

Universidad Central “Marta Abreu” de Las Villas
Facultad de Matemática-Física-Computación
Licenciatura en Ciencia de la Computación



Trabajo de Diploma

**Software para la caracterización de aceros al carbono mediante el
procesamiento digital de imágenes metalográficas**

Autor:

Manuel Alejandro Castro Fuentes

Tutores:

Lic. Marcelino Rodríguez Cancio

MSc. Damny Magdaleno Guevara

Dr. Alejandro Duffus Scott

“Año 54 de la Revolución”

Santa Clara, 2012



Declaración Jurada

El que suscribe, _____, hago constar que el trabajo titulado _____ fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de _____, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del Laboratorio

Fecha

*Ah, pero los límites de un hombre deben exceder su capacidad.
¿O para qué existe el cielo?
Robert Browning*

A mi familia por todos los sacrificios hechos a lo largo de mi carrera como estudiante.

A mis muy buenos viejos amigos, y a mis buenos amigos nuevos.

Agradecimientos

Al colectivo Blizzard por cultivar siempre con su trabajo mi amor por computación.

Resumen

El análisis microestructural de aceros de bajo y mediano contenido de carbono permite estimar el porcentaje de fases presentes, contenido de carbono, tamaño de grano y algunas propiedades mecánicas fundamentales. Por lo general, los análisis cuantitativos de las propiedades de los aceros a partir del estudio micro estructural usando un microscopio óptico metalográfico, se realizan directamente sobre la microestructura y sus resultados dependen en gran medida de la habilidad y experiencia del técnico especialista. Por tal motivo en este trabajo se desarrolló *AceroPro*, un sistema para el procesamiento digital de imágenes obtenidas a través de un microscopio óptico metalográfico, con la finalidad de caracterizar metalográficamente aceros al carbono. Se utilizó para esto varias herramientas para el desarrollo de aplicaciones de procesamiento digital de imágenes. Los resultados obtenidos por la aplicación fueron evaluados con técnicas estadísticas, obteniéndose buenos resultados.

Abstract

Microstructural analysis of steels of low and medium carbon content allows estimating the percentage of phases present, carbon content, grain size and some basic mechanical properties. Generally, quantitative analyzes of the properties of the steels from the study of the microstructure using a metallographic microscope, are performed directly on the microstructure and their results are highly dependent on the skill and experience of the technician. Therefore in this paper AceroPro was developed, a system for digital processing of images obtained through a metallographic optical microscope, in order to characterize carbon steels metallographically. For this we used several tools for the development of digital image processing applications. The results were evaluated by applying statistical techniques, obtaining good results.

Tabla de Contenidos

INTRODUCCIÓN	1
CAPÍTULO 1. PROCESAMIENTO DIGITAL DE IMÁGENES Y SUS APLICACIONES.....	4
<i>1.1 Procesamiento digital de imágenes: una herramienta para la ciencia y la técnica</i>	<i>4</i>
1.1.1 Jerarquía de las operaciones del procesamiento de imágenes	5
1.1.2 Componentes de un sistema de procesamiento de imágenes	6
<i>1.2 Técnicas de procesamiento digital de imágenes.....</i>	<i>6</i>
1.2.1 Pre-procesamiento.....	7
1.2.2 Segmentación	8
<i>1.3 Software para el procesamiento digital de imágenes</i>	<i>10</i>
1.3.1 Aplicaciones disponibles para el procesamiento de imágenes	10
1.3.2. Herramientas para el desarrollo de aplicaciones de procesamiento de imágenes	12
<i>1.4 Metalografía cuantitativa</i>	<i>14</i>
1.4.1 Metalografía cuantitativa convencional	14
1.4.2 Software para el procesamiento de imágenes aplicado a la metalografía.	15
<i>1.5 Conclusiones parciales</i>	<i>16</i>
CAPÍTULO 2. CARACTERIZACIÓN DE ACEROS MEDIANTE EL PROCESAMIENTO DE IMÁGENES METALOGRÁFICAS.....	19
2.1 Características de las imágenes metalográficas	19
2.2 Procesamiento y análisis	20
2.3 Concepción de la aplicación.....	21
2.3.1 Lenguaje de programación Python.....	23
2.3.2 Marco de trabajo Qt	24
2.3.3 Paquete de herramientas ITK.....	26
2.3.4 Paquete de herramientas VTK.....	30
2.4 Diseño de la aplicación	32
2.4.1 Visión general del funcionamiento de la aplicación.....	32
2.4.2 Diagrama de clases	34
2.5 Conclusiones parciales	36
CAPÍTULO 3. EVALUACIÓN DE LA IMPLEMENTACIÓN Y DESCRIPCIÓN A NIVEL DE USUARIO DE LA APLICACIÓN.....	38
3.1 Interfaz de usuarios AceroPro para la caracterización de aceros al carbono mediante el procesamiento digital de imágenes metalográficas	38
3.1.1 Ventana principal de la aplicación.....	38
3.1.2 Interfaz de procesamiento automático	40

3.1.3 Interfaz del segmentador Otsu	40
3.1.4 Interfaz del filtro de erosión	41
3.1.5 Interfaz del filtrado bilateral	42
3.1.6 Interfaz del filtro de rellenado de huecos	42
3.1.7 Interfaz de análisis	43
3.2 Evaluación de la aplicación	45
3.2.1 Definición del caso de estudio para la aplicación del procesamiento digital de imágenes a través de AceroPro	45
3.2.2 Diseño y aplicación de los experimentos	45
3.3 Conclusiones parciales	48
CONCLUSIONES	49
RECOMENDACIONES	50
REFERENCIAS BIBLIOGRÁFICAS.....	51

INTRODUCCIÓN

El análisis micro estructural de aceros de bajo y mediano contenido de carbono permite estimar el porcentaje de fases presentes, contenido de carbono, tamaño de grano y algunas propiedades mecánicas fundamentales. Por lo general, los análisis cuantitativos de las propiedades de los aceros a partir del estudio micro estructural usando un microscopio óptico metalográfico, se realizan directamente sobre la microestructura y sus resultados dependen en gran medida de la habilidad y experiencia del técnico especialista. Las herramientas profesionales existentes para el tratamiento de imágenes digitales en el campo de la ciencia de los materiales son pocas y muy costosas. De aquí se deriva el **problema científico** a resolver que se manifiesta en la carencia de un software especializado para el análisis estructural de muestras de aceros a partir del cálculo del porcentaje de fases presentes, mediante el procesamiento digital de imágenes obtenidas a través de microscopía óptica, esto trae como consecuencia:

- Alta dependencia de expertos.
- La necesidad de aplicar ensayos destructivos.
- Uso de equipos y máquinas costosas.

Se pretende por tanto, desarrollar un programa especializado, que permita a partir de una imagen digital de la microestructura de los aceros, estimar las características del material; evitando así la necesidad de poseer un laboratorio con gran inversión.

Para contribuir a la solución del problema científico antes plateado, se formuló la **hipótesis general de investigación** siguiente:

La implementación de un software para el procesamiento digital de imágenes obtenidas a través de un microscopio óptico metalográfico, permitiría determinar el contenido de fases en muestras de aceros, lo que facilitaría la estimación de propiedades mecánicas, sin necesidad del uso de las máquinas tradicionales utilizadas para los ensayos mecánicos.

En conformidad con la hipótesis de investigación identificada, el **objetivo general** de la investigación consiste en: Desarrollar un software para el procesamiento digital de

imágenes obtenidas a través de un microscopio óptico metalográfico, con la finalidad de caracterizar metalográficamente aceros al carbono mediante la aplicación de técnicas de metalografía cuantitativa.

Este objetivo general fue desglosado en los **objetivos específicos** siguientes:

1. Realizar un estudio bibliográfico de las herramientas existentes que permiten el procesamiento de imágenes, así como de las técnicas de metalografía convencional.
2. Diseñar e implementar, haciendo uso de las herramientas seleccionadas, una aplicación para la caracterización de imágenes de aceros al carbono.
3. Evaluar los resultados obtenidos mediante la comparación de estos con otra herramienta de procesamiento de imágenes y con los datos de las normas de las muestras analizadas.

Para la presentación de esta investigación, este Trabajo de Diploma se estructuró de la forma siguiente. En el Capítulo 1 se realiza un análisis del estado del arte sobre el procesamiento digital de imágenes y sus aplicaciones, específicamente la metalografía. El Capítulo 2 resume y explica aspectos relacionado con las herramientas computacionales utilizadas para el desarrollo de la aplicación, así como el proceso de implementación de la misma. En el Capítulo 3, se describe a nivel de usuario el software implementado, donde se muestran los casos de aplicación; además se realiza un estudio estadístico con los resultados obtenidos, como vía para demostrar la hipótesis de investigación planteada. Finalmente, un cuerpo de Conclusiones y Recomendaciones derivadas de la investigación realizada y la Bibliografía consultada.

Procesamiento digital de imágenes y sus aplicaciones

CAPÍTULO 1. PROCESAMIENTO DIGITAL DE IMÁGENES Y SUS APLICACIONES

Desde los inicios de la ciencia, la observación visual ha jugado un papel importante. En ese momento, la única manera de documentar los resultados de un experimento era por descripción verbal y dibujos manuales. El siguiente paso importante fue la invención de la fotografía que permitió documentar objetivamente los resultados. Debido a que los procedimientos de evaluación manual consumían mucho tiempo, la evaluación cuantitativa de las imágenes no encuentra aplicación generalizada en ese momento. En general, las imágenes sólo fueron utilizadas para la documentación, descripción cualitativa, y la ilustración de los fenómenos observados. (Jähne, 2005)

Hoy en día, se está en medio de una segunda revolución provocada por el rápido progreso en las tecnologías de la computación. Las computadoras personales se han tornado lo suficientemente potentes para el procesamiento de imágenes. Como resultado, el software y hardware multimedia se está convirtiendo en estándar para la manipulación de imágenes, secuencias de imágenes, e incluso de visualización 3-D. La tecnología ya está disponible para cualquier científico o ingeniero. En consecuencia, el procesamiento de imágenes se ha expandido y se está ampliando con rapidez de unas pocas aplicaciones especializadas a una herramienta científica estándar. En la actualidad, las técnicas de procesamiento de imágenes se aplican prácticamente a todas las ciencias naturales y disciplinas técnicas, ya que permiten investigar fenómenos complejos a los que no se puede acceder adecuadamente mediante técnicas de medición convencionales. (Jähne, 2005)

1.1 Procesamiento digital de imágenes: una herramienta para la ciencia y la técnica

El procesamiento digital de imágenes ha adquirido en años recientes un papel importante en las tecnologías de la información y el cómputo. Al conjunto de técnicas y procesos para descubrir o hacer resaltar la información contenida en una imagen usando como herramienta principal una computadora se le conoce como procesamiento digital de

imágenes (PDI). El interés en el estudio del procesamiento digital de imágenes se basa en dos áreas de aplicación primordiales: a) El mejoramiento de la calidad de la información contenida en una imagen con el fin de que esta información pueda ser interpretada por los humanos, y b) El procesamiento de los datos contenidos en un escenario a través de una máquina de percepción autónoma. (Rojas et al., 2011)

En la actualidad, el procesamiento digital es de gran importancia en diversas disciplinas que necesitan captar y procesar imágenes en tiempos relativamente pequeños. De tal forma, permite funciones de gran interés, entre otras, para la medicina, la ingeniería robótica, asistencia en procesos productivos, ingeniería mecánica, ingeniería de materiales y la soldadura. Dentro de estas últimas, la caracterización microestructural y macroestructural de materiales puede mejorarse comparados con las obtenidas con los métodos tradicionales que aún son usados. (Rojas et al., 2011)

1.1.1 Jerarquía de las operaciones del procesamiento de imágenes

En el procesamiento de imágenes se pueden distinguir varios pasos, los que se realizan uno tras otro para hacer posible la extracción de datos de interés de la escena observada. El procesamiento digital de imágenes comienza con la captura de una imagen, luego esta se convierte a una forma que pueda ser tratada con computadoras digitales. Este proceso se conoce como digitalización. (Jähne, 2005)

El primer paso del procesamiento digital puede incluir diferentes operaciones que se conocen como pre-procesamiento. Su principal función es mejorar características de la imagen tales como brillo y contraste, aplicar transformaciones de coordenadas, o realizar correcciones geométricas y radiométricas. Luego es necesario aplicar toda una cadena de pasos de procesamiento para analizar e identificar objetos, aplicando filtros adecuados para distinguir los objetos de interés de otros objetos presentes en la imagen. Las herramientas básicas para esta tarea son: promediar la imagen, la detección de bordes, y el análisis de vecindarios simples y patrones complejos conocidos como texturas. Estos objetos necesitan ser separados del fondo, lo cual significa que las regiones de rasgos y discontinuidades constantes deben ser identificadas mediante segmentación. Esta tarea es sencilla siempre

que el objeto sea distinguible del fondo mediante rasgos locales, sin embargo usualmente ese no es el caso. (Jähne, 2005)

1.1.2 Componentes de un sistema de procesamiento de imágenes

Un sistema de adquisición y procesamiento de imágenes de propósito general cuenta típicamente de cuatro componentes esenciales:

1. Un sistema de adquisición de imágenes.
2. Un dispositivo para convertir la señal eléctrica del sistema de adquisición en una imagen digital que se pueda almacenar.
3. Una computadora que provea del poder de procesamiento necesario.
4. Software de procesamiento de imágenes que proporcione herramientas para manipular y analizar las imágenes.

