



Universidad de las Ciencias Informáticas
AIRI, Artificial Intelligence Research and Innovation, Facultad 3

Análisis de los efectos de la interpolación en la segmentación de imágenes de opacidad capsular

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

Autores:

Ismael Montano Muñoz

Tutores

Ing. Michel Álvarez Cancio

La Habana, del 2018



La creatividad simplemente consiste en conectar cosas.

Steve Jobs

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Ismael Montano Muñoz

Firma del autor

Ing. Michel Álvarez Cancio

Firma del tutor

AGRADECIMIENTOS

Quiero agradecer primeramente a mi madre Belkis y a mi padre Richard que me han apoyado desde el comienzo y me han dado fuerza para cumplir mis sueños, de cierto modo esta tesis es fruto de sus esfuerzos y sacrificios y me atrevo a decir que mis logros también son de ellos. Agradezco también a mis abuelos a los cuales nunca les ha faltado un consejo para mí; gracias a su amor, hoy soy ingeniero. A mi novia Aimé, que ha sido en muchas ocasiones ese apoyo que se necesita para luchar, ha convertido mi cansancio en disposición y mis debilidades en fortalezas. A mis tíos y primos, David, Alina, Ivón, Emir, Cinthia, Dayanis, Lester y todos aquellos que no mencioné, gracias a todos por apoyarme siempre. A mis amigos Yudiel y Franci por estar ahí siempre que los necesité. A mis compañeros de aula por brindarme tantos buenos momentos y por ser casi una familia para mí. A mi tutor Michel Álvarez Cancio por la ayuda siempre durante el desarrollo de este trabajo. A todos los profesores que me han impartido clases en mi estancia aquí en la UCI. A nuestra revolución por permitir que la juventud cubana sea protagonista indispensable del crecimiento de nuestra sociedad.

DEDICATORIA

Todo mi trabajo y dedicación se lo dedico a mi familia y amigos, especialmente a mis padres, a mis abuelos, a mis tías y tíos, a mi novia y a todas las personas que de una forma u otra me han apoyado en todo y me han ayudado a cumplir mis metas.

RESUMEN

Actualmente, el campo de la medicina se ve en un avance tecnológico a nivel mundial, es un hecho que gracias al desarrollo del equipamiento médico se ha logrado generar una gran cantidad de imágenes médicas, dichas imágenes se ven en la necesidad de ser procesadas para aumentar y lograr un mejor nivel de detalle. Sin embargo, cuando se segmentan estas imágenes para obtener la región de interés se obtienen resultados que podrían mejorar si se hubiera realizado un proceso previo de interpolación. Precisamente, en este trabajo se investigaron e implementaron distintos métodos de interpolación para llevarlos a la práctica y comprobar que beneficios se obtienen del uso de estos métodos en la calidad de la segmentación sobre dichas imágenes. Para dar solución a este problema se desarrolló una aplicación capaz de segmentar estas imágenes después de haber mejorado su resolución mediante una interpolación. Al sistema se le han realizado un conjunto de pruebas para determinar la eficiencia de los resultados obtenidos con respecto a la calidad de imagen. Además de que ha sido validado en base a un algoritmo de detección de bordes mediante el cual se puede demostrar la mejoría de los resultados.

Palabras claves: Interpolación, segmentación, imágenes, opacidad capsular

CONTENIDO

Índice

Introducción	10
Capítulo 1: Fundamentación Teórica	14
1.1. Introducción	14
1.2. Lámpara de hendidura	15
1.3. Procesamiento digital de imágenes.....	15
1.4. Segmentación	17
1.4.1. Métodos de segmentación	18
1.4.1.1. Umbralización (Segmentación de Otsu)	19
1.4.1.2. Transformada circular de Hough	19
1.4.1.3. Crecimiento de regiones	21
1.5. Interpolación	22
1.5.1. Métodos de Interpolación	23
1.5.1.1. Interpolación bilineal	23
1.5.1.2. Interpolación bicúbica	24
1.5.1.3. Interpolación mediante spline.....	25
1.5.1.4. Interpolación mediante el vecino más cercano.....	26
1.6. Metodología, herramientas y lenguaje de programación utilizado durante el desarrollo.....	27
1.6.1. Metodología utilizada para el desarrollo	27
1.6.2. Lenguaje de Programación	28
1.6.3. Entorno Integrado de Desarrollo	29
1.6.4. Herramienta para el modelado	29
1.7. Conclusiones Parciales	30
Capítulo 2: Análisis y diseño de la propuesta de solución	30
2.1. Introducción	30
2.2. Descripción de la solución.....	31
2.2.1. Requisitos Funcionales	35
2.2.2. Requisitos no funcionales	36

2.2.3. Historias de Usuario.....	36
2.2.4. Estimación de esfuerzo por historias de usuario	38
2.2.5. Plan de Iteraciones	39
2.2.6. Plan de Entrega	40
2.3. Diseño.....	41
2.3.1. Tarjetas CRC.....	41
2.3.2. Estándares de codificación	42
2.3.3. Patrones de Diseño.....	43
2.4. Conclusiones Parciales.....	44
Capítulo 3: Resultados y Validación.	44
3.1. Introducción	44
3.2. Fase de Implementación.....	45
3.3. Interfaz del Sistema	45
3.4. Pruebas	46
3.4.1. Pruebas unitarias	47
3.5.1.1. Pruebas de caja blanca.....	47
3.5.1.2. Pruebas de caja negra	49
3.6. Validación de la solución	52
3.6.1. Validación por detección de bordes.....	52
3.7. Conclusiones Parciales.....	53
Conclusiones generales	54
Referencias.....	55
Anexos.....	60

Índice de Imágenes

Ilustración 1 . Lámpara de Hendidura.....	15
Ilustración 2 . Resultado del incremento de la resolución de una imagen.....	16
Ilustración 3 . Ejemplo de imagen multicanal.....	17
Ilustración 4 . Escala de grises.....	17
Ilustración 5 . Opacidad según especialista y opacidad según métodos de segmentación.....	18

Ilustración 6 . Imagen de círculos sin procesar y aplicación de la transformada circular de Hough a dicha imagen	21
Ilustración 7 . A) Región de una lesión muscular B) Objeto segmentado mediante crecimiento de regiones	22
Ilustración 8 . Ejemplo de interpolación	23
Ilustración 9 . Vecindad en el proceso de interpolación bilineal	24
Ilustración 10 . Vecindad en el proceso de Interpolación bicúbica	25
Ilustración 11 . Vecindad en el proceso de Interpolación por el vecino más cercano	26
Ilustración 12 . Descripción de la solución.....	31
Ilustración 13 . Splines para interpolación	33
Ilustración 14 . Interfaz del software	46
Ilustración 15 . Código fuente del método binarizar().....	47
Ilustración 16 . Grafo de flujo del método binarizar()	48
Ilustración 17 . Área del zoom en Imagen original. Resolución: 573 x 430	50
Ilustración 18 . Zoom de la imagen original	50
Ilustración 19 . Zoom de imagen interpolada. Resolución: 1093 x 643	51
Ilustración 20 . Comparación del algoritmo de detección de bordes	53

Índice de Tablas

Tabla 1 . HU: Cargar Imagen.....	37
Tabla 2 . HU: Aplicar a la Imagen el método de interpolación bilineal	38
Tabla 3 . Estimación de esfuerzos por HU	39
Tabla 4 . Plan de Iteraciones	40
Tabla 5 . Plan de Entrega	40
Tabla 6 . Tarjeta CRC de la clase Escala de Grises.....	41
Tabla 7 . Caso de prueba para el camino básico 4	49
Tabla 8 . Caso de prueba para la HU 3.....	51
Tabla 9 . Valores de comparación de detección de bordes	53
Tabla 10 . HU: Aplicar a la imagen el método de interpolación bicúbico	60
Tabla 11 . HU: Aplicar a la imagen el método de interpolación mediante Splines	60
Tabla 12 . HU: Aplicar a la imagen el método de interpolación mediante el vecino más cercano	61
Tabla 13 . HU: Segmentar imagen mediante el método de Umbralización de Otsu .	62
Tabla 14 . HU: Segmentar la imagen mediante el método de Transformada circular de Hough	62

Tabla 15 . HU: Segmentar imagen mediante el método de Crecimiento de regiones	63
Tabla 16 . HU: Guardar imagen	64

Introducción

Actualmente en el mundo se han descubierto y combatido muchas de las enfermedades o discapacidades visuales a las cuales se enfrenta la raza humana, entre ellas la catarata, la cual puede llegar a provocar la pérdida de visión. La catarata es la opacidad parcial o total del cristalino. La opacidad provoca que la luz se disperse dentro del ojo y no se pueda enfocar en la retina, creando imágenes difusas (1).

Esta enfermedad tiene diversas causas pero se le atribuye mayormente a la edad. Con el tiempo se depositan partículas de un color café-amarillo que poco a poco van opacando el cristalino y generan problemas para apreciar los colores y cambios de contraste. Cuando vemos algo, los rayos de luz viajan a nuestro ojo a través de la pupila y se enfocan sobre la retina por medio del lente. El lente debe ser transparente con el fin de poder enfocar la luz adecuadamente sobre la retina (2).

Existen dos tipos de cataratas, la catarata congénita que es producida por la existencia de una lesión hereditaria, y la catarata adquirida que es el tipo más frecuente y es la principal causa de pérdida de visión entre los mayores de 55 años, está causada por la acumulación de células muertas en el cristalino. La transmisión de luz disminuye con la edad, sobre todo para las longitudes de onda más cortas. Esto se debe a que las células del cristalino pierden la organización de su citoesqueleto y se produce una modificación progresiva de las proteínas del cristalino, que genera un incremento de los niveles de sodio en el interior del cristalino y una disminución del transporte de fluidos. Todo ello influye en la pérdida de transparencia del cristalino. (3)

En la actualidad, la mayoría de las veces que se realiza un diagnóstico o cirugía de catarata, los médicos suelen apoyarse en resultados de pruebas de microscopia óptica digital. Después de una intervención quirúrgica el paciente puede recuperar su visibilidad total o parcialmente, pero no en todos los casos la cirugía es un éxito, ya que el paciente puede presentar complicaciones postoperatorias. Entre estas complicaciones la más frecuente a largo o mediano plazo es la Opacificación de la Capsula Posterior (OCP). (4)

La OCP es una condición que se presenta varios meses o años después de la operación de las cataratas; la visión se va deteriorando y vuelven a incurrir los problemas con los reflejos y brillos en la visión (5). Ocurre usualmente debido a que comienza a blanquear la parte posterior de la cápsula que rodea al lente intraocular. La regeneración de las células restantes del cristalino podrían ser la causa, y mientras

más joven sea el paciente, mayor es la probabilidad de que ocurra (4).

La opacificación de la cápsula posterior es actualmente el aspecto más importante en la cirugía de catarata de los tiempos modernos (6). Su incidencia en la actualidad se encuentra entre 0,7-47,6 % en los primeros cinco años de la cirugía, según un análisis de 90 estudios publicados. Que el rango de variación reportado sea tan amplio probablemente responda a diferentes criterios de selección de la muestra, edad, técnica quirúrgica, tipo de lente intraocular (LIO) y sistemas de evaluación de la opacidad (7).

En la actualidad existen muchos programas para la cuantificación de la opacidad en la lente intraocular, entre estos está POCO (Opacificación de cápsula posterior), EPCO (Evaluación de la opacidad de la Cápsula Posterior) (8), AQUA (Cuantificación Automática de Catarata secundaria) (9) y AA (Análisis Aslam) (10). Dichos sistemas trabajan con imágenes digitales obtenidas de las lámparas de hendiduras o generadas por ellos mismos (11). La medicina apoyándose en su tecnología utiliza la imagen digital como un medio importante de información ya que existe la posibilidad de modificar su contenido para lograr un objetivo deseado.

En la Universidad de las Ciencias Informáticas se desarrolló el software PANDOC con el objetivo de identificar y cuantificar de forma automática la OCP. Este sistema provee al oftalmólogo de una herramienta que es capaz de cuantificar y detectar diferencias de opacidad (a veces imperceptibles para el ojo humano), logrando así una evaluación objetiva del grado de opacidad. Una de las imágenes con las que trabaja el software PANDOC, son las imágenes en retroiluminación provenientes de la lámpara de hendidura. Para ellas se han implementado varios métodos para la segmentación de la opacidad de la cápsula posterior, pero no se ha tenido en cuenta la utilización de un proceso previo de interpolación que además de mostrar eficiencia en cuanto a tiempo de procesamiento también eleve la resolución de la imagen con el fin de mejorar los resultados.

Por tal motivo el presente trabajo trata acerca de utilizar algoritmos de interpolación que tomarán lugar antes de la segmentación, para de esta manera verificar que tan buenos son los resultados obtenidos en cuanto a detalle y a dimensión de imagen con respecto a los resultados que se obtienen de una segmentación sin un proceso previo de interpolación. De esta forma la calidad de la imagen de OCP del paciente obtenida de la lámpara de hendidura juega un papel fundamental. La ausencia de detalles, niveles altos de ruido o desenfoque, dificultan al especialista poder realizar un

diagnóstico adecuado.

Dada la situación problemática planteada anteriormente se identifica el siguiente problema a resolver: ¿cómo identificar de manera más eficiente estructuras en imágenes en retroiluminación de pacientes operados de catarata aplicando un proceso previo de interpolación?

