de visión de un robot móvil. . 2010.

26. VALENZUELA, Javier Vidal. Interpolación de Formas en Imágenes Usando Morfologıa Matemática. . 2014.

27. LA SERNA, Dra.Nora Palomino and CONCHA ROMÁN, LIc.Ulises. Técnicas de Segmentación en Procesamiento Digital de Imágenes. *Facultad de Ingeniería de Sistemas e Informática Universidad de San Marcos*. 2015. P. 9–16.

28. LUGO, Jorge, SAMPALLO and THOMAS, Guillermo M Gonzáles. Detección de Zonas de Diferentes Texturas Usando Análisis de Multiresolución. *Depto. de Física Fa.C.E.N.A. - U.N.N.E - Corrientes - Argentina* [online]. 2015. No. February. Available from: https://www.researchgate.net/publication/267413851_Deteccion_de_Zonas_de_Diferentes_Texturas_Usando_Analisis_de_Multiresolucion

29. LI,JIANGHONG; CHEN, LIANG ;CAI, Yuanhu. Dynamic Texture Segmentation Using Fourier Transform. *Modern Applied Science* [online]. 2009. Vol. vol.3,no 9. Available from: http://dx.doi.org/10.5539/mas.v3n9p29

30. AL-KADI, Omar S. Supervised Texture Segmentation : A Comparative Study. *King Abdullah II School for IT University of Jordan, Amman*. 2014.

31. VAIJINATH, V. BHOSLE; VRUSHSEN, P. Pawar. Texture Segmentation : Different Methods. *International Journal of soft Computing and Engineering*. 2013. Vol. 3, no. 5, p. 69–74.

32. JAHANE, B. Digital Image Processing. . 2010.

33. NICOT, Miriela Milagros Escobedo, GARCÍA, Silena Herold and ÁLVAREZ, Frank Calzadilla. Determinación De Zonas De Interés Con El Empleo De Texturas. Ciencia en su PC. . 2015.

34. PINTO, ALEJANDRA CAROLINA, Leal. *Segmentación de imágenes por textura*. UNIVERSIDAD DE CONCEPCIÓN, 2006.

35. LÓPEZ-ESPINOZA, Erika Danaé and ROBLES, Leopoldo Altamirano. Segmentación Markoviana

Usando Modelos de Textura. . 2014.

36. SOTO, A. Textures classification using markov random shields. . 2013.

37. ANDRÉS TRIBUJ, Mariano, DAVID WAISBAUM, Arie, DRA. MEJAIL, Marta and DR. JACOBO BERLLÉS, Julio. Segmentación de imágenes texturadas. . 2009.

38. ORTIZ, A. Texturas y descripciones de regiones. . 2013.

39. PRESUTTI, Miriam. LA MATRIZ DE CO-OCURRENCIA EN LA CLASIFICACIÓN MULTIESPECTRAL: TUTORIAL PARA LA ENSEÑANZA DE MEDIDAS TEXTURALES EN CURSOS DE GRADO UNIVERSITARIO. *Universidad Nacional de La Plata Facultad de Ciencias Agrarias y Forestales Departamento de Ambiente y Recursos Naturales*. 2004.

40. G. KARYPIS, E. H. Han. And V. Kumar. Chameleon: A hierarchical clustering algorithmusing dynamic modeling",. *IEEE Computer,*. 1999. Vol. 32(8), p. 68–75.

41. DUDA, R. O., HART, P. E. and STORK, D. Pattern Classification. *Wiley series in Probabilistic and Statistic*. 2001.

42. M. ESTER, H.P. KRIEGEL, J. Sander and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise". *", In Proceedings of International Conference on Knoowledge Discoveryand Data Mining (KDD´96*. 1996. P. 226–231.

43. SANCHEZ, M.A.M. Metodologías de Desarrollo de Software. . 2012.

44. LETELIER, PATRICIO Y Mª CARMEN PENADÉS, José H. Canó. Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP). . 2017.

45. MATHWORKS. Matlab. [online]. 2018. Available from: https://es.mathworks.com/products/matlab.html

46. PARDO, A. Y RUIZ, M. A. Guía para el análisis de datos . Madrid: McGraw-Hil. . 2002.

47. WAKE, William C. *Extreme programming explored*. 2017.

48. ORDOÑEZ, H., et al. *Business Processes as a Strategy to Improve Re-quirements Elicitation in Extreme Programming (XP).* 2015.

49. KULIS, BRIAN, et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*. 2013. Vol. 5, p. 287–364.

50. BOUZA-HERRERA, Carlos N. UNA MIRADA A LOS MÉTODOS Y ALGORITMOS DEL ANÁLISIS DE CLÚSTERES. . 2013.

51. VICENTE CESTERO, ELOY MATEOS CABALLERO, Alfonso and JIMÉNEZ MARTÍN, Antonio. Elicitación de probabilidades difusas. *2014*.