Gracias a los avances en la ciencia y la tecnología, el proceso de captura de imágenes y de conversión a una imagen digital a menudo se unifica en un solo dispositivo, cuyo ejemplo más conocido son las cámaras digitales. Dicho dispositivo cuenta entre sus bondades con la capacidad de acoplarse a un microscopio óptico, con lo que se pueden obtener imágenes digitales de alta resolución de la observación directa de procesos y estructuras a nivel microscópico.

1.2 Técnicas de procesamiento digital de imágenes

El procesamiento digital de imágenes comprende el uso de múltiples técnicas y algoritmos que se engloban en categorías según el propósito que persiguen en el trabajo con la imagen. Dentro de estas categorías se encuentran como las más reconocidas:

- **Pre-procesamiento**, cuyo objetivo fundamental es obtener, a partir de una imagen origen, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de la misma que posibilite efectuar operaciones del procesado sobre ella. (Acharya and Ray, 2005)

- **Segmentación**, que comprende 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. Más precisamente, la segmentación de la imagen es el proceso de asignación de una etiqueta a cada pixel de la imagen, de forma que los pixeles que compartan la misma etiqueta también tendrán ciertas características visuales similares. (Acharya and Ray, 2005)
- **Registro de la imagen**, proceso de transformación de diferentes conjuntos de datos a un sistema de coordenadas. Los datos pueden ser múltiples fotografías, datos de diferentes sensores, de diferentes épocas o de diferentes puntos de vista. Se utiliza en visión artificial, imagen médica, reconocimiento automático de objetivo y en la recopilación y análisis de imágenes y datos de los satélites. El registro es necesario para poder comparar o integrar los datos obtenidos de estas diferentes mediciones. (Brown, 1992)

1.2.1 Pre-procesamiento

En la fase de pre-procesamiento de imágenes digitales es común la aplicación de filtros para resaltar rasgos en la imagen, atenuar grandes diferencias de intensidades y lograr una estructura más homogénea de la escena. Esto se conoce también como reducción de ruido o suavizado de la imagen. La desventaja de la aplicación de filtros de suavizado es que tiende a difuminar los bordes afilados en la imagen que ayudan a distinguir los objetos que se intenta caracterizar. Aun cuando el suavizado no borra por completo esos bordes, tiende a distorsionar la fina estructura de la imagen cambiando aspectos sutiles de las estructuras en cuestión. (Ibañez et al., 2005)

Muchos esfuerzos han sido dedicados a la reducción de este efecto indeseable. Entre ellos se destaca el Filtrado Bilateral desarrollado por C. Tomasi y R. Manduchi para imágenes de escala de grises o a color. (Tomasi and Manduchi, 1998)

El Filtrado Bilateral realiza el suavizado usando vecindarios en el dominio y el rango. Píxeles que están cercanos a un píxel en el dominio de la imagen y son similares a un píxel en el rango de la imagen, son usados para calcular el valor filtrado. El Filtrado Bilateral es capaz de reducir el ruido de una imagen en un orden de magnitud y mantener sus bordes. (Tomasi and Manduchi, 1998)

Otro método de reducción de ruido muy utilizado, con capacidad para preservar bordes es, la Difusión Anisotrópica, introducida por Perona y Malik (Perona and Malik, 1990) y que está relacionado con un trabajo realizado anteriormente por Grossberg (Grossberg, 1984) que utilizó un modelo de difusión no lineal similar para modelar la visión humana.

1.2.2 Segmentación

La segmentación es una tarea desafiante, muchos métodos diferentes han sido propuestos e implementados en años recientes. A pesar de los enormes esfuerzos invertidos en este problema, no hay un único acercamiento que pueda resolver de forma general el problema de la segmentación para la gran variedad de modalidades de imágenes existentes hoy en día. Los algoritmos de segmentación más efectivos se obtienen mediante la personalización cuidadosa de una combinación de componentes. Los parámetros de estos componentes se deben ajustar a las características de la modalidad de imagen que se recibe como entrada y a los rasgos de las estructuras a segmentar. (Ibañez et al., 2005)

Algunos enfoques para la solución de este problema son:

- Segmentación por crecimiento de regiones (Region Growing). Los algoritmos de crecimiento de la región han demostrado ser un método eficaz para la segmentación de la imagen. El enfoque básico de un algoritmo de crecimiento de región es comenzar a partir de una región o semilla (típicamente uno o más píxeles) que se consideran dentro del objeto a ser segmentado. Los píxeles vecinos de esta región son evaluados para determinar si también deben ser considerados parte del objeto. Si es así, se agregan a la región y el proceso continúa mientras se añaden nuevos píxeles a la región. Los algoritmos de crecimiento de región varían en función de los criterios utilizados para decidir si un píxel debe ser incluido en la región o no, el

tipo de conectividad utilizada para determinar los vecinos, y la estrategia para visitar los píxeles vecinos. (Ibañez et al., 2005)

- Segmentación basada en cuencas (Watersheds). Este método clasifica los píxeles en regiones usando gradientes de descenso en las características de imagen y análisis de los puntos débiles a lo largo de las fronteras de la región. Extrapolando la idea, sería como la lluvia cayendo sobre la topología del paisaje y que fluye por gravedad a recogerse en las cuencas bajas. El tamaño de las cuencas crecerá con mayor cantidad de precipitaciones hasta que se derrame una en la otra, haciendo que las cuencas pequeñas se fusionen en grandes cuencas. Las regiones (cuencas hidrográficas) se forman mediante el uso de la estructura geométrica local para asociar los puntos en el dominio de la imagen con los extremos locales en alguna medida de esas características, como la curvatura o la magnitud del gradiente. Esta técnica es menos sensible a los umbrales definidos por el usuario que los métodos clásicos de crecimiento de regiones, y puede ser más adecuado para la fusión de diferentes tipos de características de conjuntos de datos diferentes. La segmentación basada en cuencas es también más flexible, dado que no produce una segmentación de la imagen única, sino más bien una jerarquía de segmentaciones de la cual se puede extraer una sola región o conjunto de regiones a priori, utilizando un umbral, o de forma interactiva, con la ayuda de una interfaz gráfica de usuario. (Yoo et al., 1991a, Yoo et al., 1991b)
- Segmentación por métodos híbridos utilizando lógica difusa. El enfoque de la segmentación híbrida integra métodos de segmentación basados en frontera y basados en regiones, que amplifican la fuerza y reducen las debilidades de ambas técnicas. La ventaja de este enfoque proviene de la combinación de los métodos de segmentación basados en regiones como la conectividad difusa y clasificación mediante un diagrama de Voronoi con la segmentación basada en fronteras usando un modelo deformable. La sinergia entre metodologías fundamentalmente diferentes

tiende a resultar en la robustez y la mayor calidad de segmentación. (Ibañez et al., 2005)

Existen múltiples algoritmos exponentes de los acercamientos al problema de la segmentación mencionado anteriormente. Algunos como el método Otsu, perteneciente al crecimiento de regiones, implementan un criterio estadístico para lograr la segmentación. El objetivo de este método es encontrar un umbral (Threshold) que clasifique la imagen en clases de tal forma que se minimice la varianza dentro de cada clase, lo que es equivalente a maximizar la varianza entre clases y a minimizar el error por mal clasificación. (Ibañez et al., 2005)

1.3 Software para el procesamiento digital de imágenes

El procesamiento digital de imágenes, por la importancia que reviste, ha sido centro de atención de múltiples grupos de desarrolladores lo que ha traído como resultado que existan en el mercado varias aplicaciones diseñadas con este fin. Algunas han sido diseñadas como aplicaciones de propósito general, y cuentan con una variedad de herramientas para el procesamiento y el análisis. Otras buscan servir a propósitos específicos y cuentan con las herramientas necesarias para el trabajo de un especialista en la materia a la cual se dedican.

1.3.1 Aplicaciones disponibles para el procesamiento de imágenes

Wolfram Mathematica ha introducido una nueva generación de herramientas de procesamiento digital de imágenes y análisis totalmente integrado con las potentes capacidades algorítmicas de *Mathematica*, con amplio y profundo soporte tanto para la programación como para un estilo interactivo y moderno de potencia industrial. La arquitectura simbólica única de *Mathematica* y el paradigma de cuadernos (notebooks) permite que las imágenes en forma visual puedan ser incluidas y manipuladas directamente, tanto de forma interactiva como en los programas, haciendo posible una nueva forma de flujo de trabajo agilizado de procesamiento de imágenes. (Wolfram, 2012)

Dentro de los rasgos principales del paquete de herramientas de procesamiento digital de imágenes de *Mathematica* están:

- Moderna plataforma extensible para procesamiento digital de imágenes.
- Una implementación eficiente de todos los algoritmos de procesamiento de imagen estándar.
- Eficiente representación de alto nivel de imágenes, abstrayéndose de la representación de datos.
- Las imágenes en forma visual se pueden utilizar directamente como entrada, y se incluyen en los programas.
- Eficientes filtros lineales y no lineales para el procesamiento espacial y dominio de la frecuencia.
- Incorpora entre otros, la detección de componente morfológico y la característica de Euler.

MATLAB viene con un paquete básico de varias funciones elementales que se pueden utilizar para cargar, guardar y llevar a cabo las funciones habituales en las imágenes. Sin embargo, a menudo es de gran utilidad para realizar operaciones más complejas en las imágenes. La caja de herramientas (*toolbox*) de procesamiento de imágenes permite manipulaciones tales como:

- La visualización directa de las imágenes en *MATLAB*.
- Conversiones de espacio de color.
- Agrupamiento de objetos y recopilación de datos.
- Filtrado de imágenes y convolución rápida.
- Análisis de Fourier en las imágenes.
- Aritmética de imágenes y operaciones morfológicas.

ImageJ es un programa de dominio público programado en *Java*; desarrollado en el National Institutes of Health de los EE.UU. Fue diseñado con una arquitectura abierta que proporciona extensibilidad vía plugins de *Java* y macros grabables (V and A, 2004). Es capaz de mostrar, editar, analizar y procesar imágenes de hasta 32 bits (millones de colores). Soporta pilas o lotes, una serie de imágenes que comparten una sola ventana, y es multiproceso, de forma que las operaciones que requieren mucho tiempo se pueden realizar en paralelo en hardware multi-CPU. *ImageJ* puede calcular el área, medir distancias y

ángulos, crear histogramas de densidad y gráficos de línea de perfil. Es compatible con las funciones estándar de procesamiento de imágenes tales como operaciones lógicas y aritméticas entre imágenes, manipulación de contraste, convolución, análisis de Fourier, nitidez, suavizado, detección de bordes y filtrado de mediana. Hace transformaciones geométricas como ampliar, rotación y volteos (flips). El programa es compatible con cualquier número de imágenes al mismo tiempo, limitado solamente por la memoria disponible. (Ferreira and Rasband, 2011)

Los plugins escritos por usuarios hacen posible resolver muchos problemas de procesado y análisis de imágenes, desde de imágenes en vivo de las células en tres dimensiones, procesado de imágenes radiológicas, comparaciones de múltiples datos de sistema de imagen hasta sistemas automáticos de hematología. La arquitectura de plugins de *ImageJ* y el entorno de desarrollo integrado lo han convertido en una plataforma popular para la enseñanza del procesamiento digital de imágenes. (W and M, 2007)

1.3.2. Herramientas para el desarrollo de aplicaciones de procesamiento de imágenes

Entre las herramientas para el desarrollo de aplicaciones de procesamiento de imágenes se encuentra *OpenCV* (Open Computer Vision Library), esta es una biblioteca libre, de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que esta biblioteca se publica bajo una licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

OpenCV es multiplataforma, existiendo versiones para los sistemas operativos *GNU/Linux*, *Mac OS X* y *Windows*. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica.