Se identifica como objeto de estudio: procesamiento de imágenes digitales, centrado en la interpolación de imágenes de opacidad capsular.

Campo de acción: interpolación de imágenes médicas

Se tiene como objetivo general: desarrollar un software que permita integrar varios métodos de interpolación previos al proceso de segmentación para identificar de manera más eficiente estructuras y regiones de interés en las imágenes médicas.

Objetivos específicos:

- Elaborar el marco teórico referencial relacionado con la interpolación de imágenes de Opacidad Capsular Posterior.
- Desarrollar un software que permita integrar varios métodos de interpolación previos al proceso de segmentación y evalué los resultados.
- Validar los resultados y la solución propuesta mediante pruebas estadísticas y pruebas del software.

Hipótesis: Con el desarrollo de un software que aplique la interpolación antes de la segmentación de imágenes en retroiluminación, se podrá identificar de manera más eficiente el proceso para la detección de la opacidad capsular.

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

- Identificación de las diferentes técnicas de procesamiento de imágenes médicas.
- Revisión de las técnicas de validación de la segmentación.
- Revisión de las técnicas de interpolación bilineal, de interpolación bicúbica, de interpolación mediante el vecino más cercano y de interpolación mediante splines.
- Estudio de la segmentación mediante el método de umbralización de Otsu.

- Estudio de la segmentación mediante el método de la transformada circular de hough.
- Estudio de la segmentación mediante el método de crecimiento de regiones.
- Descripción de los pasos a seguir para el uso del algoritmo de segmentación propuesto.
- Diseñar la solución propuesta en función de los requisitos especificados.
- Implementación del software diseñado para la medición de la eficacia de la interpolación en la segmentación de imágenes en retroiluminación.
- Validación de los resultados mediante pruebas estadísticas no paramétricas y variables de calidad de la segmentación.
- Validación de la implementación del sistema a partir de la aplicación de pruebas unitarias y funcionales.

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

- **Analítico-Sintético**, él método analítico combina el proceso formal de resolución con el método científico para lograr la resolución del problema. El método analítico parte de la división de un todo en muchas partes para estudiarlas por separado, en tanto que el método sintético reúne un conjunto de conceptos para formular una solución global (12). La utilización de este método permitió tanto caracterizar como comprender los algoritmos de interpolación, los algoritmos de segmentación y la problemática a resolver. Estas operaciones no existen independientemente una de otra, el análisis de un objeto se realiza a partir de la relación que existe entre los elementos que conforman dicho objeto como un todo; y a su vez, la síntesis se produce sobre la base de los resultados previos del análisis (13).
- **Inducción-Deducción**, inducción es el modo de razonar desde lo particular a lo general, desde los hechos a las síntesis y deducción es el modo de razonar desde lo general a lo particular, desde las síntesis generales a las conclusiones particulares. Toda deducción científica es el resultado del previo estudio inductivo del material y se basa en este estudio. A su vez, la inducción se basa en el conocimiento de las leyes generales que rigen el proceso del desarrollo. (14) . La aplicación de este método posibilitó llegar desde las bases de la problemática, a las bases de su resolución analizando todo los criterios relacionados con la trayectoria, desarrollo y evolución de software de análisis

de OCP en pacientes operados de catarata.

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

Capítulo 1. “Fundamentación teórica”: se abordan de manera sintetizada los elementos básicos teóricos, así como la fundamentación de cada uno de los algoritmos de interpolación y de los algoritmos de segmentación a utilizar como parte del procesamiento de imágenes médicas. Además, se describen las herramientas y tecnologías a utilizar durante el desarrollo de la solución. En el mismo son expuestas las razones que derivan en la selección de dichas herramientas y tecnologías; y se explican las principales características de la metodología de software XP y las ventajas que ofrece en la presente investigación.

Capítulo 2: “Propuesta de solución”: se describen las principales características del algoritmo propuesto como solución a la problemática, guiando todo el proceso por las fases definidas por la metodología XP (Programación Extrema). Se definen los requisitos funcionales y no funcionales y se analizan los elementos necesarios para garantizar el éxito en el proceso de desarrollo, tales como: las historias de usuario, patrones de diseño, tarjetas CRC (Clase, Responsabilidad y Colaborador) y los estándares de codificación.

Capítulo 3: “Validación de la solución propuesta”: se da continuidad a las fases de la metodología XP, con la realización de pruebas de caja blanca y de caja negra. Además, ya procesada la imagen se describen y exponen los resultados de la aplicación de la detección de bordes como métrica para validar la calidad.

Capítulo 1: Fundamentación Teórica

1.1. Introducción

En el presente capítulo se realiza la descripción y análisis desde el punto de vista teórico del problema general en que se enmarca la investigación. Como puntos fundamentales se abordan los principales conceptos asociados al dominio del problema, como el de procesamiento digital de imágenes, segmentación e

interpolación de imágenes de Opacidad Capsular. También se exponen los algoritmos de segmentación y de interpolación argumentando sus características e importancia en el procesamiento imágenes provenientes de la Lámpara de hendidura.

1.2. Lámpara de hendidura

Es un dispositivo óptico que se utiliza en oftalmología y optometría para la exploración de las estructuras de la porción anterior al ojo. Consiste en un microscopio binocular dotado de una potente fuente de luz que permite ver en tres dimensiones y con una amplificación que oscila entre 6 y 40 aumentos, las estructuras del polo anterior del ojo: párpado, conjuntiva, córnea, iris, cristalino y cámara anterior. Si se le adaptan unas lentes especiales, es posible visualizar las estructuras más profundas del globo ocular como el humor vítreo, la retina y la papila óptica o disco óptico. El sistema de iluminación va montado sobre un brazo móvil, lo que da la posibilidad de variar el ángulo con el que incide la luz sobre la superficie ocular (15). En la Ilustración 3 podemos observar una Lámpara de Hendidura.

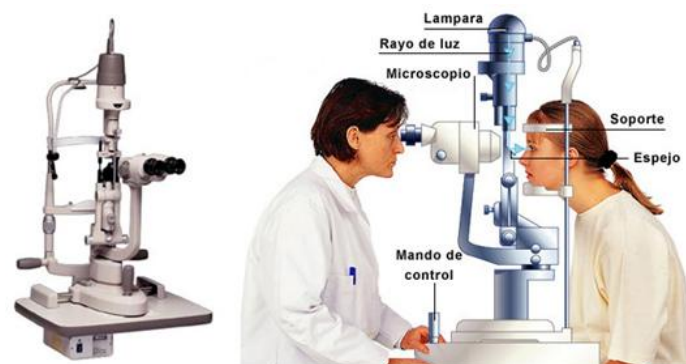


ILUSTRACIÓN 1 . LÁMPARA DE HENDIDURA

1.3. Procesamiento digital de imágenes

Antes de que se pueda definir el procesamiento digital de imágenes se debe definir la palabra imagen. Entonces una imagen es una representación o imitación de un objeto o cosa, es decir una representación gráfica. Una imagen contiene información descriptiva acerca del objeto que representa aunque hay que tener en cuenta que bajo esta definición de imagen se encuentran muchas representaciones no percibibles por el ojo. (16)

Por otra parte la palabra digital se relaciona al cálculo por métodos numéricos o por unidades discretas. De esta forma una imagen digital se define como la representación numérica de un objeto, es un arreglo bidimensional de píxeles donde el valor de cada píxel se representa mediante una función f , donde $f(x, y)$ representa el nivel de brillantez, color o intensidad de la imagen en tales coordenadas. (17)

Entonces se puede definir el procesamiento digital de imágenes como el sometimiento de la representación numérica de objetos a una serie de operaciones para obtener un resultado deseado, es el conjunto de técnicas que se aplican a las imágenes digitales; consiste en alterar la información visual para obtener mejores resultados o para aislar algunas características particulares de la imagen. El procesamiento digital de imágenes toma una imagen y produce una versión modificada de la misma. (18)

A continuación se exponen otros de los principales conceptos relacionados con el procesamiento digital de imágenes:

- **Resolución de una imagen:** indica el nivel de detalle que presenta una imagen, tener mayor resolución se traduce en tener una imagen con más detalle o calidad visual. La resolución de la imagen se representa con dos números enteros, donde el primero es la cantidad de columnas de píxeles (cuántos píxeles tiene la imagen a lo ancho) y el segundo es la cantidad de filas de píxeles (cuántos píxeles tiene la imagen a lo alto), lo cual se puede observar en la Ilustración 2. (19)

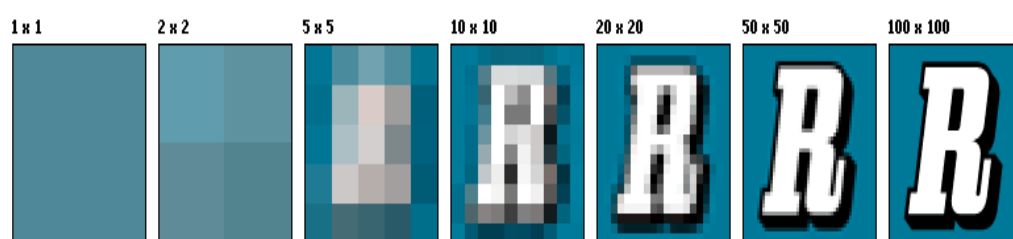


ILUSTRACIÓN 2 . RESULTADO DEL INCREMENTO DE LA RESOLUCIÓN DE UNA IMAGEN

- **Retroiluminación:** se refiere a cualquier técnica de iluminación en la que la luz venga reflejada del iris o de la retina. Se utiliza para iluminar cualquier condición corneal o de cristalino desde atrás. (20)
- **Imágenes multicanal:** una imagen multicanal es un arreglo de imágenes definidas en un dominio común donde a cada píxel se asocia un vector de escalares definido por el valor de cada canal. La dimensión de la imagen

corresponde al número de canales de la imagen, un ejemplo de imagen multicanal se puede ver en la Ilustración 3. (21)



ILUSTRACIÓN 3 . EJEMPLO DE IMAGEN MULTICANAL

- **Escala de grises:** la escala de grises es la representación de una imagen en la que cada píxel se dibuja usando un valor numérico individual que representa su luminancia, en una escala que se extiende entre blanco y negro. Es el modo de color en el que cada píxel se describe dentro de una serie limitada de valores de un mismo tono neutro (22). En la Ilustración 4 se muestra la escala de negro a blanco que contiene los tonos de grises.



ILUSTRACIÓN 4 . ESCALA DE GRISES

1.4. Segmentación

La segmentación de imágenes se define como la partición de una imagen en regiones constituyentes no solapadas, las cuales son homogéneas con respecto a alguna característica como la intensidad, los bordes, la luminancia, el color o la textura. Más precisamente, la segmentación de la imagen es el proceso de asignación de una etiqueta a cada píxel de la imagen de forma que los píxeles que compartan la misma etiqueta también tendrán ciertas características visuales similares. La agrupación de los píxeles se hace a razón de que sus vecinos sean similares en criterios y características como las antes mencionadas. (23)

El resultado de la segmentación de una imagen es un conjunto de segmentos que cubren en conjunto a toda la imagen, o un conjunto de las curvas de nivel extraídas de la imagen. La imagen queda definida por un grupo de objetos, habiendo pasado de un nivel bajo a otro más elaborado o nivel medio visual. La información estará preparada para el reconocimiento e interpretación de la imagen. (24)

El objetivo de la segmentación es simplificar y/o cambiar la representación de una imagen en otra más significativa y más fácil de analizar. La segmentación se usa tanto para localizar objetos como para encontrar los límites de estos dentro de una imagen. Idealmente, un método de segmentación encuentra aquellos conjuntos que corresponden a distintas estructuras o regiones anatómicas de interés en la imagen (23). En la Ilustración 5 se puede apreciar un ejemplo de opacidad según un especialista y de opacidad según un algoritmo de segmentación.

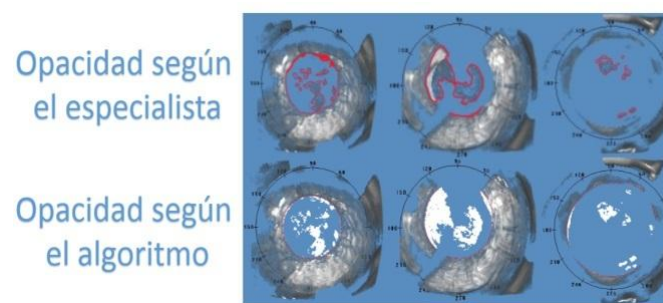


ILUSTRACIÓN 5 . OPACIDAD SEGÚN ESPECIALISTA Y OPACIDAD SEGÚN MÉTODOS DE SEGMENTACIÓN

1.4.1. Métodos de segmentación

En esta sección se describen varios métodos de segmentación comunes de segmentación de imágenes médicas, se define cada uno de ellos y se caracterizan teniendo en cuenta sus ventajas y desventajas. Aunque cada técnica fue creada separadamente, frecuentemente se utilizan múltiples técnicas en conjunto con otras para resolver diferentes problemas de segmentación.

Los algoritmos de segmentación se basan en los siguientes principios (25):

- Discontinuidades del nivel de gris: consisten en segmentar la imagen a partir de los cambios grandes en los niveles de gris entre los píxeles. Las técnicas que utilizan las discontinuidades como base son la detección de líneas, de bordes y de puntos aislados.
- Conectividad: los objetos corresponden a áreas de píxeles con conectividad. Las particiones se corresponden con regiones continuas de píxeles.