52. GREEN, Samuel B. and SALKIND, Neil J. Using SPSS for Windows and Macintosh, Books a la Carte. *Pearson*. 2016. Vol. 2, p. 123–234.

53. ALAN, W. A. W. and ELLIOTT, C. Statistical analysis quick reference guidebook: With SPSS examples. *Sage*. 2013. P. 201 –203.

54. PÉRTEGA DÍAZ, SONIA; PITA FERNÁNDEZ, S. Métodos no paramétricos para la comparación de dos muestras. *Cad Aten Primaria*. 2006. Vol. 13, p. 109–113.

Anexos

Historia de usuario 1

Historia de usuario	
Número:1	**Nombre:** Cargar Imagen
Usuario: Especialista	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados[1]: 0,5	**Iteración asignada:** 1
Programador responsable: Jessie Romero Pérez	
Descripción: Cargar la imagen en formato jpg,png o bmp, donde el usuario especifica donde se encuentra la imagen	
Observaciones:	

Historia de usuario 2

Historia de usuario	
Número:2	**Nombre:** Mejorar contraste de la imagen
Usuario: Especialista	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 2	**Iteración asignada:** 1
Programador responsable: Jessie Romero Pérez	
Descripción: Las imágenes resultantes de la lámpara de hendidura presentan diferentes sombras debido a que hay parte de la misma que no siempre se pueden iluminar con la luz proveniente de artefacto, por lo que se hace necesario obtener una distribución uniforme de los diferentes contrastes de la imagen para obtener identificar con mayor claridad la región.	
Observaciones:	

Tabla 19 Historia de usuario 3

Historia de usuario	
Número:3	**Nombre:** Aplicar filtro lineales
Usuario: Especialista.	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 2,5	**Iteración asignada:** 1
Programador responsable: Jessie Romero Pérez	
Descripción: Pueden ser seleccionados los filtros gaussiano y la medida, los cuales permiten eliminar el ruido existente en la imagen en el caso de ser seleccionado, en el caso de ser seleccionado la media se obtiene una difuminación de la imagen, produciendo un suavizado en los bordes, mientras que al aplicar el gaussiano se obtiene un efecto de desenfoque en la imagen.	
Observaciones:	

Tabla 20. Historia de usuario 4

Historia de usuario	
Número:4	**Nombre:** Obtención de la matriz de los niveles de grises
Usuario: Especialista	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 1	**Iteración asignada:** 1
Programador responsable: Jessie Romero Pérez	
Descripción: Se subdivide en matrices 21 x 21 píxeles calculando las relaciones entre pixel en las 4 direcciones N, NE, E y SE obteniendo una matriz con la cantidad de veces que se repite las posibles combinaciones de los niveles de grises.	
Observaciones:	

Tabla 21. Historia de usuario 5

Historia de usuario

Número:5	**Nombre:** Obtención de matriz de probabilidad
Usuario: Especialista	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 2	**Iteración asignada:** 1
Programador responsable: Jessie Romero Pérez	
Descripción: Esta matriz muestra las posibles combinaciones de los niveles de grises, es convertida en una matriz simétrica luego, y finalmente se expresa como matriz de probabilidad	
Observaciones:	

Tabla 22. Historia de usuario 6

Historia de usuario	
Número:6	**Nombre:** Obtención de las variables de textura
Usuario: Especialista	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 2	**Iteración asignada:** 1
Programador responsable: Jessie Romero Pérez	
Descripción: Se normaliza la matriz de coocurrencia de nivel de gris (GLCM) de modo que la suma de sus elementos sea igual a 1. Cada elemento (r, c) en el GLCM normalizado es la ocurrencia conjunta de probabilidad de pares de píxeles con una relación espacial definida que tiene valores de nivel de gris r y c en la imagen. Luego se calcula las estadísticas especificadas en las propiedades de la matriz de coocurrencia de nivel de grises normalizada	
Observaciones:	

Tabla 23 Historia de usuario 7

Historia de usuario	
Número:7	**Nombre:** Cálculo de distancia y clúster
Usuario: Especialista	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados:2	**Iteración asignada:** 1
Programador responsable: Jessie Romero Pérez	

Descripción: A partir de las variables de textura obtenidas se calcula la distancia euclidiana de los píxeles al clúster más cercano. Seguida una etapa de recalculo de clúster y las distancia de los pixeles a los mismo.
Observaciones:

Tabla 24 Historia de usuario 8

Historia de usuario	
Número:8	**Nombre:** Agrupamiento
Usuario: Especialista	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados:2	**Iteración asignada:** 1
Programador responsable: Jessie Romero Pérez	
Descripción: Los clústeres y las distancias obtenida en la primera fase son recalculados para obtener mejor convergencia en el agrupamiento de cada pixel a su clase k-means	
Observaciones:	

Tabla 25. Tarea de desarrollo #2

Tarea	
Número de tarea:2	**Numero de Historia de usuarios :** 2
Nombre de la tarea: Desarrollar la funcionalidad para aplicar filtro lineal	
Tipo de tarea: configuración-desarrollo	**Puntos estimados:** 2,5
Fecha de Inicio:	**Fecha de fin:**
Programador responsable : Jessie Romero Pérez	
Descripción: Para implementar el filtro lineal gaussiano se utiliza la función ***fspecial*** con los parámetros tipo de filtro *gaussian, kernel* [15 15] e intensidad 0.8.Mientras para implementar el filtro i***msharpen*** se definen los parámetro imagen, fortaleza del efecto de realce 5, desviación estándar del filtro 5.	