El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado, realizando su programación en código *C* y *C++* optimizados,

aprovechando además las capacidades que proveen los procesadores multi-núcleo. *OpenCV* puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel. (Sourceforge, 2012)

OpenCV ha sido usada en el sistema de visión del vehículo no tripulado Stanley de la Universidad de Stanford, ganador en el año 2005 del Gran desafío DARPA (U.S. Defense Advanced Research Projects Agency).

Fiji (*Fiji is Just ImageJ*): es un paquete de procesamiento de imágenes, que incluye el programa de procesamiento de imágenes *ImageJ* (ver 1.3.1 Aplicaciones disponibles para el procesamiento de imágenes), *Java*, *Java 3D* y muchos complementos organizados de *ImageJ*. Está enfocado fundamentalmente para uso en investigación en las ciencias de la vida. Para los usuarios, *Fiji* es fácil de instalar y tiene una opción de actualización automática, engloba muchos complementos y ofrece una completa documentación. *Fiji* es un proyecto de desarrollo libre, por lo que cualquier persona interesada puede colaborar en su mejora accediendo al código fuente o participando en los grupos de desarrollo de la herramienta y sus complementos. (Fiji)

ITK: En 1999 la Biblioteca Nacional de Medicina del Instituto Nacional de Salud de los Estados Unidos otorgó un contrato por tres años para el desarrollo de un conjunto de herramientas de código abierto para el registro y la segmentación, el cual eventualmente llegó a conocerse como *ITK* (Insight Toolkit). (Kitware, 2012)

ITK está implementado en *C++* y es multi-plataforma, utilizando el ambiente de construcción de *CMake* para manejar el proceso de configuración. Además, es un proceso automatizado de envoltura, genera interfaces entre *C++* y lenguajes de programación interpretados tales como *Java* y *Python*. Esto permite a los desarrolladores crear software usando una variedad de lenguajes de programación. El estilo de implementación de *ITK* se conoce como programación genérica (usando código emplantillado). Ese emplantillado significa que el código es muy eficiente, y que muchos problemas de software se descubren en tiempo de compilación, antes de la ejecución del programa. Esta genericidad permite además a *ITK* trabajar en dos, tres o más dimensiones. Debido a que *ITK* es un proyecto de

código abierto, desarrolladores de todo el mundo pueden usar, mantener y extender la herramienta. En *ITK* se aplica un proceso extensivo de pruebas (usando *CDash*) para medir la calidad diariamente. (Kitware, 2012)

1.4 Metalografía cuantitativa

La metalografía cuantitativa juega un rol importante en la ciencia de los materiales y la ingeniería (Underwood, 1970, DeHoff and Rhines, 1968). Puede proveer de relaciones entre procesos, microestructuras y propiedades mecánicas y suministrar la información de primera mano que es necesaria e importante para establecer un modelo matemático razonable. Permite que el costoso método de prueba y error sea reemplazado por un procedimiento científico experimental, resultando en el mejoramiento de los materiales conocido y el desarrollo de nuevos materiales. (Huang and Froyen, 2001)

La parte más importante de la metalografía es el examen microscópico de una probeta adecuadamente preparada, empleando el microscopio metalográfico entre 50 y 1000 aumentos. Tales estudios proporcionan abundante información sobre la constitución de la aleación o del metal investigado. (Rojas et al., 2011)

La caracterización microestructural es el proceso por el cual se pueden definir diversas características estructurales de muchos materiales como los aceros, aleaciones, entre otros. En estos se pueden obtener características como el tamaño de grano, la forma y distribución de las fases que componen la aleación y de las inclusiones no metálicas, la presencia de segregaciones y heterogeneidades que pueden modificar las propiedades mecánicas y el comportamiento general del metal. (Rojas et al., 2011)

1.4.1 Metalografía cuantitativa convencional

Normalmente todas las determinaciones cuantitativas de fases y propiedades en aceros a partir del análisis microestructural mediante el uso del microscopio óptico metalográfico, se realizan directamente sobre la microestructura o sobre las micrografías reveladas y sus resultados dependen en gran medida de la habilidad del técnico especialista.

Debido a la naturaleza de las imágenes, la extracción manual de los parámetros microestructurales implica largo tiempo de análisis, limitaciones en el tratamiento de los datos estadísticos y el empleo de un criterio poco objetivo en las medidas, lo que conlleva a una poca precisión. En realidad el observador se puede distraer fácilmente de lo que es importante, por detalles triviales de la imagen, sumado al cansancio visual, tenderá a hacer suposiciones que pueden falsear la cuantificación.

A esto se suma la dependencia creciente de laboratorios con equipamiento de alta tecnología y de su personal técnico para el análisis estructural de los materiales, en particular de los aceros, ya que estos son considerados como la única alternativa para lograr los análisis respectivos. El empleo de probetas normalizadas, las prácticas destructivas, el número de veces que hay que repetir un ensayo, aumentan el costo y el tiempo requerido para lograr la caracterización de los materiales.

Cuando un examen microscópico ha permitido la determinación de todas las características constitucionales, es posible predecir con gran seguridad el comportamiento del metal cuando se le utilice para un fin específico (Fajardo et al., 2007). Importancia parecida tiene el hecho de que con ciertas limitaciones, la microestructura refleja casi la historia completa del tratamiento mecánico y térmico que ha sufrido el material. Solo el estudio a fondo de los metales en el microscopio ha permitido resolver muchos problemas de la metalurgia física (Callister W, 1999) y se puede predecir que los progresos en este campo serán parcial o totalmente debidos a los resultados del examen microscópico.

1.4.2 Software para el procesamiento de imágenes aplicado a la metalografía.

La aplicación de la tecnología de análisis de imágenes mediante analizadores automáticos en las mediciones de fases y tamaño de grano posibilitaría un método rápido y confiable para la cuantificación microestructural y posterior caracterización del material observado. El desarrollo y avance de tecnologías en el área de la electrónica y la computación han permitido la automatización de procedimientos de análisis de imágenes los cuales reducen el tiempo y los posibles errores durante el análisis microestructural de las muestras metalográficas.

En la actualidad los programas profesionales para el tratamiento de imagen digital en el campo de la ciencia de los materiales son escasos y los existentes son muy costosos. De poseer programas especializados, accesibles a pequeños empresarios, a centros de investigaciones o instituciones con pocos recursos, se permitiría a partir del tratamiento de una imagen digital de la microestructura de los aceros, estimar algunas propiedades mecánicas como dureza, resistencia a la rotura, límite de fluencia, resistencia a la fatiga, sin la necesidad del uso de un laboratorio con gran inversión, es decir, con máquinas tradicionales como durómetros, máquinas para el ensayo de fatiga, máquinas para ensayo de tracción, máquinas para la determinación de porcentaje de carbono por análisis químico, lo cual ahorraría tiempo, dinero y además se evitaría el uso de probetas normalizadas permitiendo entonces la realización de estimación o caracterización de estos aceros in situ.

Algunos ejemplos de software desarrollado en el campo de la metalografía son:

- **Dewinter**, que es útil para medir y analizar el tamaño de partícula, forma partículas, recuento de partículas, la detección de cuerpos extraños, estructura de la partícula. Precio: \$37,990.00
- **Nascent Metallurgy Plus**; Esta herramienta complementa y amplía el análisis logrado a través de los resultados obtenidos con un microscopio metalográfico. Permite la pre-visualización de reportes, varios modos de captura, zoom y galería de imágenes, entre otros.

1.5 Conclusiones parciales

A partir de la consulta de la bibliografía internacional y nacional realizada, así como de otras fuentes referenciales se pueden extraer las conclusiones fundamentales siguientes:

- Existen una serie de herramientas y aplicaciones para el procesamiento digital de imágenes, con aplicación a la metalografía. Las aplicaciones profesionales existentes en el campo de las ciencias de los materiales, específicamente la metalografía son pocas y generalmente son poco accesibles.

- Al analizar las herramientas de desarrollo de aplicaciones de procesamiento digital de imágenes, se pudo concluir que *ITK* es una solución factible, gracias a su amplio espectro de aplicación y a las garantías de calidad que ofrece.
- Con el estudio realizado, es posible mejorar el proceso convencional de análisis de las micrografías de acero, tarea muy complicada para los especialistas, debido al tiempo que debe ser destinado al análisis de estas imágenes y la fatiga visual generada por la misma.

**Caracterización de aceros mediante el
procesamiento de imágenes
metalográficas**

CAPÍTULO 2. CARACTERIZACIÓN DE ACEROS MEDIANTE EL PROCESAMIENTO DE IMÁGENES METALGRÁFICAS

La aplicación de técnicas de procesamiento digital de imágenes a la metalografía, en sustitución de métodos convencionales de observación y análisis, tiene varias ventajas. Además de servir como una alternativa más económica a pruebas de laboratorio que determinan propiedades mecánicas, permite un aumento de la precisión de las mediciones, un gran ahorro en tiempo y asegura la reproducibilidad del proceso.

2.1 Características de las imágenes metalográficas

Las imágenes metalográficas de aceros normalizados o recocidos, de medio y bajo contenido de carbono, tienen como característica la presencia de dos fases fácilmente distinguibles: perlita y ferrita. Desde el punto de vista visual, estos aceros muestran una microestructura de ferrita continua, con un color predominantemente blanco, con islas o gránulos de perlita que se muestran de un color oscuro (Brick and Phillips, 1949).

Desde el punto de vista computacional, se puede decir que la microestructura de los aceros objeto de análisis de la aplicación tienen los siguientes rasgos:

- Se pueden representar, sin perder generalidad, a partir de imágenes en escala de grises. O sea, el valor de cada pixel está dado por una función que determina la intensidad, con el valor correspondiente al color negro como mínimo de la función y el valor correspondiente al color blanco como máximo.
- En la imagen se pueden apreciar dos clases que se diferencian en la intensidad de los pixeles que las componen. La ferrita, predominantemente blanca, está compuesta por pixeles de alta intensidad; y la perlita, que se muestra de un color oscuro, compuesta por pixeles de baja intensidad.

Deficiencias en el procedimiento de preparación de la muestra y captura de la imagen hacen que estas características tan deseables para lograr un correcto análisis se vuelvan imprecisas, creando fluctuaciones en las intensidades de los pixeles constituyentes de cada

clase que se consideran como ruido, y hacen que el proceso de distinción entre las dos clases no sea obvio. Mediante la aplicación de filtros se puede reducir en cierta medida estas imprecisiones, lo que permite hacer una determinación más exacta de los componentes de la imagen.

2.2 Procesamiento y análisis

Siguiendo la jerarquía de operaciones de la teoría de procesamiento digital de imágenes, se desarrolló el siguiente procedimiento para las micrografías de acero, utilizando en cada paso operaciones de filtrado que se ajustan al problema y que mostraron tener buenos resultados.

El procedimiento desarrollado para el procesamiento de imágenes metalográficas consta de tres fases principales:

- 1- **Pre-procesamiento.** Esta fase se encarga de reducir el ruido presente en la imagen, atenuando las diferencias de intensidad entre píxeles vecinos y haciendo resaltar más los bordes que definen los límites de los gránulos de perlita. Esto se logra haciendo uso del filtrado bilateral que provee la herramienta *ITK*.
- 2- **Segmentación.** La segmentación crea una imagen binaria a partir de la imagen pre-procesada. Los píxeles de la imagen resultante pueden tener solo dos valores, uno que identifica que el píxel pertenece a un objeto de interés, y otro que identifica al píxel como fondo de la imagen. Para esto se utilizó el filtro de segmentación *Otsu* presente en *ITK*, que calcula un umbral de intensidad que divide la imagen en dos clases, maximizando la varianza entre estas, lo que equivale a minimizar el error de clasificación. Los valores de la imagen binaria resultante de la segmentación se escogieron como blanco para ferrita, y negro para perlita, debido a la similitud que guardan con la representación original de la microestructura.
- 3- **Refinamiento de la segmentación.** Esta fase hace un refinamiento de la imagen binaria obtenida luego de la segmentación en base a criterios obtenidos mediante la consulta a expertos. Primeramente se hace una erosión de la imagen, removiendo píxeles del borde de las zonas identificadas como objetos de interés, y luego se hace

un relleno de huecos dentro de los objetos de interés, para lograr una representación uniforme de estos. Las facilidades para realizar estas dos operaciones se pueden encontrar en filtros presentes en la herramienta *ITK*, de los cuales se usaron un filtro de erosión para imágenes binarias y un filtro de relleno de huecos para imágenes binarias mediante votación.