- Similitud de niveles de gris: las divisiones de la imagen se hacen agrupando los píxeles que tienen unas características similares. Algunas técnicas que usan esto son la umbralización, el crecimiento de regiones, etc.

1.4.1.1. Umbralización (Segmentación de Otsu)

La umbralización es un método que busca segmentar imágenes escalares creando una partición binaria de las intensidades de las imágenes. Una umbralización trata de determinar un valor de intensidad, llamado umbral, que separa las clases deseadas. La segmentación se logra agrupando todos los píxeles con mayor intensidad al umbral en una clase, y todos los otros píxeles en otra clase. La determinación de más de un valor umbral es un proceso llamado multiumbralización. (26)

La umbralización es una técnica efectiva para obtener la segmentación de imágenes donde estructuras diferentes tienen intensidades contrastantes u otras características diferenciables. Para imágenes escalares, los métodos interactivos pueden estar basados en la apreciación visual del usuario ya que la operación de umbralización es implementable en tiempo real. Su principal limitación es que en su forma más simple solo se generan dos clases y por lo tanto no se puede aplicar a imágenes multicanal. (26)

Al aplicar un umbral, la imagen de niveles de grises queda binarizada; etiquetando con '1' los píxeles correspondientes al objeto y con '0' aquellos que son del fondo. Por ejemplo, si los objetos son claros respecto del fondo, se aplica:

$$f(x,y) = \begin{cases} 1, & f(x,y) > T \\ 0, & f(x,y) \leq T \end{cases} \quad (1)$$

$f(x,y)$ es la función que retorna el nivel de gris del píxel (x,y) , $g(x,y)$ es la función de la imagen binarizada y T es el umbral. En el caso de que los objetos sean oscuros respecto del fondo, la asignación sería a la inversa:

$$f(x,y) = \begin{cases} 1, & f(x,y) < T \\ 0, & f(x,y) \geq T \end{cases} \quad (2)$$

1.4.1.2. Transformada circular de Hough

Este algoritmo trata de detectar formas geométricas sencillas en una imagen. En su procesamiento emplea el conjunto total de la imagen, haciéndolo robusto ante la

presencia del ruido o discontinuidades de las etapas previas. De hecho, para su ejecución requiere de una imagen binarizada en la que se han seleccionado previamente los bordes (27). Por tanto, la entrada al algoritmo de Hough es una imagen en la que se ha hecho la detección de los bordes. Hough, desde una perspectiva global, intentará extraer primitivas de más alto nivel como son líneas, circunferencias, elipses o cualquier tipo de curva parametrizada o no. El mayor inconveniente es su alto coste computacional. (28)

Para este caso, el espacio paramétrico será 3D, correspondiente a la posición del centro del círculo, (x_0, y_0) , y al radio, r :

$$(x_i - x_0)^2 + (y_i - y_0)^2 = r^2 \quad (3)$$

Las votaciones de cada celdilla del espacio paramétrico se realiza variando la posición del centro del círculo, (x_0, y_0) , para cada píxel etiquetado, (x_i, y_i) , obteniendo los distintos valores del radio, r . Por tanto, entrado en la rejilla de acumulación por posición del centro del círculo y del radio calculado se procede a depositar los votos correspondientes. Los círculos seleccionados son aquellos que superen un cierto umbral de votos recibidos. Una forma de ahorro en el cálculo es prefijar el valor del radio del círculo. Si se conoce el radio, junto con la información de la orientación del gradiente del píxel etiquetado, (x_i, y_i) , las posibles coordenadas del centro del círculo dan una nube de puntos limitados. La idea básica es encontrar curvas que puedan ser parametrizadas como círculos. Se puede analíticamente describir un segmento de línea de varias formas, la ecuación conveniente para describir un conjunto de líneas es la notación paramétrica o normal. (28)

$$p = x \cos \theta + y \sin \theta \quad (5)$$

Las ventajas de esta detección de círculos, provienen de las características inherentes de las transformadas de Hough, la inmunidad al ruido y que es capaz de detectar círculos, aunque exista oclusión del mismo. Por otro lado, en este tipo de curvas, al ser cerradas, no hay problemas del carácter infinito de las líneas rectas. Los círculos están definidos en la imagen, sin ninguna expansión. En la Ilustración 6 se puede observar un ejemplo.

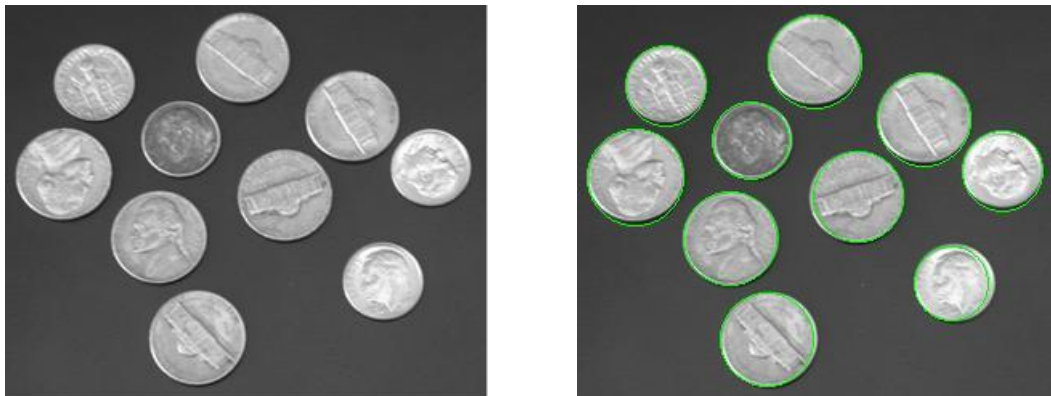


ILUSTRACIÓN 6 . IMAGEN DE CÍRCULOS SIN PROCESAR Y APLICACIÓN DE LA TRANSFORMADA CIRCULAR DE HOUGH A DICHA IMAGEN

1.4.1.3. Crecimiento de regiones

La estrategia a emplear en este tipo de segmentación está basada en los criterios de similitud y continuidad de los píxeles que forman una región. Bajo esta perspectiva, la imagen se considera formada por n regiones disjuntas, cada una de las cuales tiene agrupada a los píxeles por alguna propiedad que los hace ser característicos de esa zona y discrepantes respecto al resto (29). Desde el punto de vista formal, las condiciones de la segmentación orientada a regiones son:

- a) $I = \cup_{i=1}^n R_i$
 - b) $R_i \cap R_j = \emptyset \quad i \neq j$
 - c) $p_i \in R_j \leftrightarrow P_{R_j}(p_i) = 1$
 - d) $p_i \in R_j \leftrightarrow P_{R_j}(p_i) = 0$
- (6)

Siendo I la imagen, R_i una región de ésta, p_i un píxel cualquiera de la imagen y P_{R_j} la regla de similitud de la región j . Las dos primeras condiciones describen la segmentación como un proceso de partición de la imagen en regiones disjuntas; mientras, las otras dos restantes hacen referencia a las propiedades de similitud de los píxeles agrupados y de discrepancia respecto al resto (30).

A partir de píxeles semillas, se hace crecer la región considerando alguna regla de similitud y considerando la propiedad de conectividad de los píxeles. Una vez definida una semilla, se analiza los vecinos según la regla de similitud de la región P_{R_j} , los

píxeles que cumplan la propiedad se añadirán a la región de crecimiento; aquellos que no cumplan significarán que pertenecen a otra región. Con los nuevos píxeles unidos a la región de crecimiento, se volverá a procesar sus nuevos vecinos (30). El algoritmo de crecimiento finalizará cuando todos los vecinos hayan sido estudiados. Nótese el carácter de conectividad y similitud de la técnica de segmentación de crecimiento de regiones en la Ilustración 7.

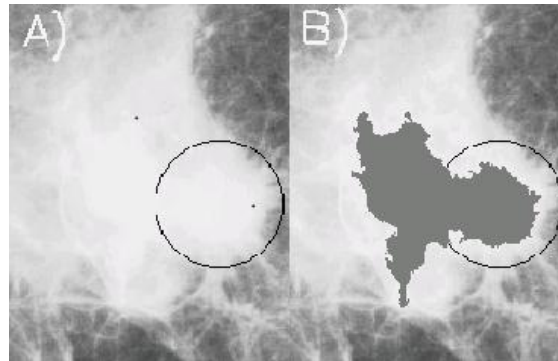


ILUSTRACIÓN 7 . A) REGIÓN DE UNA LESIÓN MUSCULAR B) OBJETO SEGMENTADO MEDIANTE CRECIMIENTO DE REGIONES

1.5. Interpolación

La interpolación es el proceso matemático consistente en calcular nuevos datos a partir de los disponibles procurando que compartan la misma forma; en fotografía digital, supone crear nuevos píxeles a partir de los existentes, de modo que la imagen resultante venga a representar lo mismo que la original, sólo que con más píxeles. La interpolación se usa, para muchos procesos relacionados con la imagen digital como la rotación de imágenes, deformación, escalado, reasignación, y transformaciones no lineales; pero vamos a centrarnos en el uso más común de la interpolación de imagen: la ampliación digital, es decir, aumentar la resolución de la imagen. (31)

La idea de la interpolación es poder estimar $f(x)$ para un x arbitrario, a partir de la construcción de una curva o superficie que une los puntos donde se han realizado las mediciones y cuyo valor si se conoce (32). En la Ilustración 8 se puede observar un ejemplo de interpolación.



ILUSTRACIÓN 8 . EJEMPLO DE INTERPOLACIÓN

1.5.1. Métodos de Interpolación

En el campo de la fotografía y mundo de la imagen digital, la interpolación aplica métodos para conseguir un tamaño mayor de una imagen inicial, rellenando la información que falta con datos generados a partir de algoritmos específicos. Existen varios algoritmos, entre ellos interpolación bilineal, interpolación bicúbica, Interpolación por Splines e Interpolación mediante el vecino más cercano.

1.5.1.1. Interpolación bilineal

La interpolación bilineal es una extensión de la interpolación lineal para interpolar funciones de dos variables en una malla regular de dos dimensiones. La idea principal es realizar una interpolación lineal en una dirección, y después en la otra; se interpola linealmente a lo largo de cada fila de una imagen, y posteriormente el resultado se interpola linealmente en la dirección por columnas. Aunque cada uno de estos pasos es lineal, la interpolación en su conjunto no es lineal sino cuadrática. (32)

Esta técnica es utilizada para calcular valores de una ubicación de cuadrícula basada en cuatro celdas de cuadrícula cercanas. Asigna el valor de la celda de salida tomando el promedio ponderado de las cuatro celdas vecinas en una imagen para generar nuevos valores. (32)

En la Ilustración 9 se muestra la vecindad alrededor de (p', q') , donde $a = p' - p$, $b = q' - q$.

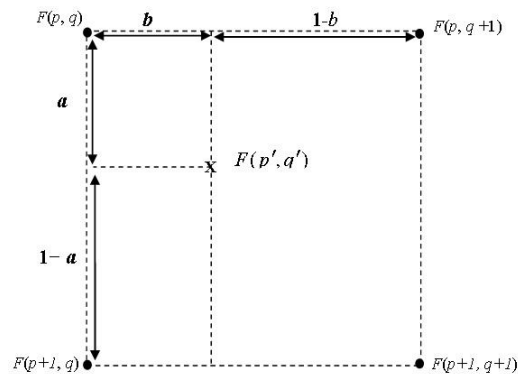


ILUSTRACIÓN 9 . VECINDAD EN EL PROCESO DE INTERPOLACIÓN BILINEAL

Realizando las dos interpolaciones lineales horizontales de $F(p, q)$ a $F(p, q+1)$ y de $F(p+1, q)$ a $F(p+1, q+1)$ se tiene como resultado $(1-b)F(p, q) + bF(p, q+1)$ y $(1-b)F(p+1, q) + bF(p+1, q+1)$, ahora bien, realizando nuevamente la interpolación lineal vertical sobre los dos valores hallados anteriormente se obtiene el valor de intensidad del pixel interpolado

$$F(p', q') = (1 - a)[(1 - b)F(p, q) + bF(p, q + 1)] + a[(1 - b)F(p + 1, q) + bF(p + 1, q + 1)] \quad (7)$$

1.5.1.2. Interpolación bicúbica

La interpolación bicúbica se basa en considerar la vecindad de los 4x4 píxeles conocidos más cercanos al que se desea interpolar, es decir, un total de 16 píxeles. O sea, se aproxima localmente el nivel de gris en la imagen original mediante la generación de una superficie polinómica bicúbica. Equivale a aplicar, en primer lugar, cuatro interpolaciones unidimensionales horizontales entre cuatro píxeles (dos a un lado y dos a otro) y posteriormente otra interpolación unidimensional, en este caso vertical, entre los cuatro valores obtenidos anteriormente (33). En la Ilustración 10 se muestra la vecindad alrededor de (p', q') .