Tabla 26. Tarea de desarrollo #3

Tarea	
Número de tarea: 3	**Numero de Historia de usuarios:** 3

Nombre de la tarea: Desarrollar funcionalidad para mejorar el contraste de la imagen	
Tipo de tarea: configuración-desarrollo	**Puntos estimados:** 2,5
Fecha de Inicio:	**Fecha de fin:**
Programador responsable: Jessie Romero Pérez	
Descripción: Para mejorar el contraste de las imágenes se utilizan las funciones adaphisteq y stretchlim las cuales mejoran el contraste de la imagen utilizando una ecualización adaptativa con contraste limitado y amplia los límites de intensidad de la imagen respectivamente. Las funciones trabajan con imágenes en escala de grises y puede ser del tipo uint8, uint16, int16, single o double que son los tipos de datos que contendrá una imagen en Matlab.	

Tabla 27. Tarea de desarrollo 5

Tarea	
Número de tarea: 5	**Numero de Historia de usuarios :** 5
Nombre de la tarea: Desarrollar funcionalidad para la obtención de la matriz simetrica de los niveles de grises	
Tipo de tarea: configuración-desarrollo	**Puntos estimados:** 2,5
Fecha de Inicio:	**Fecha de fin:**
Programador responsable : Jessie Romero Pérez	
Descripción: Se subdivide en matrices nxn calculando la relaciones entre pixel en las 4 direcciones N,NE,E y SE obteniendo una matriz con la cantidad de veces que se repite las posible combinaciones de los niveles de grises. Para definir las direcciones se crea la matriz donde cada para de valore indica la dirección del pixel vecino con relación al pixel central offsets = [0 1; -1 1; -1 0; -1 -1]; y para dividir la imagen en submatrices se utilizó la función imcrop.	

Tabla 28. Tarea de desarrollo 6

Tarea	
Número de tarea:6	**Numero de Historia de usuarios :6**
Nombre de la tarea: Desarrollar funcionalidad para Obtención de matriz de probabilidad	
Tipo de tarea: configuración-desarrollo	**Puntos estimados:** 3
Fecha de Inicio:	**Fecha de fin:**
Programador responsable : Jessie Romero Pérez	

Descripción: Esta matriz a partir de las posibles combinaciones de los niveles de grises, es convertida en una matriz simétrica luego se expresa como matriz de probabilidad. Para la obtención de la matriz de probabilidad se utiliza la función ***graycomatrix*** donde por parámetro recibe las direcciones y la matriz a analizar.

Tabla 29. Tarea de desarrollo 7

Tarea	
Número de tarea: 7	**Numero de Historia de usuarios :**
Nombre de la tarea: Desarrollar funcionalidad obtención de las variables de textura	
Tipo de tarea: configuración-desarrollo	**Puntos estimados:**
Fecha de Inicio:	**Fecha de fin:**
Programador responsable : Jessie Romero Pérez	
Descripción: A partir de la matriz de probabilidades se obtienen un grupo de variables de texturas necesarias para realizar el agrupamiento y posteriormente la segmentación. Para la obtención de las variables de textura se utilizó la función **graycoprops**,	

Tabla 30. Caso de uso de prueba de aceptación #1

Caso de prueba de aceptación
Código: HU_P1 | Historia de Usuario: 1
Nombre: Cargar Imagen.
Descripción: Comprueba la funcionalidad que permite explorar los archivos y verificar el formato de las mismas.
Condiciones de Ejecución: Que exista una imagen con las características y el formato definido.
Entrada/Pasos de Ejecución:
• El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario carga la imagen en el formato seleccionado. • El sistema muestra la imagen en pantalla
Resultado Esperado:
Evaluación de la Prueba: Prueba satisfactoria**.**

Tabla 31. Caso de uso de prueba de aceptación #3

Caso de prueba de aceptación
Código: HU_P1 | Historia de Usuario: 3
Nombre: Mejorar Contraste de la imagen.
Descripción: Comprueba la funcionalidad que mejora el contraste de la imagen
Condiciones de Ejecución: Se debe haber cargado una imagen resultante de la lámpara de hendidura
Entrada/Pasos de Ejecución:
• El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción mejorara el contraste. • El **sistema m**uestra la imagen en escala de grises, logrando mejorara el contraste
Resultado Esperado:
Evaluación de la Prueba: Prueba satisfactoria**.**