Luego de realizado el procesamiento, la imagen resultante es una imagen binaria con solo dos valores posibles de sus píxeles y una representación uniforme de los gránulos de perlita, lo que posibilita que el análisis se haga de forma sencilla. Este análisis se hace mediante un conteo de píxeles de interés, y el valor total hallado se divide entre la cantidad total de píxeles de la imagen, lo que da como resultado el porcentaje que representa.

Al obtener el porcentaje de perlita en la imagen, es trivial calcular el porcentaje de ferrita que representa el resto. Con estos dos valores se pueden calcular propiedades mecánicas del material, despejándolos en ecuaciones de la rama de Ciencias de los Materiales, y se puede hallar además el porcentaje de carbono presente en la muestra lo que permite clasificar el acero mediante la norma AISI (American Iron and Steel Institute) para tener un referente de comparación en los valores de las propiedades mecánicas.

En el siguiente epígrafe se detallarán los detalles que se tuvieron en cuenta para la concepción de la aplicación computacional encargada de realizar los procesos descritos anteriormente.

2.3 Concepción de la aplicación

La aplicación para la caracterización de aceros al carbono mediante el procesamiento digital de imágenes metalográficas fue concebida de manera que tuviera las siguientes prestaciones:

- Capacidad para cargar imágenes 2D en formato JPEG, BMP o PNG.
- Capacidad para guardar las imágenes procesadas en formato JPEG.

- Visualización de las imágenes procesadas, permitiendo mostrar de forma opcional la imagen original para efectuar una evaluación de la calidad de los resultados del procesamiento mediante comparación visual.
- Interfaz para efectuar procesamiento automático, que comprende la aplicación de todos los filtros mencionados anteriormente en cada una de las fases del procedimiento desarrollado. Esta interfaz se concibe de manera que requiera la menor interacción posible por parte del usuario.
- Interfaz para la aplicación de cada uno de los filtros, comprendidos en el procedimiento desarrollado, de forma independiente y con mayor interacción por parte del usuario para permitir la obtención de resultados más precisos. Esto también permite que el usuario vaya observando la evolución de la imagen resultante en cada paso del procedimiento.
- Interfaz para realizar el análisis de la imagen, que muestre como resultados los porcentajes de fases presentes, contenido de carbono de la muestra y los valores de otras propiedades mecánicas.
- Capacidad para guardar los resultados del procesamiento y análisis de la imagen en formato Texto. Dentro de los datos que se guardan están: nombre del archivo que se procesa, nombre de cada uno de los filtros que se aplicaron con sus parámetros, y resultados del análisis.

Para lograr una aplicación con las prestaciones mencionadas fue necesario el uso de las siguientes herramientas:

- **Qt** como biblioteca de componentes gráficos para la construcción de la interfaz gráfica de usuario (GUI, Graphic User Interface).
- **ITK** como herramienta de procesamiento de imágenes.
- **VTK** como herramienta que provee los componentes para la visualización de las imágenes.
- **Python** como lenguaje de programación para interconectar las herramientas y desarrollar la aplicación.

A continuación se realiza una pequeña caracterización de estas herramientas, así como los detalles utilizados para el procesamiento y análisis de las imágenes.

2.3.1 Lenguaje de programación Python

Python es un potente lenguaje de programación fácil de aprender. Tiene eficientes estructuras de datos de alto nivel y un enfoque de programación orientada a objetos simple pero eficaz. La elegante sintaxis de *Python* y el tipado dinámico, junto con su naturaleza interpretada, lo hacen un lenguaje ideal para realizar *script* y el desarrollo rápido de aplicaciones en muchas áreas, en la mayoría plataformas. (Swaroop, 2005)

Características de *Python*:

- Lenguaje simple, es un language simple y minimalista. Leer un buen programa en Python es casi como leer en inglés. Este “pseudo-código” de *Python* se considera una de sus grandes fortalezas. Permite concentrarse en la solución del problema en vez de en el lenguaje mismo.
- Fácil de aprender, es extremadamente fácil comenzar a trabajar con *Python* ya que tiene una sintaxis extraordinariamente simple.
- Libre y de Código Abierto, es un ejemplo de FLOSS (Free/Libré and Open Source Software).
- Lenguaje de alto nivel, cuando se escriben programas en *Python* no es necesario preocuparse por detalles de bajo nivel como el manejo de memoria del programa, etc.
- Portable, gracias a su naturaleza de código abierto, ha sido portado a varias plataformas.
- Interpretado, no requiere compilación, los programas se corren directamente del código fuente. Internamente *Python* convierte el código fuente en código intermedio y lo traduce al lenguaje nativo de la computadora para luego correrlo.
- Orientado a Objetos, *Python* soporta programación procedural y programación orientada a objetos.

- Extensible, se pueden escribir partes de un programa en otro lenguaje, como C++, y luego usarlos desde un programa Python.
- Bibliotecas Extensivas, la biblioteca estándar de *Python* es inmensa. Además existen varias bibliotecas de gran calidad tales como *wxPython*, la cual es posiblemente el toolkit para desarrollo de interfaces gráficas en *Python* más popular, superando incluso a *TKinter*, que se incluye por defecto con el intérprete de *Python*.

La decisión de utilizar *Python* como lenguaje para desarrollar la aplicación se basó principalmente en sus características que permiten el desarrollo rápido de aplicaciones, y en la portabilidad de bibliotecas de procesamiento de imágenes y visualización a este lenguaje.

2.3.2 Marco de trabajo Qt

Qt es un marco de trabajo multiplataforma ampliamente usado para desarrollar software con interfaz gráfica (GUI) (en cuyo caso se clasifica como un widget toolkit), y además usada para desarrollar programas sin interfaz gráfica tales como herramientas de líneas de comandos y consolas para servidores. Es utilizado notablemente en aplicaciones como Autodesk Maya, elementos de Adobe Photoshop, Skype, VLC media player, Virtual Box y Mathematica. Además, es utilizado por la Agencia Espacial Europea, DreamWorks, Google, HP, KDE, Lucasfilm, Panasonic, Philips, Samsung, Siemens, los estudios de animación Walt Disney, entre otros (Qt-project).

Qt es desarrollado por un proyecto de código abierto, el Proyecto *Qt*, involucra desarrolladores individuales y firmas que trabajan para evolucionar *Qt*, tales como Nokia, Digia y otras. *Qt* usa el lenguaje C++ estándar, pero hace uso extensivo de un generador de código especial (MOC, del inglés Meta Object Compiler), conjuntamente con varias macros para enriquecer el lenguaje. Es posible utilizarlo en varios lenguajes de programación, corre en las mayores plataformas desktop y algunas plataformas móviles, teniendo soporte de internacionalización extensivo. Algunas características no gráficas incluyen acceso a bases de datos *SQL*, análisis gramático (parsing) de XML, manejo de hilos, soporte de red, y un API (del inglés Application Programming Interface) multiplataforma unificada para el manejo de archivos. Distribuido bajo los términos de la

GNU Lesser General Public License (entre otras), *Qt* es software libre y de código abierto. Todas sus ediciones tienen soporte para varios compiladores, incluidos GCC, el compilador C++ y la suite de Visual Studio (Qt-project).

PyQt es una distribución para *Python* del conjunto de herramientas multiplataforma *Qt*. Es una de las alternativas para la programación de GUI en *Python*. Es software libre al igual que *Qt*. *PyQt* implementa cerca de 440 clases y cerca de 6,000 funciones y métodos incluyendo:

- Una cantidad substancial de widgets para GUI.
- Clases para acceder a bases de datos SQL (ODBC, MySQL, PostgreSQL, Oracle).
- Widgets de datos que se rellenan automáticamente a partir de una base de datos.
- Un analizador gramático para XML.

PyQt4 contiene, entre otros, los siguientes módulos de *Python*:

- *QtCore*, módulo que contiene el núcleo de las clases no-GUI, incluyendo el mecanismo de eventos, señales y ranuras (slots). Además incluye una abstracción de Unicode dependiente de la plataforma, hilos, mapeo de ficheros, memoria compartida, expresiones regulares y otras configuraciones de usuario y aplicación.
- *QtGui*, que contiene la mayoría de las clases GUI. Estas incluyen un número de clases de tablas, árboles y listas basadas en el patrón de diseño MVC (Modelo-Vista-Controlador).

En el desarrollo de la aplicación se utiliza de esta herramienta, principalmente las clases necesarias para construir componentes gráficos atracables, (DockWidgets) por las facilidades que brindan para crear una interfaz gráfica muy dinámica y ajustable al gusto del usuario. Además se utilizan las clases para construir una ventana principal y otros componentes gráficos básicos para lograr la funcionalidad de la aplicación. El comportamiento de estos objetos está definido en las clases *QDockWidget* y *QMainWindow*, para los componentes atracables y la ventana principal respectivamente.

La clase *QDockWidget* provee un componente gráfico que puede ser atracado dentro de una ventana principal (*QMainWindow*) o mantenerse flotando como una ventana. Los

DockWidgets son ventanas secundarias colocadas en las áreas de atraque alrededor del widget central en una ventana principal, se pueden mover dentro de su área actual, hacia nuevas áreas y hacer flotar (no atracar) por el usuario. El API de la *QDockWidget* permite al programador restringir la habilidad de los widgets para moverse, flotar, o cerrarse, así como las áreas en las que se puede colocar. Un DockWidget consiste de una barra de título y un área de contenido. La barra de título muestra el título de la ventana, un botón para flotar y otro para cerrar. Dependiendo del estado del *QDockWidget*, estos botones pueden estar deshabilitados o no mostrarse. En la figura 2.1 se muestra la estructura de una ventana principal en *Qt*.

La clase *QMainWindow* provee de un marco de trabajo para construir aplicaciones con interfaz de usuario. Una ventana principal puede contener una barra de menú, barras de herramientas, y una barra de estado. La presencia de un widget central es obligatoria, aunque este no contenga contenido, y alrededor de él se definen las áreas de atraque: superior, inferior, izquierda y derecha.

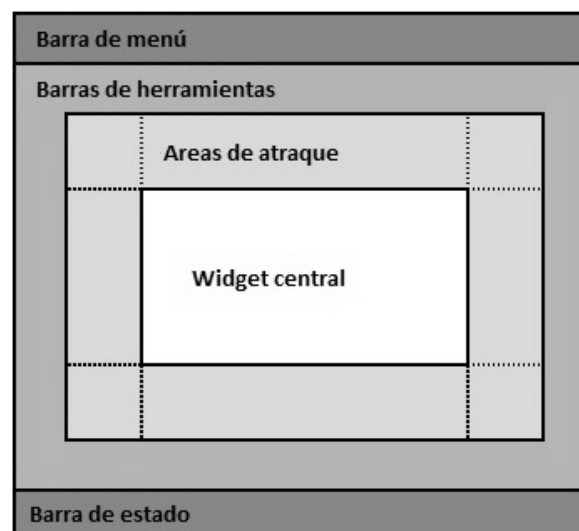


Fig 2.1 Estructura de una ventana principal en *Qt*

2.3.3 Paquete de herramientas ITK

Para utilizar el paquete de herramientas *ITK* desde *Python* se puede optar por una de al menos dos opciones:

- 1- Compilar y portar el código fuente en *C++* hacia *Python* utilizando herramientas de envoltura (wrapping) tales como *CableSwig*. Esta vía tiene como ventaja que permite contar en *Python* con todas las prestaciones que necesite el usuario, definiéndolas en el momento de realizar el proceso de envoltura. Su desventaja es que al distribuir una aplicación que utilice la biblioteca *ITK* portada a *Python* por esta vía, es necesario distribuir también esta.
- 2- Utilizar la biblioteca *ITK* incorporada en una de las distribuciones libres de bibliotecas, herramientas y documentación para *Python*, *PythonXY*. La ventaja de utilizar esta vía radica en que se utiliza la biblioteca de una forma estándar y al distribuir la aplicación solo es necesario dar el código fuente y que el usuario tenga instalada una distribución *PythonXY* con *ITK*. Su desventaja radica en que estas distribuciones de *ITK* no portan todas las clases y características de la biblioteca original.

Para el desarrollo de la aplicación se optó por la segunda vía en interés de lograr una fácil distribución y uso de la misma por parte de los usuarios que la reciben. Además, la distribución de la biblioteca *ITK* utilizada carece de una sola característica necesaria para el desarrollo de la aplicación, y esta es de fácil reproducibilidad.