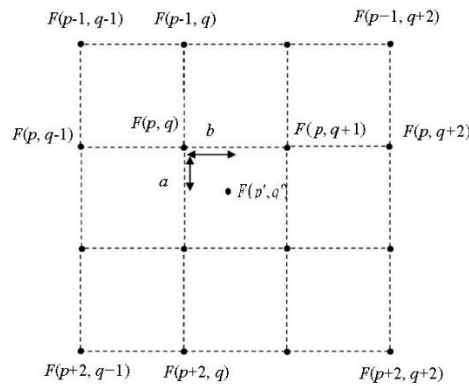


ILUSTRACIÓN 10 . VECINDAD EN EL PROCESO DE INTERPOLACIÓN BICÚBICA

En este caso el pixel interpolado se puede expresar en forma compacta como:

$$F(p', q') = \sum_{m=-1}^2 \sum_{n=-1}^2 F(p + m, q + n) \cdot H_c[m - a] \cdot H_c[n - b] \quad (8)$$

Este algoritmo completo está caracterizado por ser un buen compromiso entre tiempo de procesado y calidad de resultado, produce imágenes con buen grado de nitidez, es un procedimiento estándar en programas de edición de imágenes y además actualmente es uno de los algoritmos de interpolación más utilizados. (33)

1.5.1.3. Interpolación mediante spline

Una función spline está formada por varios polinomios, cada uno definido en un intervalo y que se unen entre si bajo ciertas condiciones de continuidad. Dicho método tiene dos características principales: proporciona una función interpoladora que se comporta suavemente y la segunda derivada de la función de interpolación es continua (33). Se define un spline de orden m en una serie de $n + 1$ puntos de interpolación $\{ (x_0, y_0) , \dots (x_n, y_n) \}$ como un conjunto de n funciones $S_k(x)$ (frecuentemente polinomios de orden m) definidas en el intervalo $[x_k, x_k + 1]$, que satisfacen

$$\begin{aligned} S_k(x_k) &= S_{k+1}(x_k) \\ S'_k(x_k) &= S'_{k+1}(x_k) \\ S''_k(x_k) &= S''_{k+1}(x_k) \\ &\vdots \\ &\vdots \\ &\vdots \end{aligned}$$

$$S_k^{m-1}(x_k) = S_{k+1}^{(m-1)}(x_k)$$

(9)

1.5.1.4. Interpolación mediante el vecino más cercano

La interpolación por el vecino más cercano es un método simple de interpolación multivariable en una o más dimensiones, es el más rápido pero el menos preciso. Es el problema de aproximar el valor de una función para un punto en el espacio cuando se administra el valor de esa función en puntos alrededor de ese punto. Dicho algoritmo selecciona el valor del punto más cercano y no tiene en cuenta los valores de puntos vecinos, produciendo un interpolante constante por trozos (34). En la Ilustración 11 se muestra la vecindad alrededor del punto a interpolar,

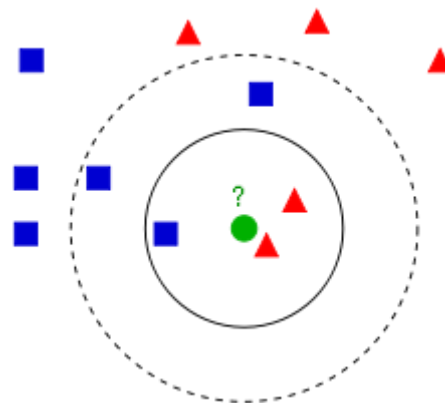


ILUSTRACIÓN 11 . VECINDAD EN EL PROCESO DE INTERPOLACIÓN POR EL VECINO MÁS CERCANO

Dicho algoritmo consiste en escoger como el valor del píxel de una imagen de salida el valor del píxel de entrada más cercano en la dirección inversa. Este tipo de interpolación logra producir un error espacial relativo a lo más de $\frac{1}{\sqrt{2}}$ píxel por unidad.

Se basa en tomar el píxel de la posición $(x, y) = (a, b)$ en la imagen original, y transformarlo mediante las ecuaciones de distorsión

$$x' = r(x, y) = C_1x + C_2y + C_3xy + C_4 \quad (10)$$

$$y' = s(x, y) = C_5x + C_6y + C_7xy + C_8 \quad (11)$$

para obtener las correspondientes coordenadas (a', b') que pueden ser enteras o racionales según los coeficientes, posteriormente se elige el vecino de coordenadas

enteras más cercano a (a', b') y finalmente se asigna el nivel de gris de este vecino al píxel ubicado en (a, b) lográndose un valor aproximado al píxel original. El método se repite de la misma forma con cada uno de los píxeles restantes de la imagen hasta lograr una imagen restaurada $f'(x, y)$ que aproxima a $f(x, y)$. (34)

1.6. Metodología, herramientas y lenguaje de programación utilizado durante el desarrollo

A continuación se describen las herramientas y tecnologías que se han utilizado durante el desarrollo del sistema, también se abordan las características de cada una de ellas así como su función en detalle para la resolución del problema. Se explican los principales elementos, ventajas y fases de la metodología de software a usar y se argumenta acerca del lenguaje de programación y el IDE de desarrollo, donde se hace una breve explicación de cada uno.

1.6.1. Metodología utilizada para el desarrollo

Se ha elegido la metodología ágil XP para el desarrollo de la presente investigación, debido a que se está en presencia de un proyecto pequeño, de corta duración y la prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software. El cliente es parte del equipo de desarrollo, el equipo de desarrollo es de solo una persona. Además de la dificultad para un equipo de desarrollo grande el adoptar una metodología robusta genera gran cantidad de documentación y alta resistencia a los cambios, por lo cual se ha identificado la Programación Extrema (XP) como una alternativa acertada, además de ser la metodología de desarrollo del grupo de investigación AIRI (Artificial Intelligence: Research and Innovation) (35), grupo al cual pertenece esta investigación.

Ventajas de XP (35):

- 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.

Fases de XP:

- **Planificación:** es la fase donde los desarrolladores y clientes establecen los

tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada iteración. (36)

- **Diseño:** la metodología XP hace especial énfasis en los diseños simples y claros. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando. (36)
- **Implementación:** en la fase de codificación de desarrolla en función de cada historia de usuario, además de ser fase donde se definen las tareas de la ingeniería y los tiempos reales en se realizaron cada una de las funcionalidades especificadas, en la cual la implementación, debe realizarse de acuerdo los estándares de codificación. (36)
- **Pruebas:** estas pruebas se realizan al final del ciclo en el que se desarrollan, para verificar que las iteraciones no han afectado a las anteriores. Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para prever que no vuelvan a ocurrir. (36)

1.6.2. Lenguaje de Programación

Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. (37)

Debido a que el software PANDOC está desarrollado en el lenguaje de programación Java y el resultado de este proyecto puede constituir un módulo de esa aplicación informática se escogió el mismo para desarrollar la solución propuesta. Java es un lenguaje de programación de propósito general, concurrente y orientado a objetos. Aplicaciones creadas en este lenguaje de programación pueden ser ejecutadas en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Es uno de los lenguajes de programación más populares en uso, y aunque su sintaxis deriva en gran medida de C y C++, tiene menos utilidades de bajo nivel que cualquiera de ellos (38). Además, posee un recolector automático de basura (garbage collector) lo cual independiza al programador de tener que administrar memoria solicitada dinámicamente de forma manual (39).

1.6.3. Entorno Integrado de Desarrollo

Para el desarrollo de la solución, como Entorno de Desarrollo Integrado (IDE, según sus siglas en inglés) se seleccionó el NetBeans 8.2 ya que es un entorno de desarrollo integrado libre, de código abierto y contiene todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Las aplicaciones construidas a partir de estos módulos pueden ser extendidas agregándole nuevos módulos, debido a que los módulos pueden ser desarrollados independientemente. Además es un entorno de desarrollo disponible para varios sistemas operativos como Windows, Mac, Linux y Solaris (40).

Dicho IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles) y adicionalmente, el NetBeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML. El NetBeans C/C++ Pack soporta proyectos de C/C++, mientras el PHP Pack, soporta PHP 5. (41)

Algunas de las características de la aplicación son:

- Gestión de la interfaz de usuario (menús y barras de herramientas)
- Gestión de configuración de usuario
- Gestión de almacenamiento (guardar o cargar algún tipo de dato)
- Gestión de ventana
- Marco Asistente (soporta diálogos para a paso)
- Librería visual de NetBeans
- Herramientas de desarrollo integrado

1.6.4. Herramienta para el modelado

Como herramienta para el modelado de la solución se ha seleccionado Visual Paradigm for UML, una de las líderes del mercado de las llamadas herramientas de

Ingeniería de Software Asistida por Computadora (CASE, según sus siglas en inglés). Dicha herramienta 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. Además es muy completa, fácil de usar y con soporte multiplataforma. Su uso es sencillo para la creación de todo tipo de diagramas UML, para los que dispone de un número considerable de estereotipos, permitiendo mayor entendimiento de los mismos (42). 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 (42).

1.7. Conclusiones Parciales

- En este capítulo se realizó un análisis de las bases teóricas referentes a la OCP y de los principales conceptos a tener en cuenta para dar solución al problema planteado como son imagen digital, interpolación y segmentación; lo cual permitió una mayor comprensión del tema de investigación.
- Se realizó un estudio de los algoritmos de interpolación a utilizar lo cual nos permitió comprender el funcionamiento lógico y la diferencia del nivel de detalle en cuanto a estimación de valores de píxeles de cada uno de estos algoritmos.
- Se identificaron y caracterizaron las herramientas y tecnologías de desarrollo de software a utilizar para el desarrollo de la propuesta de solución destacándose la metodología XP, el NetBeans como IDE de desarrollo, Java como lenguaje de programación y el Visual Paradigm for UML como herramienta de modelado.

Capítulo 2: Análisis y diseño de la propuesta de solución

2.1. Introducción

En el presente capítulo se describen los principales elementos que se tuvieron en cuenta en el desarrollo de la solución propuesta, las técnicas de procesamiento de imágenes empleadas, los patrones de diseño y los estándares de codificación. El trabajo se organizó en función de las fases que define la metodología XP y además se hace referencia a los artefactos generados en cada una de ellas.

2.2. Descripción de la solución

El proyecto “Análisis de los efectos de la interpolación en la segmentación de imágenes de opacidad capsular” tiene como objetivo construir una aplicación basada en el procesamiento de imágenes digitales. Consta de tres actividades fundamentales: Obtención de la imagen, Interpolación de la imagen y Segmentación de la imagen.

Para la realización de este proyecto se decidió integrar al proceso de segmentación un método previo de interpolación con el fin de obtener mayor calidad en las imágenes resultantes. En la Ilustración 12 se muestra el diagrama de flujo que describe la solución propuesta.

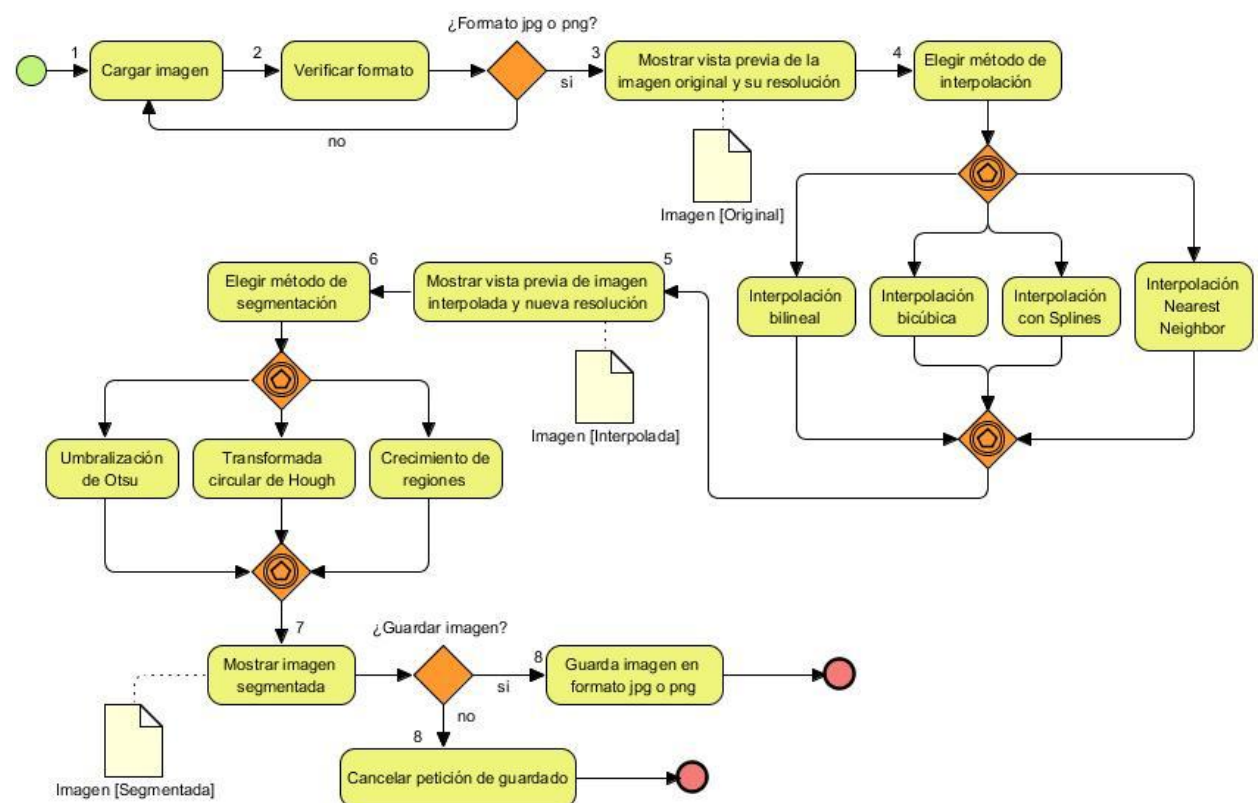


ILUSTRACIÓN 12 . DESCRIPCIÓN DE LA SOLUCIÓN

Paso 1. Cargar imagen

El usuario debe elegir la imagen que desea procesar y confirmar la elección para posteriormente subirla al sistema.