De *ITK* se utilizaron las siguientes clases que definen el comportamiento de los filtros necesarios para el procesamiento de las imágenes y otras funciones dentro de la aplicación:

- *Image*, clase plantilla que define una imagen n -dimensional. Las imágenes se emplantillan sobre un tipo de y una dimensión.
- *ImageFileReader*, clase para la lectura de archivos de imágenes de una variedad de formatos.
- *ImageFileWriter*, clase para escribir hacia archivo una imagen. Normalmente, con solo ajustar el nombre de archivo con un sufijo adecuado y configurar la entrada al objeto de la clase es suficiente para que funcione correctamente.
- *BilateralImageFilter*, filtro de suavizado de imágenes que además preserva bordes.

- *OtsuThresholdImageFilter*, filtro que segmenta la imagen separándola en dos componentes: objetos (foreground) y fondo (background).
- *FlatStructuringElement*, clase que provee métodos estáticos para crear elementos estructurales que se utilizan en operaciones morfológicas realizadas sobre la imagen.
- *BinaryErodeImageFilter*, operación morfológica de erosión sobre imágenes binarias (ej. resultantes de una segmentación).
- *VotingBinaryHoleFillingImageFilter*, llena huecos y cavidades aplicando una operación de votación en cada pixel. Píxeles de fondo con más del 50 por ciento de píxeles en su vecindario siendo de tipo objeto, se transforman en píxeles tipo objeto.
- *ImageToVTKImageFilter*, filtro de conversión del formato de representación de imágenes en *ITK* al tipo de datos imagen en *VTK*, para la visualización por parte de este último.

Debido a la implementación genérica de *ITK*, cada clase que define filtros es necesario emplantillarla con los tipos de datos sobre los que va a trabajar, y luego se instancia mediante el método *New()*.

Ejemplos de código:

`ImageType = itk.Image[itk.UC, 2]` define una variable de tipo clase imagen bidimensional con píxeles tipo *char* (8-bits) sin signo. El tipo de imagen que se utiliza a lo largo de la aplicación.

`lector_imagen = itk.ImageFileReader[ImageType].New()` instancia la clase para leer archivos de imagen. El filtro devuelve una imagen *ITK* del tipo con que se emplantilla la clase (*ImageType*).

`filtro_bilateral=itk.BilateralImageFilter[ImageType, ImageType].New()` instancia la clase para el filtrado bilateral, emplantillada sobre los tipos de imágenes de entrada al filtro y de salida del filtro.

`Kernel = itk.FlatStructuringElement[2]` define una variable de tipo clase de elementos estructurales bidimensionales.

`filtro_erosion = itk.BinaryErodeImageFilter [ImageType, ImageType, Kernel].New()` la clase del filtro de erosión se emplantilla sobre los tipos de imágenes de entrada y salida, y sobre la clase de elementos estructurales utilizados para las operaciones morfológicas.

Los filtros de *ITK* comparten un mismo diseño para su utilización. Estos se basan en los siguientes métodos principales:

- `SetInput(imagen)` método para proveerle la imagen de entrada al filtro.
- `Update()` método que ejecuta el filtro y pone sus resultados disponibles en la salida.
- `GetOutput()` método para tomar la imagen de salida del filtro.

Además de estos métodos, cada filtro tiene métodos *Set/Get* para los parámetros necesarios para su funcionamiento, otros datos de salida aparte de la imagen resultante, y algunas propiedades internas del filtro. Los filtros también se pueden conectar en lo que se denomina “tubería”, donde la salida de un primer método es utilizado como parámetro de entrada de un segundo método. Esto hace que el método *Update()* de filtros anteriores se dispare automáticamente (no es necesario programarlo implícitamente) cuando se actualiza o se pide la salida de un filtro en la tubería.

En la figura 2.2 se muestra un fragmento de código donde se ejemplifica la utilización de filtros.

Esta tubería lee una imagen de archivo, le aplica filtrado bilateral y luego la guarda en un fichero. Nótese que solo es necesario llamar explícitamente al método *Update()* del último filtro de la tubería.

Dentro de las funcionalidades requeridas para la aplicación, y que no están presentes en la distribución utilizada de *ITK* para *Python*, se encontraron los iteradores sobre imágenes, que permiten obtener información de esta pixel a pixel. Dicha funcionalidad se implementó

como parte del proceso de análisis de la imagen, y su comportamiento está definido en la clase *ITKImageIterator* que utiliza el patrón de diseño de los iteradores de *ITK*.

```
lector = itk.ImageFileReader[TipoImagen1].New()
lector.SetFilename(archivo)
bilateral = itk.BilateralImageFilter[TipoImagen1, TipoImagen2].New()
bilateral.SetInput(lector.GetOutput())
escritor = itk.ImageFileWriter[TipoImagen2].New()
escritor.SetInput(bilateral.GetOutput())
escritor.SetFilename(nuevo_archivo)
escritor.Update()
```

Fig 2.2 Ejemplo de utilización de los filtros

2.3.4 Paquete de herramientas VTK

El paquete de herramientas VTK (del inglés Visualization Toolkit) es un sistema de software libre y de código abierto para procesamiento de imágenes, visualización y gráficos 3D. Fue creado originalmente en 1993 como software acompañante del libro "The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics" publicado por Prentice-Hall. El libro y el software fueron escritos por tres desarrolladores: Will Schroeder, Ken Martin y Bill Lorensen. (Schroeder et al., 2006)

VTK consiste en una biblioteca de clases escritas en C++ y varias capas de interfaz interpretadas para *Tcl/Tk*, *Java* y *Python*. Kitware, cuyo equipo creó y continua extendiendo el paquete de herramientas, ofrece soporte profesional y servicios de consulta para *VTK*. *VTK* soporta una amplia variedad de algoritmos de visualización incluyendo visualización de: escalares, vectores, tensores, texturas y volúmenes; y técnicas avanzadas de modelación tales como: modelación implícita, reducción de polígonos, suavizado, corte y contorno de mallas, y triangulación de Delaunay. *VTK* tiene un marco de trabajo extensivo de visualización de la información, tiene una suite de widgets de interacción 3D, soporta procesamiento paralelo y se integra con varias bases de datos en paquetes de herramientas GUI tales como *Qt* y *Tk*. Además es multi-plataforma y corre en Linux, Windows, Mac y plataformas Unix. Incluye soporte auxiliar para widgets de interacción

3D, anotación en dos y tres dimensiones y procesamiento en paralelo. En su núcleo VTK está implementado como un paquete de herramientas C++, requiriendo que los usuarios construyan sus aplicaciones combinando varios objetos en una aplicación. Este sistema además soporta envoltura automática del núcleo C++ hacia *Python*, *Java* y *Tcl*, de tal forma que las aplicaciones puedan ser escritas usando esos lenguajes de programación interpretados.

VTK se usa en todo el mundo en aplicaciones comerciales, investigación y desarrollo, y es la base de muchas aplicaciones avanzadas de visualización tales como: ParaView, VisIt, VisTrails, 3DSlicer, MayaVi, y OsiriX.

De esta herramienta se utilizaron las clases necesarias para la construcción de un visor de imágenes 2D y su integración con la GUI que provee *PyQt*. Estas son:

- `vtkImageViewer2`, una clase conveniente para mostrar una imagen 2D. Empaca la funcionalidad encontrada en `vtkRenderWindow`, `vtkRenderer`, `vtkImageActor` y `vtkImageMapToWindowLevelColors` dentro de una sola clase fácil de utilizar. Esta clase además crea un estilo de interacción de imagen (`vtkInteractorStyleImage`) que permite acercar, alejar y hacer movimientos horizontales de la imagen, y soporta operaciones interactivas a nivel de ventana en la imagen.
- `vtkRenderer`, un renderer es un objeto que controla el proceso de visualización de los objetos. La visualización es el proceso de convertir una geometría, una especificación de luces, y un ángulo de cámara en una imagen. Ciertas características avanzadas de visualización tales como iluminación desde dos ángulos también se pueden controlar.
- `vtkCamera`, una cámara virtual para la visualización 3D. Provee métodos para posicionar y orientar el punto de vista, y el punto focal.
- `vtkImageActor`, usada para visualizar una imagen en una escena 3D. La imagen se coloca en el punto de origen de la escena, y su tamaño se controla por las dimensiones de la imagen y su espaciado. La orientación de la imagen es ortogonal a uno de los ejes *x-y-z* dependiendo del plano en que la imagen se defina.

vtkImageActor duplica convenientemente la funcionalidad de combinaciones de otras clases de *VTK* en una sola clase.

- *QVTKRenderWindowInteractor*, clase que define el comportamiento de un widget para *PyQt* versión 4 con la posibilidad de interacción con el usuario. Dentro de este se muestra la visualización de la imagen.

Estas clases de *VTK* se integran en la aplicación dentro de una sola clase, *VTKRenderEngine*, que interactúa con el widget de visualización para mostrar las imágenes.

2.4 Diseño de la aplicación

La aplicación fue concebida de desarrollo rápido, enfocándose en lograr una funcionalidad en un tiempo relativamente corto y por tanto descartando la creación de un diseño meticuloso. Aun así, su diseño final permite extenderla para el uso de más operaciones de filtrado mediante la programación e inclusión de clases que agreguen funcionalidad, y cambios mínimos en el código existente.

2.4.1 Visión general del funcionamiento de la aplicación

El funcionamiento de la aplicación se divide en tres momentos principales:

- **Carga de la imagen.** Utilizando la clase *ImageFileReader* de *ITK*, se carga una imagen desde un archivo especificado mediante la interfaz de usuario. En este momento se inicializan los componentes necesarios de *VTK*, cuya funcionalidad combinada se define en la clase *VTKRenderEngine*, y se visualiza la imagen en los visores activos.
- **Procesamiento.** El procesamiento es la aplicación de uno o más filtros a la imagen. Se considera la opción realizar procesamiento automático, que deja a la imagen lista para ser analizada con un mínimo de interacción; o la aplicación de filtros individuales para la obtención de resultados más personalizados y así mostrar la evolución de la imagen en cada paso del procesamiento. Es responsabilidad del usuario aplicar una correcta secuencia de filtros. La aplicación de filtros binarios

(erosión, rellenado de huecos) sobre una imagen no segmentada no producirá los resultados deseados. Asimismo, analizar una imagen que no ha sido segmentada producirá resultados erróneos.

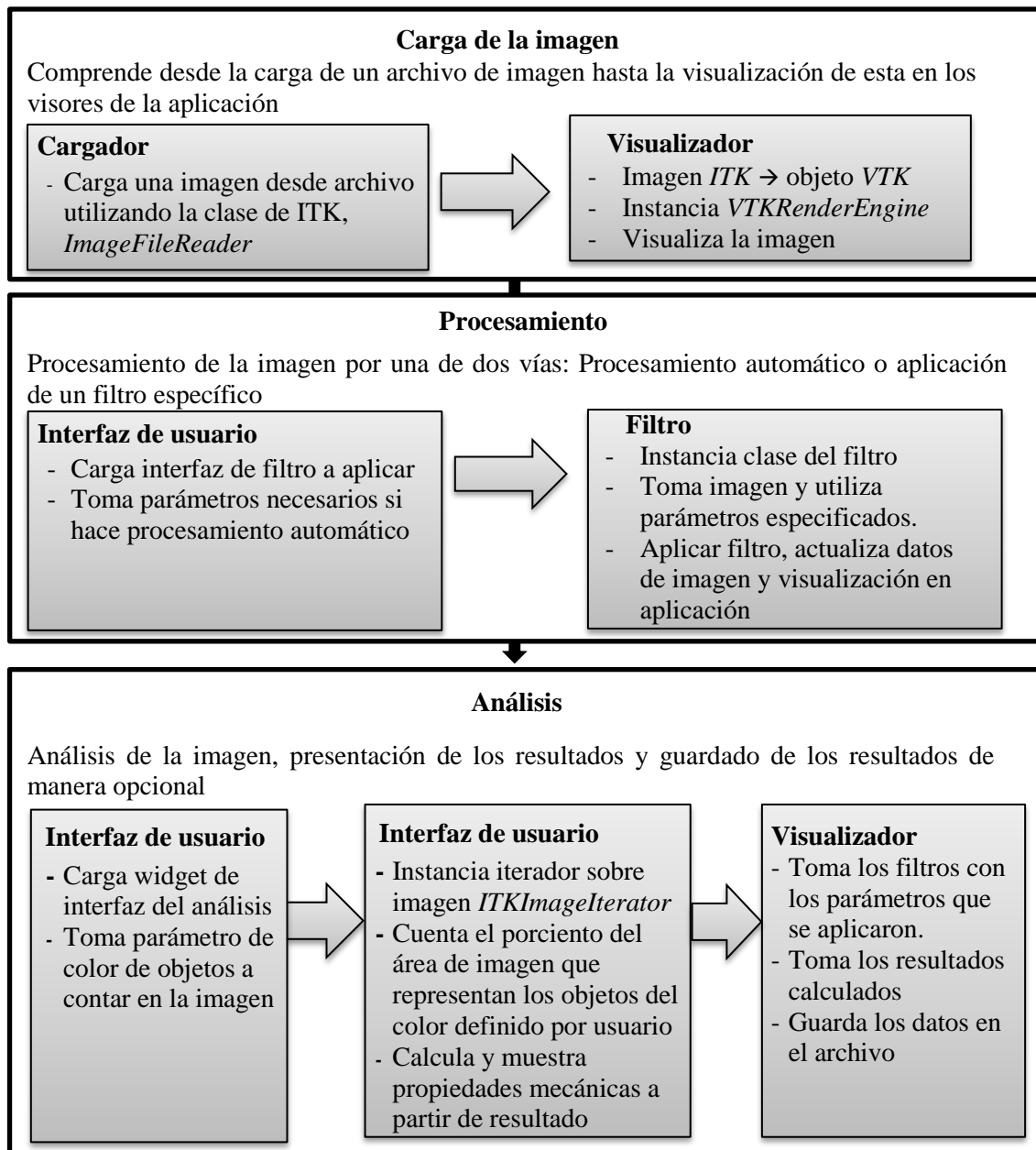


Fig 2.3 Diagrama con la visión general del sistema.

- **Análisis.** La fase de análisis toma la imagen resultante del procesamiento y determina el porciento del área total que representan los objetos del color que se

especifica en la interfaz. A partir de estos datos se calculan las propiedades mecánicas. La interfaz también brinda la posibilidad de guardar resultados en formato texto, estos resultados se pueden guardar en cualquier momento a partir que se carga la imagen y reflejan los procesos aplicados hasta ese momento. En la figura 2.3 se muestra un diagrama que resume lo explicado anteriormente.

2.4.2 Diagrama de clases

La interfaz gráfica de cada filtro y la interfaz del análisis se construyeron siguiendo el mismo diseño.

En la figura 2.4 se muestra el diagrama de clases para la interfaz de procesamiento automático. La clase *Qt4AutoProcessDockWidget* hereda la funcionalidad de un widget atracable de la clase *QDockWidget* del módulo *Gui* de *Qt*, y configura su interfaz gráfica mediante la llamada al método *setupUi()* de la clase *Ui_dockAutoProcess* que toma como parámetro una instancia de *Qt4AutoProcessDockWidget*. La clase *ITKAutoProcessPipeline* se instancia dentro de *Qt4AutoProcessDockWidget* y es la encargada de la captura de los datos y el procesamiento de la imagen.

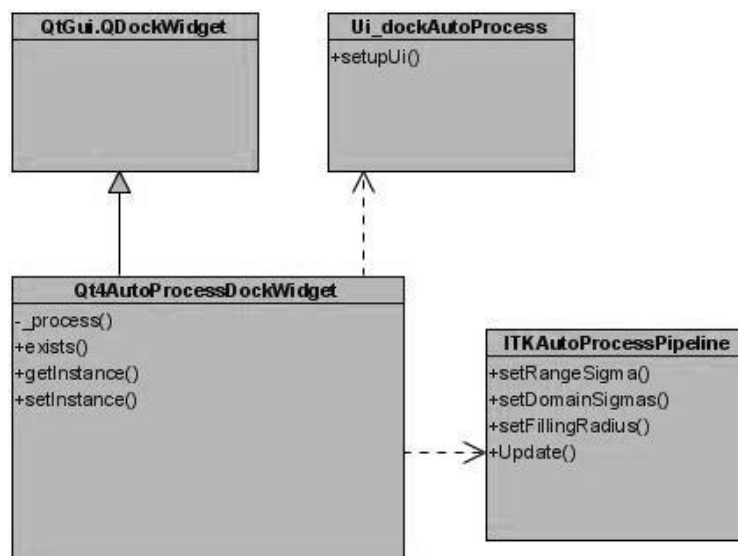


Fig. 2.4 Diagrama de diseño de la interfaz gráfica de los filtros

El resto de las interfaces de filtrado y la de análisis siguen este modelo cambiando solo la parte del nombre de las clases que expresa la interfaz a la que pertenece. Estas son:

- `BilateralFilter`, para el filtrado bilateral de reducción de ruido
- `FillHoles`, para el rellenado de huecos
- `ErodeFilter`, para la erosión
- `OtsuSegmentation`, para la segmentación Otsu
- `Analysis`, para la interfaz de análisis

Para todas las clases del paquete de herramientas *ITK* que se utilizan en la aplicación se implementaron clases que envuelven su comportamiento con el objetivo de separar todo el proceso de emplantillado descrito anteriormente, de la implementación de la funcionalidad de la aplicación (Ver figura 2.5). Estas son:

- `ITKImageFileLoader`, `ITKImageFileWriter` para las clases de lectura y escritura de archivos de imágenes.
- `ITKBilateralFilter`, `ITKErodeFilter`, `ITKOtsuSegmentation` y `ITKFillHoles`, para los filtros bilateral, de erosión, de segmentación y de rellenado de huecos respectivamente.
- `ITKImageToVTK` para envolver la clase de conversión de imágenes de tipo *ITK* a imágenes de tipo *VTK*.

La clase *ITKAutoProcessPipeline* utiliza las cuatro clases que envuelven filtros *ITK*, mencionadas para crear una tubería de procesamiento. Esta clase no accede a componentes *ITK* directamente.

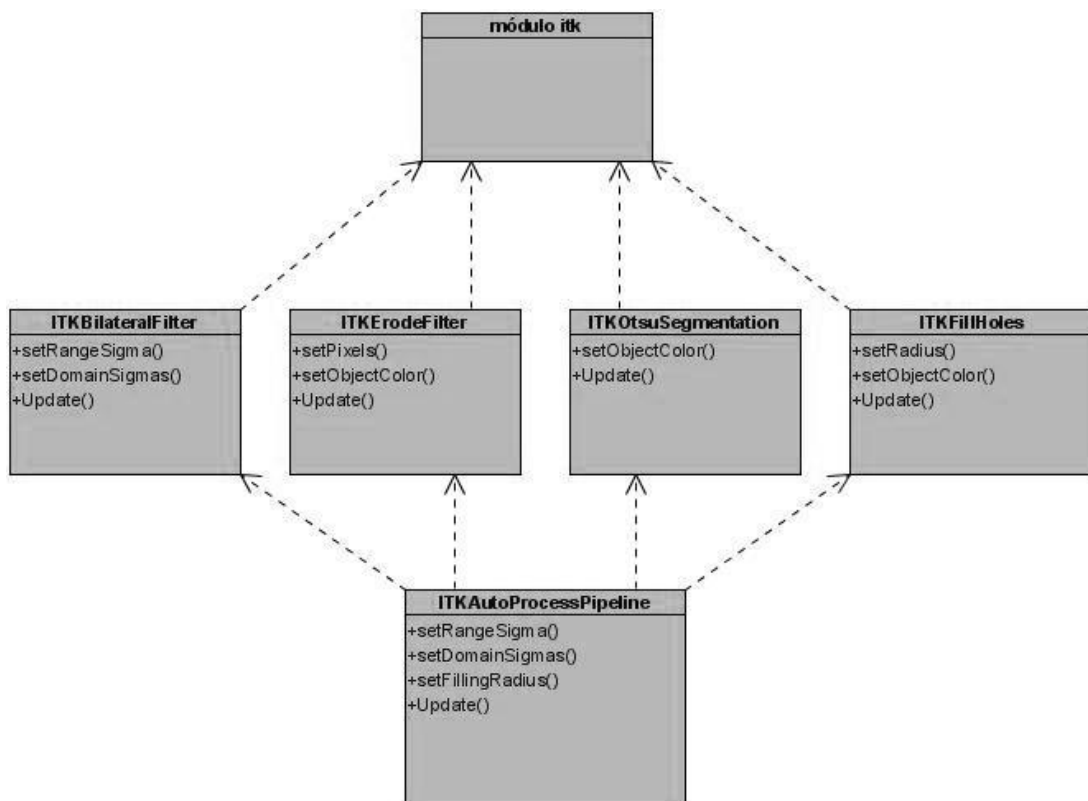


Fig 2.5 Diagrama de clases de los filtros

2.5 Conclusiones parciales

Basándose en las funcionalidades que brinda la herramienta para el desarrollo de aplicaciones de procesamiento digital de imágenes *ITK*, se desarrolló un procedimiento para procesar y analizar las micrografías de acero.

El procedimiento fue conformado conjuntamente con especialistas en el área de la Ciencia de los Materiales.

En la implementación del sistema se tomó como lenguaje de programación *Python* por las facilidades que brinda para la comunicación con el resto de las herramientas utilizadas en la solución del problema, tales como, *Qt* y *VTK*.

**Evaluación de la implementación y
descripción a nivel de usuario
de la aplicación**

CAPÍTULO 3. EVALUACIÓN DE LA IMPLEMENTACIÓN Y DESCRIPCIÓN A NIVEL DE USUARIO DE LA APLICACIÓN

A continuación se describirá cómo es posible utilizar a nivel de usuario la aplicación para la caracterización de aceros al carbono mediante el procesamiento digital de imágenes metalográficas (*AceroPro*). Además, se presentará el caso de estudio definido para la evaluación del sistema implementado y finalmente se analizarán, mediante técnicas estadísticas, los resultados de la aplicación.

3.1 Interfaz de usuarios AceroPro para la caracterización de aceros al carbono mediante el procesamiento digital de imágenes metalográficas

En este epígrafe se describirá cómo utilizar *AceroPro* para realizar la caracterización de aceros al carbono mediante imágenes. Detallando cada una de las opciones que brinda el sistema para realizar el procesamiento de las imágenes.

3.1.1 Ventana principal de la aplicación

La figura 3.1 muestra la ventana principal de la aplicación, en la misma se puede observar una imagen ya cargada en el panel de visualización. En esta figura se marcaron secciones importantes y se detallan a continuación.

1: Barra de menú

- Menú Archivo, posee la opción **Abrir imagen** (cargar una imagen de archivo), **Guardar imagen** (guardar la imagen que se muestra en el visor Imagen Procesada), y **Salir**.
- Menú Edición, posee la opción **Deshacer**. Deshace el último proceso realizado a la imagen. Si se hizo procesamiento automático deshace únicamente el último filtro aplicado.
- Menú Ver, contiene la opción **Imagen original**, para mostrar opcionalmente el visor de la imagen original.

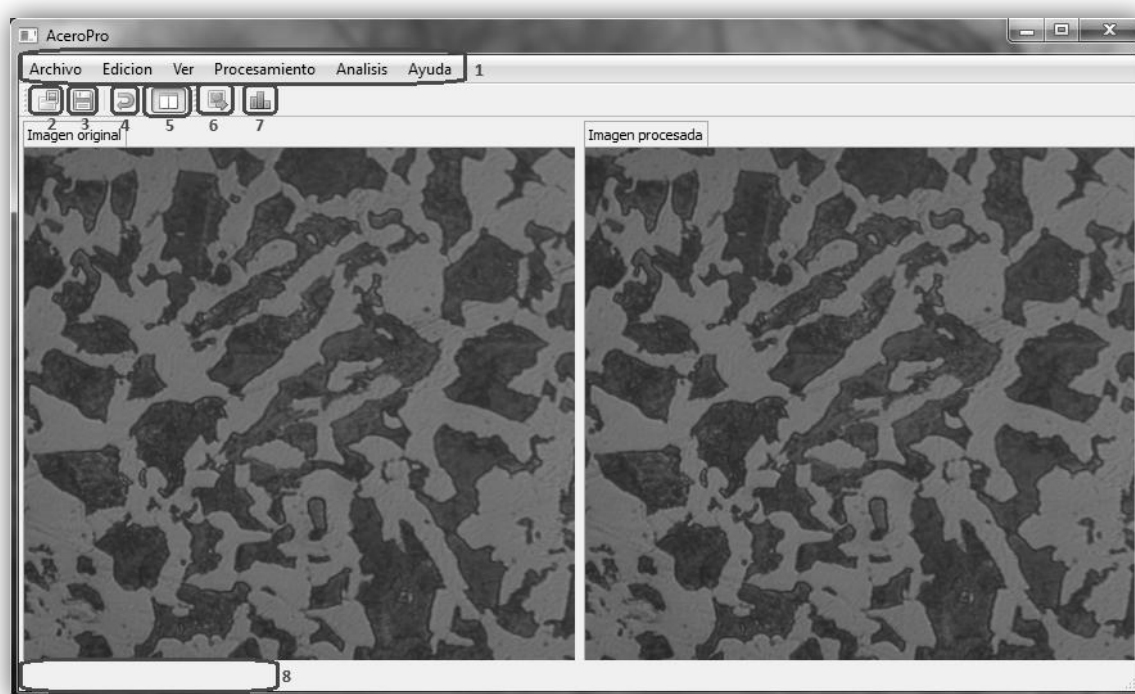


Fig 3.1 Ventana principal de la aplicación.