Paso 2. Verificar formato

Se verifica que la imagen seleccionada por el usuario este en los formatos jpg o png

para poder procesarla, en caso de que esta condición no se cumpla el usuario debe reelegir otra imagen que cumpla con estas características.

Paso 3. Mostrar vista previa de la imagen original y su resolución

Muestra una vista previa de la imagen original y su resolución actual, además transforma la imagen en `BufferedImage` para obtener la matriz de píxeles de la imagen.

Paso 4. Elegir método de interpolación

Se selecciona el algoritmo de interpolación mediante el cual se aumentará la calidad de la imagen para luego segmentarla. Devuelve como resultado una imagen con mayor resolución debido a que se ha aumentado el tamaño de la matriz de píxeles insertando nuevos píxeles en ella que toman valores dependiendo de su vecindad y de la aplicación del método de interpolación seleccionado.

Algoritmos de interpolación:

- Interpolación bilineal:

Supóngase que se quiere encontrar el valor para la función f desconocida en el punto $P = (x, y)$. Se conoce el valor de f en los cuatro puntos $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$ y $Q_{22} = (x_2, y_2)$. Primero se hace una interpolación lineal en la dirección x y después se hace una interpolación lineal en la dirección y , esto proporciona una estimación de $f(x, y)$. Hay que tener en cuenta que se obtienen los mismos resultados si la interpolación se hace primero en la dirección y y después en la dirección x .

- Interpolación bicúbica:

Al igual que en la Interpolación bilineal se quiere encontrar el valor para la función f en el punto $P = (x, y)$ y se conoce el valor de f en los 16 puntos $Q_n = (x_n, y_n)$. Se hace una interpolación lineal en la dirección x y se hace una interpolación lineal en la dirección y , esto proporciona una estimación de $f(x, y)$.

Este método implica ajustar una serie de polinomios cúbicos a los valores de brillo contenidos en una matriz de píxeles de 4×4 que rodean la dirección calculada. En primer lugar, cuatro polinomios cúbicos $f(i)$ están instalados en los puntos de control en la dirección y (la elección de la dirección de partida es arbitraria).

A continuación, la parte fraccionaria de la dirección del pixel calculado en la dirección del eje y se usa para encajar con otro polinomio de tercer grado en la dirección x , en base a los valores de brillo interpolados que se encuentran en las curvas.

Sustituyendo la parte fraccionaria de la dirección del pixel calculada en la dirección x en el polinomio cúbico resultante proporciona entonces el valor de brillo del pixel interpolado.

- Interpolación mediante Splines:

Se coloca un polinomio de grado 3 en cada intervalo de datos como se muestra en la Ilustración 13, cumpliendo las siguientes condiciones:

- Debe haber certeza de que $y_i = f(x_i)$
- $x_i < x_{i+1}$, No se permite que hayan dos x_i iguales $\forall i, y$

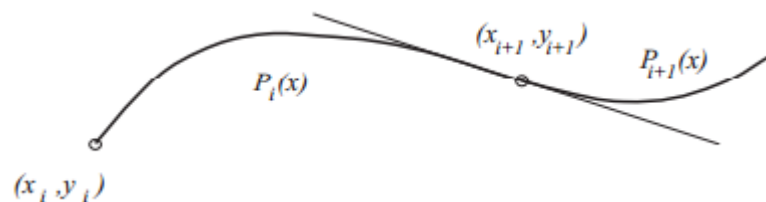


ILUSTRACIÓN 13 . SPLINES PARA INTERPOLACIÓN

Sea $P_i(x)$ el polinomio de aproximación de la verdadera relación $f(x)$ en el intervalo $(x_i, y_i) \sim (x_{i+1}, y_{i+1})$ con las siguientes restricciones:

- $P_i(x_{i+1}) = P_{i+1}(x_{i+1})$, esto es, que los polinomios se unan,
- $P'_i(x_{i+1}) = P'_{i+1}(x_{i+1})$, misma pendiente en la unión,
- $P''_i(x_{i+1}) = P''_{i+1}(x_{i+1})$, misma concavidad en la unión.

- Interpolación mediante el vecino más cercano:

En este método, la parte fraccionaria de las coordenadas del pixel se descartan y el valor de brillo de los pixeles en la dirección resultante en la imagen de origen se copia en la imagen ampliada. Debido a la inexactitud de la correspondencia espacial entre las dos imágenes, se harán más copias de ciertos pixeles de la imagen original que a otros. Esto puede provocar distorsión espacial de las características de la imagen con zoom, la interpolación resultante del vecino más próximo es por lo tanto poco confiable.

Paso 5. Mostrar vista previa de imagen interpolada y nueva resolución

Muestra una vista previa de la imagen ya interpolada y su nueva resolución obtenida

como parte del proceso de interpolación.

Paso 6. Elegir método de segmentación

Se selecciona el algoritmo de segmentación a utilizar como parte del procesado de la imagen original, toma como entrada la imagen obtenida de la interpolación realizada, recorre su matriz de píxeles y extrae todos aquellos píxeles o regiones de interés.

Algoritmos de segmentación:

- Segmentación mediante umbralización de Otsu:

Basta elegir un umbral dentro de los niveles de grises (en este caso 180), de tal forma que el histograma forme un valle en ese nivel. Todos los niveles de grises contenidos en los píxeles de la imagen menores al umbral calculado se convertirán en negro y todos los mayores en blanco.

- Segmentación mediante transformada circular de Hough:

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

- Segmentación mediante crecimiento de regiones:

Este algoritmo agrupa los píxeles o subregiones de la imagen en regiones mayores basándose en un criterio prefijado. Normalmente se empieza con unos puntos para formar una determinada región, añadiendo aquellos píxeles vecinos que cumplan la propiedad especificada (por ejemplo, que estén en un rango de nivel de gris).

La propiedad considerada en el crecimiento de regiones debe tener en cuenta la información sobre conectividad o adyacencia de la imagen. Sea $f(x,y)$ la imagen de entrada; $S(x,y)$ la matriz semilla conteniendo unos en las localizaciones de los píxeles semilla y ceros en el resto; y Q un predicado para ser aplicado en cada posición (x,y) . Asumimos que f y S con matrices del mismo tamaño.

Un algoritmo de crecimiento de regiones basado en la conectividad consiste en:

- a. Encuentra todas las componentes conexas en $S(x, y)$ y erosiona cada componente conexa a un píxel; etiqueta todos los píxeles encontrados con unos y el resto de píxeles con ceros.
- b. Forma una imagen f_Q tal que, $f_Q(x, y) = 1$ si la imagen f satisface el predicado en la posición (x, y) ; en caso contrario $f_Q(x, y) = 0$.
- c. Sea g una imagen formada anexando a cada píxel semilla de S todos los píxeles 8 – conexos tales que $f_Q = 1$.
- d. Etiqueta cada componente conexa de g (1, 2, 3, ...). La imagen resultante es la imagen segmentada usando crecimiento de regiones.

.Paso 7. Mostrar imagen segmentada

Muestra la imagen resultante obtenida del proceso como un todo, dicha imagen fue interpolada para aumentar su nivel de detalle y posteriormente segmentada para extraer las zonas de interés.

Paso 8. Guardar imagen en formato jpg o png

Por último, posibilita guardar la imagen resultante en formatos jpg o png en cualquier directorio deseado por el usuario.

2.2.1. Requisitos Funcionales

Los requisitos funcionales (RF) describen lo que el sistema debe hacer para mostrar la solución propuesta, así como las condiciones y capacidades que debe cumplir la herramienta. (43)

- RF1: Cargar imagen
- RF2: Aplicar a la imagen el método de Interpolación Bilineal
- RF3: Aplicar a la imagen el método de Interpolación Bicúbica
- RF4: Aplicar a la imagen el método de Interpolación mediante Splines.
- RF5: Aplicar a la imagen el método de Interpolación mediante el vecino más cercano.
- RF6: Segmentar imagen mediante el método de Umbralización de Otsu.
- RF7: Segmentar imagen mediante el método de la Transformada Circular de Hough
- RF8: Segmentar imagen mediante el método de Crecimiento de Regiones.

- RF9: Guardar Imagen

2.2.2. Requisitos no funcionales

Los requisitos no funcionales imponen restricciones en el diseño o la implementación del sistema. Se relacionan con atributos de calidad sistémica que conllevan al buen funcionamiento del sistema como un todo y se refieren a propiedades que debe cumplir el software, tales como capacidad de almacenamiento, fiabilidad o mantenibilidad. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. No son parte de la razón fundamental del producto pero si son necesarios para hacer funcionar el producto de la manera deseada. A continuación, se presentan los requisitos no funcionales definidos (43):

Requisito de usabilidad: el sistema puede ser utilizado en diferentes ramas y por personas sin mucho conocimiento de informática por la fácil utilización que este tiene.

Requisitos de fiabilidad: para el funcionamiento del sistema son necesarios los siguientes recursos:

- 1Gb de memoria RAM
- Procesador Intel(R) Celeron(R)

Requisitos de portabilidad: la herramienta podrá ser usada bajo cualquier sistema operativo siempre que tenga instalada la máquina virtual de Java por la facilidad de portabilidad que tiene el lenguaje en que fue programada.

2.2.3. Historias de Usuario

Las historias de usuario conforman la parte central de muchas metodologías de desarrollo ágil, tales como XP, son técnicas utilizadas para la especificación de requisitos. Estas son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. (43)

Las historias de usuario definen lo que se debe construir en el proyecto de software, tienen una prioridad definida por el cliente de manera que se indica cuáles son las más importantes para el resultado final, son divididas en tareas y su tiempo es

estimado por los desarrolladores y además permiten responder rápidamente a los requisitos cambiantes. (43)

Según Kent Beck cada HU recoge al menos los siguientes aspectos (44):

- 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
- 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

TABLA 1 . HU: CARGAR IMAGEN

Historia de Usuario	
Número: 1	Nombre de la Historia de Usuario: Cargar imagen.
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1semanas
Riesgo en desarrollo: Bajo	Puntos reales: 1 semanas

Programador responsable: Ismael Montano Muñoz
Descripción: el usuario selecciona la imagen deseada desde un directorio y agrega al sistema. Cargar imagen: permite agregar la imagen al sistema para posteriormente procesarla.

TABLA 2 . HU: APLICAR A LA IMAGEN EL MÉTODO DE INTERPOLACIÓN BILINEAL

Historia de Usuario	
Número: 2	Nombre de la Historia de Usuario: Aplicar a la imagen el método de interpolación bilineal.
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Alto	Puntos reales: 3 semanas
Programador responsable: Ismael Montano Muñoz	
Descripción: se interpola la imagen utilizando la interpolación bilineal. Procesar imagen: se obtiene una imagen Interpolada mediante el método Bilineal.	

2.2.4. Estimación de esfuerzo por historias de usuario

Las historias de usuario (HU) sirven para registrar los requerimientos de los clientes y

son utilizadas para crear las pruebas de aceptación y realizar la estimación de cada una de las iteraciones durante la fase de planificación (45). El tratamiento de las HU es muy dinámico y flexible.

La estimación de esfuerzo asociado a la implementación de las historias de usuario la establece los programadores utilizando como medida el punto. Un punto, equivale a una semana de programación. (45)

TABLA 3 . ESTIMACIÓN DE ESFUERZOS POR HU

Historias de usuario	Puntos de estimación
Cargar imagen	1
Aplicar a la imagen el método de Interpolación Bilineal	3
Aplicar a la imagen el método de Interpolación Bicúbica	3
Aplicar a la imagen el método de Interpolación mediante Splines	3
Aplicar a la imagen el método de Interpolación mediante el vecino más cercano	3
Segmentar imagen mediante el método de Umbralización de Otsu	2
Segmentar imagen mediante el método de la Transformada Circular de Hough	2
Segmentar imagen mediante el método de Crecimiento de regiones	2
Guardar Imagen	1

2.2.5. Plan de Iteraciones

Ya definidas las historias de usuarios e identificado el tiempo para la su implementación, se ha diseñado un plan de iteraciones que contiene cada una de las

historias de usuario y define exactamente qué HU es implementada en cada iteración.

TABLA 4 . PLAN DE ITERACIONES

Iteraciones	Historias de usuario a implementar	Duración
1	HU 1	7
	HU 2	
	HU 3	
2	HU 4	6
	HU 5	
3	HU 6	4
	HU 7	
4	HU 8	3
	HU 9	

2.2.6. Plan de Entrega

El plan de entregas tiene como objetivo definir las historias de usuario por iteraciones durante el transcurso del proyecto y además la fecha de entrega de cada una de dichas iteraciones. (45)

TABLA 5 . PLAN DE ENTREGA

Iteraciones	Historias de usuario	Fecha de Entrega
Iteración 1	Cargar imagen	20 / 2 / 2018
	Aplicar a la imagen el método de Interpolación Bilineal	
	Aplicar a la imagen el método de Interpolación Bicúbica	
Iteración 2	Aplicar a la imagen el método de Interpolación mediante Splines	7 / 4 / 2018

	Aplicar a la imagen el método de Interpolación mediante el vecino más cercano	
Iteración 3	Segmentar imagen mediante el método de Umbralización de Otsu	7 / 5 / 2018
	Segmentar imagen mediante el método de la Transformada Circular de Hough	
Iteración 4	Segmentar imagen mediante el método de Crecimiento de regiones	1 / 6 / 2018
	Guardar imagen	

2.3. Diseño

En el presente acápite se hace referencia a los elementos que conforman el diseño del sistema a construir, que acorde a lo planteado por la metodología XP debe ser un diseño simple, sencillo y de fácil interacción, basado principalmente en el desarrollo de las tarjetas Clases, Responsabilidad y Colaboración (CRC), los Estándares de Codificación, los Patrones de Diseño y la Propuesta de Solución.