- Menú **Procesamiento**, tiene la opción **Procesamiento automático**, submenús **Reducción de ruido** y **Segmentación**. El submenú **Reducción de ruido** posee **Filtrado Bilateral**, **Erosión**, y **Rellenar de huecos**. El submenú **Segmentación** tiene el **Segmentador Otsu**.
- Menú **Análisis**, tiene la opción **Caracterización metalográfica**.
- Menú **Ayuda**, posee las opciones **Contenido** y **Acerca de...** para la ayuda y los créditos.

2: Acción Abrir imagen.

3: Acción Guardar imagen.

4: Deshacer.

5: Mostrar el visor de la imagen original.

6: Acción procesamiento automático, abre la GUI de procesamiento Automático.

7: Caracterización metalográfica, abre la GUI de análisis.

8: Barra de estado, muestra información sobre los elementos bajo el cursor, muestra los mensajes **Cargando imagen**, **Procesando**, **Analizando** cuando se carga, procesa o analiza una imagen respectivamente.

3.1.2 Interfaz de procesamiento automático

Para realizar el procesamiento automático de las imágenes es necesario utilizar la ventana que se muestra en la figura 3.2.



Fig 3.2 Ventana para el procesamiento automático.

1: Parámetro del filtro bilateral referente a la similitud entre píxeles en el dominio de la imagen (ver capítulo 1).

2: Parámetro del filtro bilateral referente a la cercanía de píxeles en el rango de la imagen (ver capítulo 1).

3: Radio máximo de los huecos que se pueden rellenar.

3.1.3 Interfaz del segmentador Otsu

Cuando se quiere aplicar técnicas de segmentación usando Otsu (ver epígrafe 1.2.2) se utiliza la interfaz visual que se muestra en la figura 3.3.

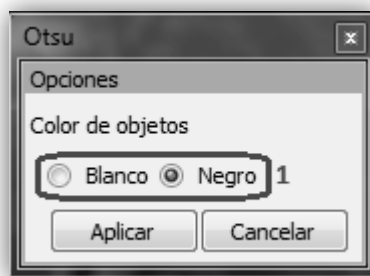


Fig 3.3 Ventana para el segmentador Otsu.

1: Color que van a tomar los objetos. Esto es, color que van a tomar los elementos de la imagen cuya intensidad es menor o igual que el umbral de intensidad. El resto de los elementos toma el color no señalado.

3.1.4 Interfaz del filtro de erosión

La interfaz para aplicar el filtro de erosión se muestra en la figura 3.4.

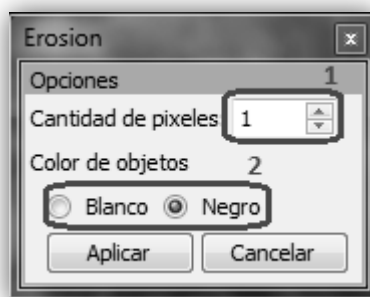


Fig 3.4 Ventana para los parámetros del filtro de erosión.

1: Cantidad de pixeles a remover del borde de los objetos, equivalente a decir que todo pixel cuya distancia en pixeles al borde sea menor o igual que ese valor se elimina.

2: Selector del color de los elementos a ser considerados como objetos y que se van a erosionar.

3.1.5 Interfaz del filtrado bilateral

Si se pretende configurar los parámetros para aplicar el filtrado bilateral es posible utilizar la interfaz visual que se muestra en la figura 3.5.

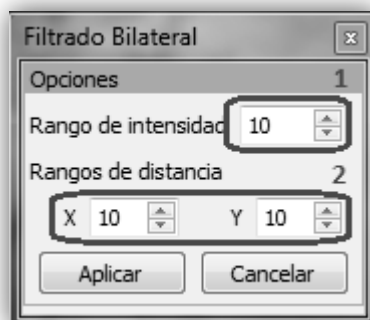


Fig 3.5 Ventana para los parámetros del filtro de bilateral.

1: Parámetro del filtro bilateral referente a la similitud entre píxeles en el dominio de la imagen (ver capítulo 1).

2: Parámetro del filtro bilateral referente a la cercanía de píxeles en el rango de la imagen (ver capítulo 1).

3.1.6 Interfaz del filtro de rellenado de huecos

Los parámetros para aplicar el rellenado de huecos se especifican en la ventana que se muestra en la figura 3.6.

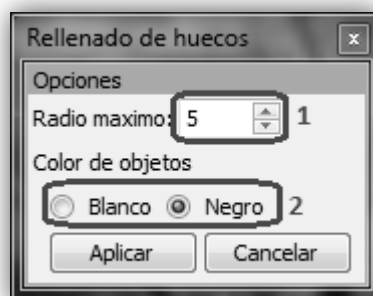


Fig 3.6 Ventana para los parámetros del filtro de rellenado de huecos.

- 1: Radio máximo de los huecos que se pueden rellenar.
- 2: Color de los elementos a considerar como objetos y con el cual se rellenarán los huecos dentro y en los bordes de los objetos.

3.1.7 Interfaz de análisis

Los resultados de realizar el procesamiento digital de imágenes se muestran en la ventana que se puede observar en la figura 3.7.

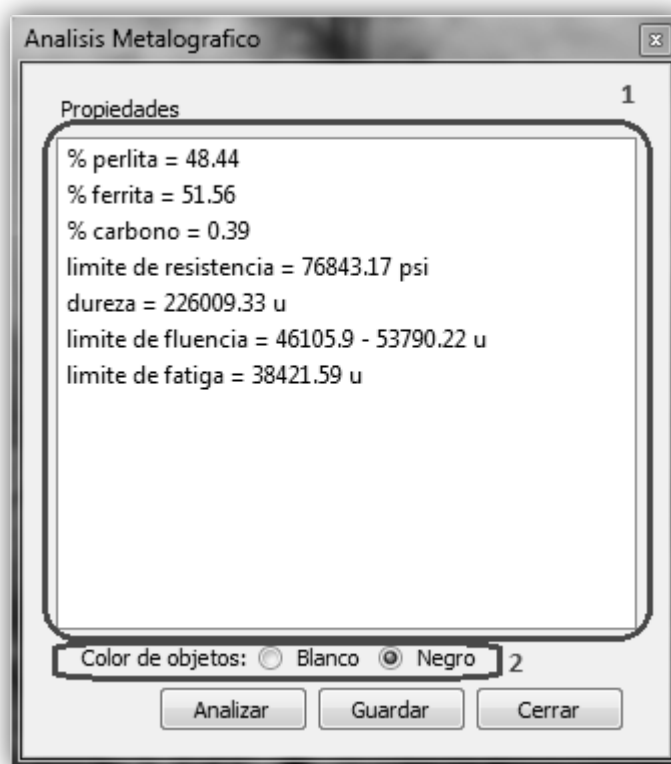


Fig 3.7 Ventana que muestra los resultados.

- 1: Vista donde se muestran los resultados del análisis
- 2: Color de los elementos a considerar como objetos (como perlita) en el análisis.

Botón Guardar: Abre un cuadro de dialogo para guardar un archivo texto con los datos de la imagen, los filtros aplicados a esta y los resultados del análisis.

Por último las figuras 3.8 y 3.9 muestran la ventana principal de la aplicación con las ventanas flotantes, acopladas a la derecha de la misma.

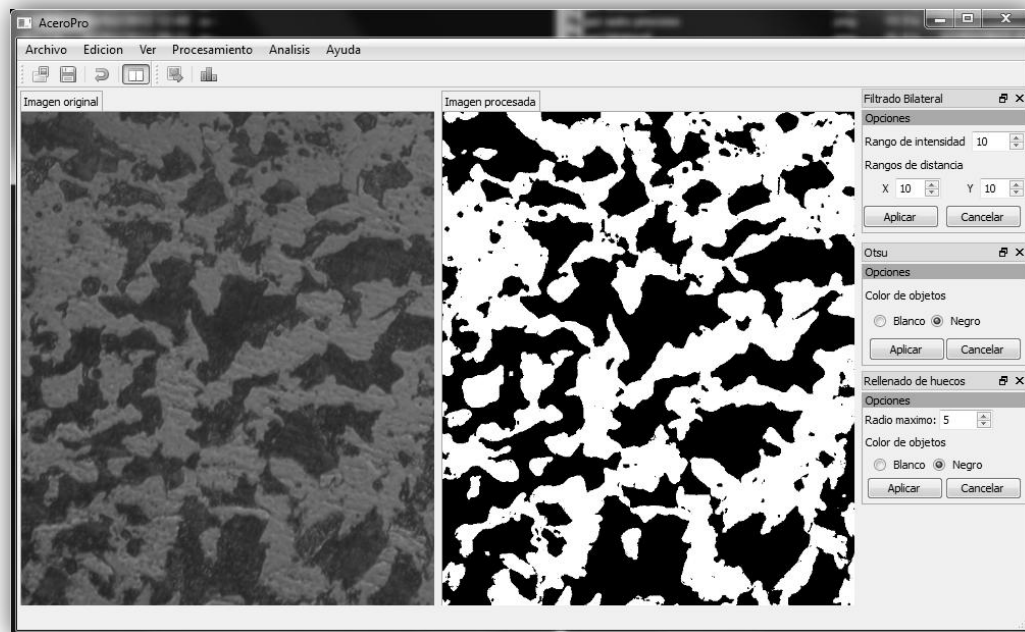


Fig. 3.8 Ventana principal con las ventanas flotantes acopladas a la derecha.

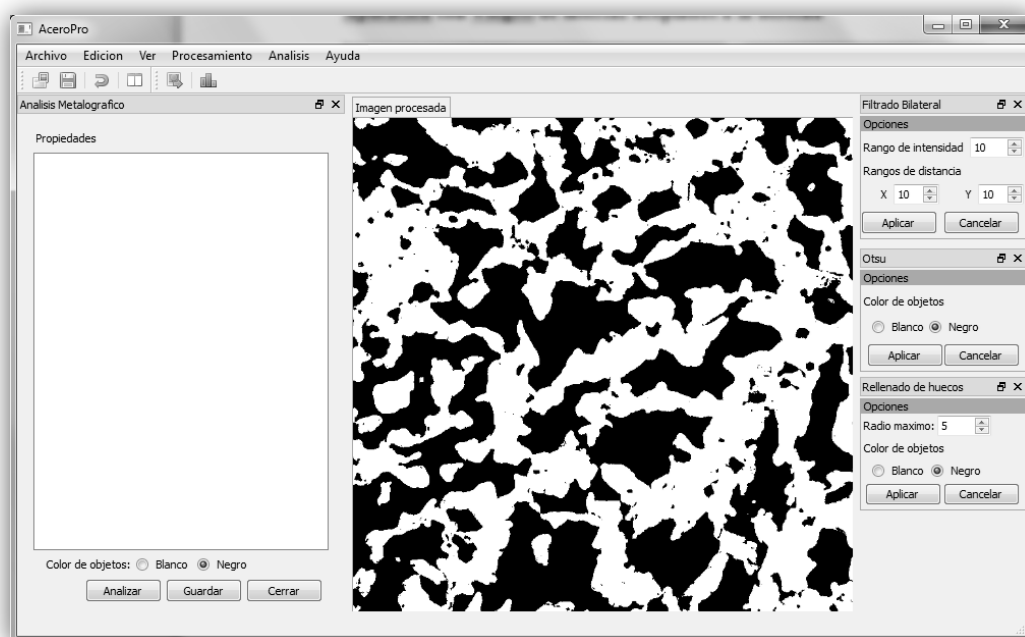


Fig. 3.9 Ventana principal con un solo visor y la ventana de los resultados.

En la figura 3.8 se muestran los dos visores, el de la imagen original (izquierda) y el de la imagen procesada (derecha). LA figura 3.9 muestra un solo visor y además la ventana de los resultados.