2.3.1. Tarjetas CRC

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) son utilizadas para representar las responsabilidades de las clases y sus interacciones y ayudan al equipo de desarrollo a definir actividades durante el diseño del sistema. Estas tarjetas permiten trabajar con una metodología basada en objetos, permitiendo que el equipo de desarrollo completo contribuya en la tarea del diseño (44). El título de la tarjeta es el nombre de la clase en cuestión y posteriormente se sitúa a la izquierda las responsabilidades de la clase, y a la derecha las clases que se utilizan para cumplir con cada responsabilidad. Las tarjetas determinan el comportamiento de cada actividad. A continuación, se muestra la tarjeta CRC de la Clase Escala de Grises

TABLA 6 . TARJETA CRC DE LA CLASE ESCALA DE GRISES

Tarjeta CRC

Clase: Escala de Grises	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Permite modificar la luminosidad de los pixeles llevándolos a una escala entre blanco y negro. • Garantiza que la imagen cumpla los requisitos para poder realizar ciertos métodos de Segmentación. 	<ul style="list-style-type: none"> • Preprocesamiento • Procesamiento

2.3.2. Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, establezca un estándar de codificación para asegurarse de que se trabaje de forma coordinada. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debe establecer cómo operar con la base de código existente. (46)

- Utiliza los comentarios para explicar qué hace el código.
- Usa nombres significativos para las clases, métodos, propiedades y variables.
- Los nombres de las clases serán con mayúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán de igual forma.
- Los identificadores para las variables y los parámetros serán con letras en minúsculas y en caso de ser un nombre compuesto las siguientes palabras se escribirán con mayúscula.
- Los nombres de variables o funciones deben ser lo suficientemente descriptivos, sin exceder de 30 caracteres.
- Los métodos de clases, se define que la primera letra debe ser minúscula, si son varias palabras se debe de intercalar entre minúsculas y mayúsculas.
- Se define una declaración por línea.
- Se evita asignar el mismo valor a varias variables en la misma sentencia, pues dificulta su lectura.

2.3.3. Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces (33). En este caso se han utilizado tres patrones fundamentales: Experto, Creador y Controlador.

Experto: el patrón Experto se usa para asignar responsabilidades y expresa simplemente la "intuición" de que los objetos realizan acciones relacionadas con la información que poseen (34).

- Este patrón se utiliza en diferentes clases, que contienen información y varias funcionalidades tal como se evidencia en la clase Image.

Image
-imgBuffer: BufferedImage -ancho: double -alto: double -nombreImagen: String
+Image() +getNombre(): String +getBufferedImage(): BufferedImage +getImage(): Image +getAlto(): double +getAncho(): double

Creador: el patrón creador guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que debe conectar con el objeto producido en cualquier evento (32).

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

```
ImageIcon imagen1 = new ImageIcon(nombreArchivo);
ImageIcon imagen2 = new ImageIcon(nombreArchivo);
imagen1 = new ImageIcon(imagen1.getImage().getScaledInstance(200, -1, Image.SCALE_DEFAULT));
```

Controlador: mantiene el control actuando como intermediario entre una determinada interfaz y el algoritmo que la implementa, además permite dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y

disminuir el acoplamiento (35).

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

```
switch (metodo_de_interpolacion) {  
    case "Bilineal":  
        q = Scalr.bilineal(src, Scalr.Method.ULTRA_QUALITY,  
            jLabel1.setIcon(new ImageIcon(q));  
            int dimensiones1 = q.getWidth();  
            String dim1 = Integer.toString(dimenciones1);  
            int dimensiones2 = q.getHeight();  
            String dim2 = Integer.toString(dimenciones2);
```

2.4. Conclusiones Parciales

- Se definió la propuesta de solución del problema lo que determinó la secuencia algorítmica a seguir y las bases del desarrollo.
- En este capítulo se describió la propuesta de solución generando 10 HU a desarrollar en 4 iteraciones, las cuales describen las funcionalidades del sistema y puntualizan el orden de su implementación.
- Se determinaron los requerimientos funcionales y no funcionales del sistema lo cual ha permitido determinar las funcionalidades básicas a desarrollar durante el proceso.
- Se describió la utilización de los patrones de diseño y de los estándares de codificación los cuales permitieron un desarrollo organizado, controlado y de fácil comprensión para otros desarrolladores.

Capítulo 3: Resultados y Validación.

3.1. Introducción

En el presente capítulo se describen los resultados obtenidos y también se desarrollan las pruebas realizadas al algoritmo propuestas por la metodología de desarrollo escogida, XP. Además, se validan los resultados aplicándoles un conjunto de métricas.

3.2. Fase de Implementación

Dentro del proceso de desarrollo de la metodología del desarrollo del software la parte más importante es la implementación, esta es la que genera los resultados reales. En esta fase se descomponen las HU en tareas de programación o ingeniería, que a su vez son convertidas en código. Además, en esta fase se especifica la implementación de las HU en su correspondiente iteración, obteniéndose en cada una de ellas una versión funcional del producto. (44)

Iteración 1

En esta iteración se implementa la funcionalidad que permiten cargar las imágenes en formato jpg o png del ordenador o cualquier almacenamiento disponible y los algoritmos de interpolación bilineal y bicúbica.

Iteración 2

En esta iteración se implementan los algoritmos de interpolación mediante Splines y mediante el vecino más cercano.

Iteración 3

En esta iteración se implementan los algoritmos de segmentación por Umbralización de Otsu y mediante Transformada circular de Hough.

Iteración 4

En esta iteración se implementa el algoritmo de segmentación por crecimiento de regiones y la funcionalidad que permite guardar las imágenes ya procesadas.

3.3. Interfaz del Sistema

A continuación se muestra la interfaz del sistema, que permite de manera fácil la interacción con el software por parte del cliente.

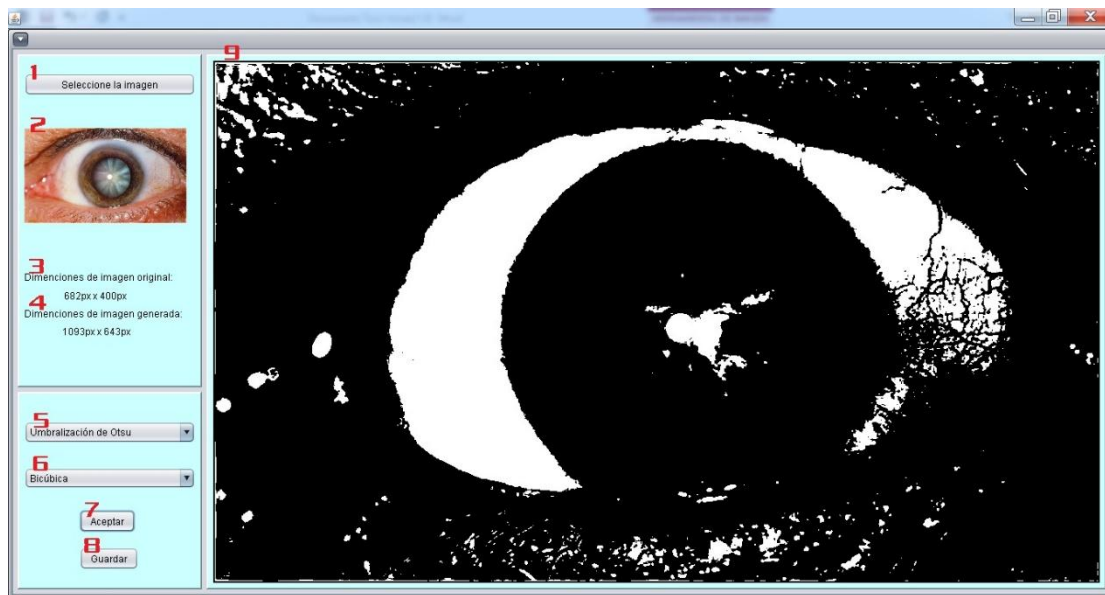


ILUSTRACIÓN 14 . INTERFAZ DEL SOFTWARE

- 1: Permite seleccionar y cargar la imagen deseada.
- 2: Vista previa de la imagen original seleccionada.
- 3: Resolución de imagen original.
- 4: Resolución de imagen obtenida del procesamiento.
- 5: Menú de selección del método de segmentación deseado.
- 6: Menú de selección del método de segmentación deseado.
- 7: Botón de aceptación de la elección.
- 8: Botón para guardar la imagen procesada en caso de que se desee.
- 9: Vista de la imagen resultante.

3.4. Pruebas

Para garantizar la calidad de lo implementado, se avanzó hacia la fase de pruebas según propone la metodología en uso. La Metodología XP divide las pruebas unitarias en dos grupos: pruebas de caja blanca y pruebas de caja negra. Las primeras son desarrolladas por los programadores y se encargan de verificar el código automáticamente y las otras están destinadas a verificar que al final de cada iteración las historias de usuario cumplan con la funcionalidad asignada y satisfagan las necesidades del cliente (36).

3.4.1. Pruebas unitarias

Las pruebas unitarias son una forma de comprobar el correcto funcionamiento de un módulo de código. La idea es escribir casos de prueba para cada función no trivial o método, de forma que cada caso sea independiente del resto (51). Como parte de las pruebas unitarias se han aplicado al algoritmo las técnicas de caja blanca, y además las técnicas de caja negra para comprobar las funcionalidades del software.

3.5.1.1. Pruebas de caja blanca

Las pruebas de caja blanca se centran en los detalles procedimentales del software, estas precisan el acceso al código del programa de modo que se pueda comprobar su lógica interna. Para realizar esta técnica, se eligió la técnica del camino básico. (51)

El método del camino básico permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, dicha técnica se aplicará al método `binarizar()` de la clase `Threshold`. (43)

A continuación se mostrara el fragmento de código al que se le aplica la técnica del camino básico y se le calcula la complejidad ciclomática:

```

7
8 public class Threshold {
9     int THRESHOLD = 180;
10    public static Picture binarizar(String filename, int THRESHOLD) {
11        Picture pic = new Picture(new File(filename));
12        //recorido de los pixeles de la imagen
13        for (int i = 0; i < pic.width(); i++) {
14            for (int j = 0; j < pic.height(); j++) {
15                Color color = pic.get(i, j);
16                double lum = Luminance.intensity(color);
17                if (lum >= 180) pic.set(i, j, Color.WHITE);
18                else pic.set(i, j, Color.BLACK);
19            }
20        }
21        System.out.println("pincha");
22        return pic;
23    }

```

ILUSTRACIÓN 15 . CÓDIGO FUENTE DEL MÉTODO BINARIZAR()

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

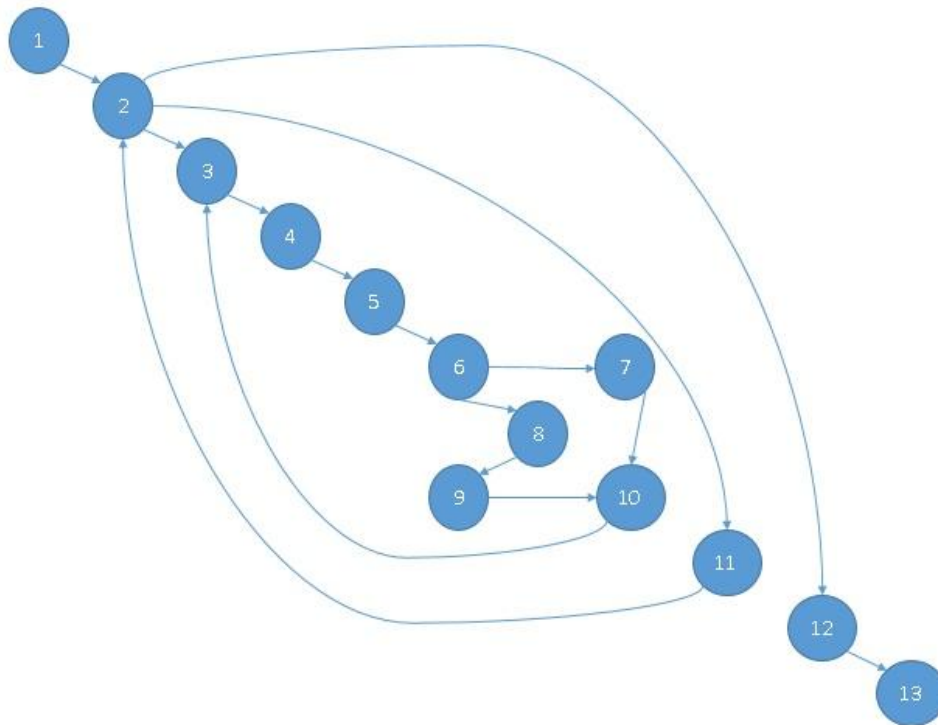


ILUSTRACIÓN 16 . GRAFO DE FLUJO DEL MÉTODO BINARIZAR()

Sobre la base de este grafo, podemos definir la complejidad ciclomática $V(G)$ de la siguiente manera:

$$V(G) = A - N + 2C$$

$$V(G) = 15 - 13 + 2$$

$$V(G) = 4$$

En esta fórmula $V(G)$ es la complejidad ciclomática, A es el número de aristas del grafo, N es el número de nodos en el grafo y C es el número de componentes conectados. Debido a que esta es una función simple con un solo punto de entrada y salida, el número de componentes es 1 y la fórmula se reduce a $V(G) = A - N + 2$.

Otra alternativa para calcular la complejidad ciclomática es:

$$V(G) = R + 1$$

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

$$V(G) = 4$$

Donde R es la cantidad de regiones cerradas en el grafo

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 4 caminos:

Camino básico 1: $1 > 2 > 12 > 13$

Camino básico 2: $1 > 2 > 3 > 4 > 11 > 2 > 12 > 13$

Camino básico 3: $1 > 2 > 3 > 4 > 5 > 6 > 8 > 9 > 10 > 3 > 11 > 2 > 12 > 13$

Camino básico 4: $1 > 2 > 3 > 4 > 5 > 6 > 7 > 10 > 3 > 11 > 2 > 12 > 13$

Los casos de pruebas se definen a partir de los caminos básicos detectados, de forma que los datos señalados causen que se visiten las sentencias vinculadas a cada nodo del camino. Para este método se calcularon cuatro caminos básicos, por lo que se deben hacer igual número de casos de prueba, para aplicar las pruebas.

TABLA 7 . CASO DE PRUEBA PARA EL CAMINO BÁSICO 4

Caso de prueba: Camino básico 4	
Entrada	Imagen con una amplia gama de colores.
Resultados Esperados	Se devuelve una imagen donde solo existen 2 colores, el blanco y el negro.
Condiciones	Se selecciona la imagen interpolada

3.5.1.2. Pruebas de caja negra

Las pruebas de caja negra cumplen con el objetivo de indicarnos cuando las funcionalidades de una iteración han sido completadas exitosamente, estas se realizan para establecer el grado de confianza en un sistema, partes del mismo o en sus características funcionales (40).

Como parte de estas pruebas se procedió a la creación de 12 casos de estudio para verificar los resultados del algoritmo, en cada caso se tomaron imágenes que corresponden a diferentes pacientes (1, 2, 3..., 12). A continuación, se presentan 1 caso de estudio de los 12 realizados donde se muestra la diferencia de calidad de imagen entre la imagen original y la imagen interpolada, para un mayor nivel de detalle

se ha realizado un zoom a ambas imágenes (Ilustración 17 e Ilustración 18) en el área señalada en la Ilustración 16. Las pruebas arrojaron los siguientes resultados:

Caso de estudio 1:

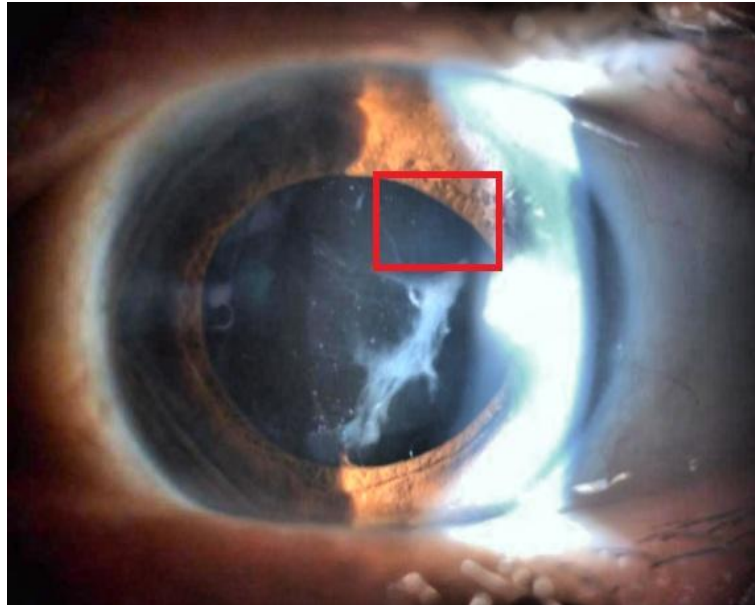


ILUSTRACIÓN 17 . ÁREA DEL ZOOM EN IMAGEN ORIGINAL. RESOLUCIÓN: 573 x 430

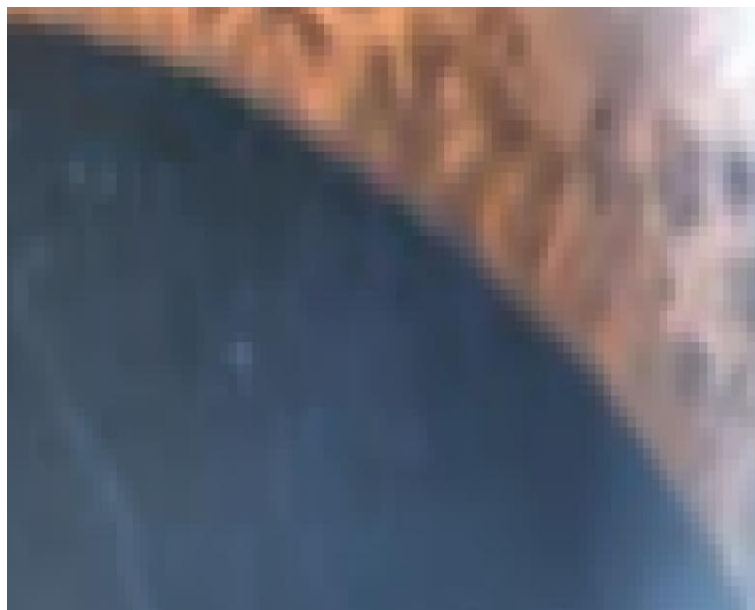


ILUSTRACIÓN 18 . ZOOM DE LA IMAGEN ORIGINAL



ILUSTRACIÓN 19 . ZOOM DE IMAGEN INTERPOLADA. RESOLUCIÓN: 1093 x 643

Como se puede ver en los resultados obtenidos del ejemplo al realizar un zoom a la imagen original esta comienza a perder calidad rápidamente mientras que la imagen interpolada cuando se le realiza un zoom exactamente igual al de la imagen original no pierde calidad tan rápidamente.

Para la descripción de estas pruebas se presenta el diseño del caso de prueba correspondiente a la historia de usuario “Interpolación bicúbica”, la cual es una funcionalidad importante para el sistema. La siguiente tabla muestra el resultado satisfactorio obtenido en dicha prueba.

TABLA 8 . CASO DE PRUEBA PARA LA HU 3

Caso de prueba 3
Código: HU_3 Historia de Usuario: 3
Nombre: Aplicar a la imagen el método de Interpolación Bicúbica.
Descripción: Redimensiona la imagen original aumentándole la resolución mediante una interpolación bicúbica.
Condiciones de ejecución: Se debe cargar la imagen original en formato jpg o png.

Entrada/ Pasos de Ejecución:

- El usuario carga la imagen en formatos jpg o png al sistema.
- El usuario selecciona que desea aplicar “Interpolación bicúbica” a dicha imagen.
- El sistema aumenta las dimensiones y resolución de la imagen utilizando la interpolación bicúbica.
- El sistema muestra una vista previa de la imagen resultante, solo interpolada.

Resultado esperado: El resultado es una imagen con más resolución que la imagen original.

Evaluación de la prueba: Prueba satisfactoria.

3.6. Validación de la solución

Las métricas dan una medida que cuan correcto y que calidad presenta el resultado obtenido. A continuación se trata la validación mediante la detección de bordes y se muestran los resultados obtenidos.

3.6.1. Validación por detección de bordes

La detección de borde es un elemento importante en el procesamiento de imágenes, su función es identificar los límites de regiones homogéneas en una imagen basado en sus propiedades como son la intensidad y la textura. Se ha decidido utilizar como métrica de calidad un algoritmo de detección de bordes ya que al tener la cantidad de pixeles que forman parte de los bordes de la imagen podemos identificar el nivel de detalle de la imagen. Por tanto al comparar la cantidad de bordes de la imagen original con la cantidad de bordes de la imagen preprocesada tenemos la comparación numérica en cuanto a calidad de imagen.

Una vez seleccionado el algoritmo se procedió a identificar los bordes de la imagen original y la imagen preprocesada, a continuación la primera imagen muestra la detección de bordes de la imagen original y la segunda muestra la detección de bordes de la imagen interpolada:

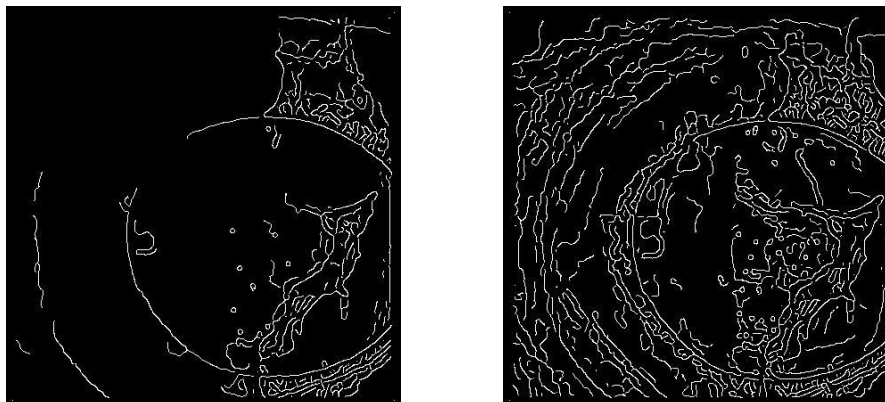


ILUSTRACIÓN 20 . COMPARACIÓN DEL ALGORITMO DE DETECCIÓN DE BORDES

En estos resultados se puede observar la calidad de información que contiene la imagen preprocesada, puesto que es más definida y nítida. Se realizaron dos comparaciones con estos resultados, primero se calculó la cantidad de píxeles que están contenidos en cada borde de estas imágenes y luego se calculó el porcentaje que esto representa del total de píxeles que tienen dichas imágenes, a continuación en la Tabla 10 se muestran los valores de bordes promedios que logró alcanzar una con respecto a la otra:

TABLA 9 . VALORES DE COMPARACIÓN DE DETECCIÓN DE BORDES

Imágenes	Cantidad de píxeles en borde	Cantidad de píxeles en borde (%)
Imagen original	3971	5.6
Imagen interpolada	25419	9.2

3.7. Conclusiones Parciales

- Se han aplicado pruebas de caja blanca y de caja negra las cuales mostraron resultados satisfactorios, lo cual permitió verificar que el software es funcional y cumple las necesidades del cliente.
- Se realizó la validación del sistema mediante un algoritmo de detección de bordes, se obtuvo un promedio de píxeles en bordes de 3971 en la imagen original y 25419 en la imagen interpolada que representan el 5.6% y 9.2 % respectivamente, lo cual permitió demostrar que el algoritmo logra mejorar la

calidad de detalle de la imagen por tanto cumple con el objetivo de la investigación.

Conclusiones generales

- El estudio de la técnica utilizada para la identificación de la OCP en el software PANDOC demostró que la misma no proporciona una respuesta eficiente al objetivo general de la presente investigación, fomentando la necesidad de desarrollar el algoritmo propuesto.
- Se determinaron los requisitos funcionales y no funcionales del sistema, se diseñaron las historias de usuario, el plan de entrega, patrones de diseño y estándares de codificación, lo cual permitió dirigir la investigación como propone la metodología en uso y determinar las funcionalidades básicas a desarrollar durante el proceso.
- La aplicación de técnicas para validar los requisitos, métricas para validar el diseño y la realización de pruebas unitarias, permitió verificar que el algoritmo es funcional y cumple las necesidades del cliente.

Referencias

1. al., Birren et. Birren Encyclopedia of Gerontology V1. s.l. : Academic Press, 1996. ISBN 0-12-226861-X..
2. Boyd, Kierstan. American Academic of Ophthalmology. What Are Cataracts? [Online] julio 11, 2014. [Cited: noviembre 22, 2017.] <https://www.aao.org/eye-health/diseases/what-are-cataracts>.
3. Forrester, John V. Elsevier. The Eye, 4th Edition: basic sciences in practice. [Online] 2016. [Cited: noviembre 22, 2018.] Forrester et al. (2002). <https://www.elsevier.ca/ISBN/9780702055546/The-Eye>. ISBN/9780702055546.
4. Estrategias de prevención de la opacidad de la cápsula posterior. Iván Hernández López, Juan Raúl Hernández Silva, Yadira Castro González, Ailén Garcés Fernández, Zucell Veitía Roviroso, Eneida Pérez Candelaria. Ciudad de la Habana : Revista Cubana de Oftalmología, 2010, Vol. Vol. 23.
5. Opacidad de la cápsula posterior. Alejandro Flores Gaitán, María Elena Morales Gómez, Humberto Matiz, Marisol Garzón. Mexico : Rev Mex Oftalmol;, 2005.
6. López, Iván Hernández, et al. Estrategias de prevención de la opacidad de la cápsula posterior. La Habana : s.n.
7. Opacidad de la cápsula posterior. Criterios y tendencias actuales. Hernández López I, Castro González Y, Trujillo Fonseca K, Cepero Hernández L, Miranda Hernández. La Habana : s.n., 2009. p. 307-17.
8. "Quantification of posterior capsular opacification in digital images after cataract surgery". SA., Barman. s. n., 2000, Vol. Vol.3979.
9. O, Findl and W., Buehl. "Comparison of 4 methods for quantifying posterior capsule opacification". Journal of Cataract & Refractive Surgery. 2003, Vol. Vol 29.
10. A, Tariq, N, Patton and B., Dhillon. "Assessment of systems of analyzing PCO". Journal of Cataract & Refractive Surgery. 2005, Vol. Vol.31.
11. DS, Friedman and DD., Duncan. "Digital image capture and automated analysis of posterior capsular opacification". Investigative Ophthalmology and Visual Science. 2010, Vol. Vol 40.