3.2 Evaluación de la aplicación

Para realizar la evaluación del sistema se debe verificar que el sistema está correctamente construido y que efectivamente es el producto que satisface los requerimientos. A continuación se mostrará la evaluación realizada a la aplicación implementada, donde se tiene en cuenta para la comparación otra herramienta desarrollada para el procesamiento digital de imágenes, *ImageJ* (ver epígrafe 1.3.1 Aplicaciones disponibles para el procesamiento de imágenes), además que se cuenta con la clasificación de referencia por las normas AISI para los porcentos de carbono en los diferentes tipos de acero. Para ello se detalla el caso de estudio diseñado, así como los principales resultados de la verificación y validación.

3.2.1 Definición del caso de estudio para la aplicación del procesamiento digital de imágenes a través de AceroPro

Para conformar el caso de estudio se tomó una muestra de 29 imágenes correspondientes a probetas de tres aceros diferentes, de tipo 1045, 1035 y 1015 por la norma AISI. Estas imágenes se dividen en 10 series. La tabla 3.1 presenta los detalles de la muestra.

En las dos últimas columnas de la tabla se muestra el rango de valores en que pueden oscilar los porcentos de carbono presentes en las muestras de acero correspondiente.

3.2.2 Diseño y aplicación de los experimentos

Verificar los resultados pretende asegurarse que el sistema sea consistente y correcto en cuanto a sintaxis. La estrategia que se ha seguido es comparar los resultados obtenidos con la aplicación *ImageJ*. Las muestras se procesaron en las dos aplicaciones (*ImageJ* y *AceroPro*), siguiendo el mismo procedimiento para cada una de las imágenes. Cada aplicación estimó un porcentaje de carbono para cada imagen de acero procesada, al

comparar estos resultados con el rango de porcentos que aparece en la tabla 3.1 se puede apreciar la efectividad de cada uno de estos, mostrándose estos resultados en la tabla 3.2.

Tabla 3.1 Detalles de la muestra tomada como caso de estudio

<i>Serie</i>	<i>Aumento</i>	<i>Tipo de Acero</i>	<i>Norma AISI</i>	
			<i>% carbono mín</i>	<i>% carbono máx</i>
B	100X	AISI 1035	0.32	0.38
B	200X	AISI 1035	0.32	0.38
B	400X	AISI 1035	0.32	0.38
D	100X	AISI 1045	0.43	0.5
D	200X	AISI 1045	0.43	0.5
D	400X	AISI 1045	0.43	0.5
F	100X	AISI 1035	0.32	0.38
F	200X	AISI 1035	0.32	0.38
F	400X	AISI 1035	0.32	0.38
J	100X	AISI 1045	0.43	0.5
J	200X	AISI 1045	0.43	0.5
J	400X	AISI 1045	0.43	0.5
M	100X	AISI 1045	0.43	0.5
M	200X	AISI 1045	0.43	0.5
M	400X	AISI 1045	0.43	0.5
N	100X	AISI 1035	0.32	0.38
N	200X	AISI 1035	0.32	0.38
N	400X	AISI 1035	0.32	0.38
P	100X	AISI 1045	0.43	0.5
P	200X	AISI 1045	0.43	0.5
P	400X	AISI 1045	0.43	0.5
S	100X	AISI 1035	0.32	0.38
S	200X	AISI 1035	0.32	0.38
S	400X	AISI 1035	0.32	0.38
U	100X	AISI 1045	0.43	0.5
U	200X	AISI 1045	0.43	0.5
U	400X	AISI 1045	0.43	0.5
Z	200X	AISI 1015	0.13	0.18
Z	400X	AISI 1015	0.13	0.18

Con estos resultados se realizaron pruebas estadísticas, específicamente la prueba de McNemar, la cual funciona bien con variables dicotómicas (1: el porciento está en el rango definido por la norma AISI; 0: en caso contrario) y un mínimo de diez casos de muestra.

Al interpretar los resultados se consideró:

- altamente significativa, una significación menor que 0.01
- significativo, un resultado de significación menor que 0.05 y mayor que 0.01
- medianamente significativo, un resultado menor que 0.1 y mayor que 0.05
- no significativo, un resultado mayor que 0.1

Tabla 3.2 Resultados de las pruebas realizadas con ImageJ y AceroPro.

<i>Serie</i>	<i>Norma AISI</i>		<i>ImageJ</i>	<i>AceroPro</i>
	<i>% carbono mín</i>	<i>% carbono máx</i>	<i>%carbono</i>	<i>% carbono</i>
B	0.32	0.38	0.33	0.35
B	0.32	0.38	0.38	0.37
B	0.32	0.38	0.36	0.37
D	0.43	0.5	0.46	0.45
D	0.43	0.5	0.47	0.46
D	0.43	0.5	0.52	0.48
F	0.32	0.38	0.36	0.36
F	0.32	0.38	0.39	0.39
F	0.32	0.38	0.37	0.38
J	0.43	0.5	0.38	0.41
J	0.43	0.5	0.34	0.35
J	0.43	0.5	0.37	0.35
M	0.43	0.5	0.53	0.45
M	0.43	0.5	0.5	0.45
M	0.43	0.5	0.52	0.47
N	0.32	0.38	0.37	0.38
N	0.32	0.38	0.36	0.37
N	0.32	0.38	0.36	0.36
P	0.43	0.5	0.38	0.39
P	0.43	0.5	0.41	0.4
P	0.43	0.5	0.41	0.4
S	0.32	0.38	0.33	0.35
S	0.32	0.38	0.34	0.35
S	0.32	0.38	0.33	0.34
U	0.43	0.5	0.42	0.45
U	0.43	0.5	0.43	0.45
U	0.43	0.5	0.43	0.43
Z	0.13	0.18	0.16	0.16
Z	0.13	0.18	0.16	0.15

En las figuras 3.10 y 3.11 se muestran los resultados del test, realizado con ayuda del software estadístico SPSS 13.

ImageJ & AceroPro

ImageJ	AceroPro	
	0	1
0	7	4
1	0	18

Fig. 3.10 Tabla de correlación cruzada obtenida por el SPSS.

Test Statistics^b

	ImageJ & AceroPro
N	29
Exact Sig. (2-tailed)	.125 ^a
Exact Sig. (1-tailed)	.063
Point Probability	.063

a. Binomial distribution used.

b. McNemar Test

Fig 3.11 Resultado del test de McNemar.

Al analizar los resultados de la prueba estadística se puede apreciar que existen diferencias medianamente significativas entre las dos aplicaciones, siendo AceroPro quien presenta los mejores resultados. Las diferencias medianamente significativas se expresan como un valor de la significación entre 0.05 y 0.1.

3.3 Conclusiones parciales

La interfaz de usuario que presenta la aplicación es amigable y fácil de usar por personas no especialistas en el área de la informática, ciencias de los materiales y procesamiento digital de imágenes. Todas sus funcionalidades han sido descritas en detalles.

El caso de estudio definido permitió demostrar la calidad de los resultados obtenidos con la aplicación.

Al realizar pruebas estadísticas para comparar los resultados de la aplicación implementada con los obtenidos a partir de otra aplicación, las diferencias existentes entre estos resultaron ser medianamente significativas, a favor de la aplicación implementada.

CONCLUSIONES

Como resultado de este trabajo de diploma se desarrolló un software para el procesamiento digital de imágenes obtenidas a través de un microscopio óptico metalográfico, con la finalidad de caracterizar metalográficamente aceros al carbono, cumpliéndose de esta forma el objetivo general planteado, ya que:

1. Del estudio bibliográfico realizado se seleccionaron herramientas competitivas para el desarrollo de aplicaciones de procesamiento digital de imágenes.
2. Con las herramientas seleccionadas se diseñó e implementó una aplicación para la caracterización de imágenes de aceros al carbono. Esta aplicación cumple con los requerimientos planteados por los especialistas en el área de la Ciencia de los Materiales.
3. La aplicación permite la reproducibilidad del proceso de análisis de imágenes, a través de la posibilidad de guardar todos los parámetros del procesamiento realizado.
4. La interfaz de usuario que presenta la aplicación es amigable y fácil de usar por personas no especialistas en el área de la informática, ciencias de los materiales y procesamiento digital de imágenes.
5. Al realizar pruebas estadísticas para comparar los resultados de la aplicación implementada con los obtenidos a partir de la aplicación *ImageJ*, las diferencias existentes entre estos son medianamente significativas, a favor de la aplicación implementada.

RECOMENDACIONES

1. Extender la aplicación a otras funcionalidades del análisis de las micrografías de aceros, como la determinación del tamaño de granos.
2. Realizar un estudio de técnicas de Inteligencia Artificial para comprobar la factibilidad de aplicarlas en la utilización de la optimización de los parámetros del procesamiento automático.
3. Pruebas preliminares muestran que el procedimiento utilizado en la aplicación puede ser empleado para el análisis de imágenes de hormigón. Por lo que se recomienda valorar las posibilidades de extender las funcionalidades de la aplicación con este fin.

REFERENCIAS BIBLIOGRÁFICAS

- ACHARYA, T. & RAY, A. K. 2005. Image processing: principles and applications. John Wiley & Sons.
- BRICK, R. M. & PHILLIPS, A. 1949. *Structure and Properties of Alloys*, McGraw-Hill.
- BROWN, L. G. 1992. A survey of image registration techniques. *ACM Computing Surveys*.
- CALLISTER W 1999. *Introducción a la Ciencia e Ingeniería de los Materiales*.
- DEHOFF, R. T. & RHINES, F. N. 1968. *Quantitative Microscopy*, USA, McGraw-Hill Book Company.
- FAJARDO, J., ABUCHAR, J., URECHE, J. & ALARCON, J. 2007. CARACTERIZACION MECÁNICA Y MICRO-ESTRUCTURAL DEL ACERO ASTM A 131A SOMETIDO A CARGAS EXPLOSIVAS. *Scientia et Technica Año XIII, Colombia, Septiembre de 2007* [Online].
- FERREIRA, T. & RASBAND, W. 2011. ImageJ User Guide.
- FIJI. *Fiji is Just ImageJ* [Online]. Available: <http://fiji.sc/wiki/index.php/Fiji> 2012].
- GROSSBERG, S. 1984. Neural dynamics of brightness perception: Features, boundaries, diffusion, and resonance. *Perception and Psychophysics*, 36.
- HUANG, Y. D. & FROYEN, L. 2001. *Quantitative analysis of microstructure in metals with computer assistance* [Online]. 2012].
- IBAÑEZ, L., SCHROEDER, W., NG, L. & CATES, J. 2005. The ITK Software Guide. In: 2 (ed.).
- JÄHNE, D. B. 2005. Digital Image Processing. 6 ed.: Springer.
- KITWARE. *ITK Insight Toolkit* [Online]. Available: www.kitware.com.
- PERONA, P. & MALIK, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions of Pattern Analysis Machine Intelligence*.
- QT-PROJECT. *Qt* [Online]. Available: <http://qt-project.org> 2012].
- ROJAS, O. G. R., MARTÍNEZ-JIMÉNEZ, B. L. & SCOTT, A. D. 2011. Técnicas de Procesamiento de Imagen Digital para la Caracterización de Materiales.
- SCHROEDER, W., MARTIN, K. & LORENSEN, B. 2006. *Visualization Toolkit. An object-oriented approach to 3D Graphics*, Kitware.
- SOURCEFORGE. *OpenCV* [Online]. Available: <http://opencvlibrary.sourceforge.net/> 2012].
- SWAROOP, C. H. 2005. A Byte of Python.

- TOMASI, C. & MANDUCHI, R. 1998. Bilateral Filtering for Gray and Color Images. *Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India.*
- UNDERWOOD, E. E. 1970. *Quantitative Stereology.* , USA, Addison-Wesley Publishing Company.
- V, G. & A, V. 2004. Affordable image analysis using NIH Image/ImageJ. *Indian J Cancer.*
- W, B. & M, B. 2007. Digital Image Processing: An Algorithmic Approach Using Java. Springer.
- WOLFRAM. *New in Wolfram Mathematica 7. Built-in Digital Image Processing & Analysis* [Online]. Available: www.wolfram.com/products/mathematica 2012].
- YOO, T. S., NEUMANN, U., FUCHS, H., PIZER, S. M., CULLIP, T., RHOADES, J. & WHITAKER, R. T. 1991a. Direct visualization of volume data. *IEEE Computer Graphics and Applications.*
- YOO, T. S., PIZER, S. M., FUCHS, H., CULLIP, T., RHOADES, J. & WHITAKER, R. T. 1991b. Achieving direct volume visualization with interactive semantic region selection. *Information Processing in Medical Images.*