12. Ruiz, R. Historia y Evolución del Pensamiento Científico. Historia y Evolución del Pensamiento Científico. Mexico : s.n., 2006.
13. Rey, Georges. The Analytic/Synthetic Distinction. Stanford Encyclopedia of Philosophy. [Online] Winter 2009 Edition, Agosto Viernes 15, 2008. [Cited: 12 12, 2017.] <https://plato.stanford.edu/archives/win2009/entries/analytic-synthetic/>.
14. Iudin, M. Rosental y P. Diccionario filosófico marxista. Montevideo : Ediciones Pueblos Unidos, 1984.
15. Jack J. Kanski. Kanski, Jack J. 2004, Vol. 5ª edición. ISBN 978-84-8174-758-4.
16. Gubern, R. Patología de la imagen. Barcelona : Anagrama., 2004.
17. Alegre, E., Sánchez, L., Fernández, R. Á., Mostaza, J. C. Procesamiento Digital de Imagen: fundamentos y prácticas con Matlab. . Universidad de León. : s.n., 2003. ISBN 84-9773-052-6..
18. Acharya, T., Ray, A. K. Image processing: principles and applications. s.l. : John Wiley & Sons., 2005.
19. Efraín García, Rubén Osuna. Foto Punto. Fundamentos de la imagen fotográfica digital. [Online] <http://www.uned.es/personal/rosuna/resources/photography/ImageQuality/fundament os.imagen.digital.pdf>.
20. Anandan, M. Organic: LED Backlight: Enhancement of picture quality donde LCD screen. Lighting Technologies LLC Austin. Texas, USA. : s.n.
21. Otero, Roger José Bello. Analisis Morfológico de imagen Multicanal. UNIVERSIDAD CENTRAL DE VENEZUELA. Caracas : s.n., 2012. Tesis.
22. Diccionario de fotografía y diseño, Escala de grises (grayscale). FotoNostra. [Online] [Cited: Abril 3, 2018.] <http://www.fotonostra.com/glosario/escalagrisegrayscale.htm>.
23. Segmentación de imágenes de color. J.J. Báez Rojasa, M.L. Guerrerob, J. Conde Acevedoc, A. Padilla Vivancoa y G. Urcid Serranoa. no. 6, Mexico : Rev. mex. fis., 2004, Vol. vol.50. ISSN 0035-001X.
24. Obregon, Ing. Luis Jara. Slide Share. Tecnicas de segmentación de la imagen. [Online] abril 11, 2017. [Cited: mayo 28, 2018.]

<https://es.slideshare.net/stalinheavy/tcnicas-de-segmentacin-de-imgenes>.

25. Segmentacion II. Segmentacion II. [Online] [Cited: mayo 28, 2018.] http://asignatura.us.es/imagendigital/Tema5-2_SegmentacionRegionesUmbralizacion.pdf.

26. P., Carlos. Apuntes de Visión Artificial. Segmentación. [Online] [Cited: mayo 28, 2018.] <https://isa.umh.es/asignaturas/rvc/cap5VASegmentacion.pdf>.

27. Thresholding Using an Illumination Model. Parker, J. R. s.n., 1991, Canada : s.n., 1991, Vol. Vol. 13.

28. JUAN PABLO URREA, EMMANUEL OSPINA. IMPLEMENTACIÓN DE LA TRANSFORMADA DE HOUGH PARA LA DETECCIÓN. 2004.

29. Segmentacion de Imágenes. Segmentacion de Imágenes. [Online] [Cited: mayo 28, 2018.] <http://alojamientos.us.es/gtocoma/pid/tema4.pdf>.

30. Segmentacion de imagenes. Segmentacion de imagenes. [Online] [Cited: mayo 28, 2018.] http://www.lcc.uma.es/~munozp/documentos/procesamiento_de_imagenes/temas/pi_cap6.pdf.

31. La Interpolacion de la imagen. [Online] [Cited: enero 16, 2018.] <http://www.fernandogago.es/interpolacion.html>.

32. Giménez, Nardo. Complejidad de algoritmos para la Interpolación polinomial en varias variables. Buenos Aires : Universidad de Buenos Aires, 2007.

33. Melo, Samuel Barreto. Transformaciones geométricas sobre imágenes digitales. Transformaciones geométricas sobre imágenes digitales. [Online] [Cited: mayo 28, 2018.] <http://alammi.info/2congreso/memorias/Documentos/martes/TRANSFORMGEOMETRICAS.pdf>.

34. Glenda Marcela H. R., Evelin Marleny L. L. Métodos de interpolación multivariable. Métodos de interpolación multivariable. [Online] [Cited: mayo 28, 2018.] <http://ri.ues.edu.sv/10295/1/19201028.pdf>.

35. Grau, Juan Luis Vila. Management Plaza. ¿Sabes cómo funciona XP? [Online] 2018. [Cited: Marzo 13, 2018.] <http://managementplaza.es/blog/sabes-como-funciona-xp/>.

36. Joskowicz, Ing. José. Reglas y Prácticas en eXtreme Programing. España : s.n., 2008.
37. Lutz, Mark. Learning Python, Fourth Edition. s.l. : O'Reilly Media, Inc., 2010.
38. Byous, Jon. Java technology: The early years. Sun Developer Network. 2005.
39. Suaza, K. V., García, J. J. T., & Jaramillo, C. M. Z. Mejora de historias de usuario y casos de prueba de metodologías ágiles con base en TDD. 2015.
40. Fernández, Oscar Belmonte. Introducción al lenguaje de JAVA.Una guia básica. 2004.
41. Domínguez-Dorado, M. Netbeans IDE.La alternativa a Eclipse. Todo Programación. Nº 13. Madrid : Editorial Iberprensa, 2005, pp. Págs. 32-34.
42. Montes, Maria. Herramientas Case.El mejor soporte para el proceso de desarrollo de software. INEI. academia. [Online] [Cited: Marzo 21, 2018.] http://www.academia.edu/4513393/Libro_HERRAMIENTAS_CASE..
43. Pressman, Roger S. Ingenieria de software. Un enfoque práctico. México : Editorial Mexicana, 2010. ISBN: 978-607-15-0314-5.
44. Beck, Kent. Extreme Programming Explained. 1999.
45. Calabria, Luis y Píriz, Pablo. Metodología XP. Uruguay : Universidad ORT, 2003.
46. García, Oscar. Nomenclatura estándar del código Java. [Online] Agosto 22, 2011. [http://www.elclubdelprogramador.com/2011/08/22/java-nomenclatura-estandar-del-codigo-java/..](http://www.elclubdelprogramador.com/2011/08/22/java-nomenclatura-estandar-del-codigo-java/)
47. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Elements of Reusable Object-Oriented Software.
48. Mühlrad, Daniel. Patrones de diseño. 2008.
49. Larman, Craig. UML y Patrones 2da Edición. 2012.
50. Sommerville, Ian. Ingeniería de Software. Madrid, España : Pearson Educación. S. A., 2005. 84-7829-074-5..
51. Barrientos, Pablo Andrés. Enfoque para pruebas de unidad basado en la generación aleatoria de objetos. 2014. p. p. 101.

-
52. PMOInformatica. [Online] [Cited: mayo 27, 2018.] <http://www.pmoinformatica.com/2016/08/pruebas-aceptacion-software-istqb.html>.
53. GALINDO, Cáceres Jesús. Técnicas de Investigación en Sociedad, Cultura y Comunicación. 1ª. Edición. Mexico : Addison Wesley Longman., 1998. ISBN 968-444-262-9.
54. SIERRA, Bravo. R. Técnicas de investigación Social. 9ª. Edición. Madrid : Editorial Paraninfo S. A., 1994. ISBN 84-283-1548-5.
55. Terrera, Gustavo. Testing Baires. Pruebas de Caja Negra y un enfoque práctico. [Online] Febrero 26, 2017. [Cited: Marzo 22, 2018.] <https://testingbaires.com/pruebas-caja-negra-enfoque-practico/>.
56. PMOInformatica. Pruebas de caja negra ISTQB. [Online] Abril 13, 2016. [Cited: Marzo 22, 2018.] <http://www.pmoinformatica.com/2016/04/pruebas-caja-negra-istqb.html>.

Anexos

TABLA 10 . HU: APLICAR A LA IMAGEN EL MÉTODO DE INTERPOLACIÓN BICÚBICO

Historia de Usuario	
Número: 3	Nombre de la Historia de Usuario: Aplicar a la imagen el método de interpolación bicúbico.
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Alto	Puntos reales: 3 semanas
Programador responsable: Ismael Montano Muñoz	
Descripción: se interpola la imagen utilizando la interpolación bicúbica.	
Procesar imagen: se obtiene una imagen Interpolada mediante el método bicúbico.	

TABLA 11 . HU: APLICAR A LA IMAGEN EL MÉTODO DE INTERPOLACIÓN MEDIANTE SPLINES

Historia de Usuario	
Número: 4	Nombre de la Historia de Usuario: Aplicar a la imagen el método de interpolación mediante Splines.

Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 2
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Alto	Puntos reales: 3 semanas
Programador responsable: Ismael Montano Muñoz	
<p>Descripción: se interpola la imagen utilizando la interpolación de Splines.</p> <p>Procesar imagen: se obtiene una imagen Interpolada mediante el método de Splines.</p>	

TABLA 12 . HU: APLICAR A LA IMAGEN EL MÉTODO DE INTERPOLACIÓN MEDIANTE EL VECINO MÁS CERCANO

Historia de Usuario	
Número: 5	Nombre de la Historia de Usuario: Aplicar a la imagen el método de interpolación mediante el vecino más cercano.
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 2
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Alto	Puntos reales: 3 semanas

Programador responsable: Ismael Montano Muñoz
Descripción: se interpola la imagen utilizando la interpolación del vecino más cercano.
Procesar imagen: se obtiene una imagen Interpolada mediante el método del vecino más cercano.

TABLA 13 . HU: SEGMENTAR IMAGEN MEDIANTE EL MÉTODO DE UMBRALIZACIÓN DE OTSU

Historia de Usuario	
Número: 6	Nombre de la Historia de Usuario: Segmentar imagen mediante el método de Umbralización de Otsu.
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 3
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Programador responsable: Ismael Montano Muñoz	
Descripción: obtiene la imagen ya interpolada y procede a segmentarla.	
Procesar imagen: se obtiene una imagen segmentada mediante el método Umbralización de Otsu.	

TABLA 14 . HU: SEGMENTAR LA IMAGEN MEDIANTE EL MÉTODO DE TRANSFORMADA CIRCULAR DE HOUGH

Historia de Usuario

Número: 7	Nombre de la Historia de Usuario: Segmentar imagen mediante el método de Transformada circular de Hough.	
Modificación a la Historia de Usuario: Ninguna		
Usuario: Michel Álvarez Cancio		Iteración asignada: 3
Prioridad en negocio: Alta		Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto		Puntos reales: 2 semanas
Programador responsable: Ismael Montano Muñoz		
<p>Descripción: obtiene la imagen ya interpolada y procede a segmentarla.</p> <p>Procesar imagen: se obtiene una imagen segmentada mediante el método de Transformada circular de Hough.</p>		

TABLA 15 . HU: SEGMENTAR IMAGEN MEDIANTE EL MÉTODO DE CRECIMIENTO DE REGIONES

Historia de Usuario		
Número: 8	Nombre de la Historia de Usuario: Segmentar imagen mediante el método de Crecimiento de regiones.	
Modificación a la Historia de Usuario: Ninguna		
Usuario: Michel Álvarez Cancio		Iteración asignada: 3

Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Programador responsable: Ismael Montano Muñoz	
<p>Descripción: obtiene la imagen ya interpolada y procede a segmentarla.</p> <p>Procesar imagen: se obtiene una imagen segmentada mediante el método de Crecimiento de regiones.</p>	

TABLA 16 . HU: GUARDAR IMAGEN

Historia de Usuario	
Número: 10	Nombre de la Historia de Usuario: Guardar imagen.
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 4
Prioridad en negocio: Bajo	Puntos estimados: 1 semanas
Riesgo en desarrollo: Bajo	Puntos reales: 1 semanas
Programador responsable: Ismael Montano Muñoz	
<p>Descripción: el usuario puede guardar la imagen resultante en cualquier directorio.</p> <p>Guardar imagen: guarda la imagen resultante.</p>	



Análisis de los efectos de la interpolación en la segmentación de imágenes de opacidad capsular

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

Autores:

Ismael Montano Muñoz

Tutores

Ing. Michel Álvarez Cancio

La Habana, del 2018