



APLICACIÓN PARA PROCESAMIENTO DE IMÁGENES METALGRÁFICAS DESDE PME3 APPLICATION FOR PROCESSING OF IMAGES METALLOGRAPHIC FROM PME3

Miguel R. Pérez P.¹

Henry A. Hernández M.²

Geimar N. Güiza F.³

Fecha de envió: Marzo de 2011

Fecha de recepción: Mayo de 2011

Fecha de aceptación: Junio de 2011

Resumen:

La información en forma de datos y medidas, suministrada por imágenes metalográficas, es importante en la caracterización de materiales. La eficiencia en el tratamiento de tal información, es crucial para una exitosa gestión de laboratorio en ingeniería. Este artículo, muestra el análisis, desarrollo, e implementación de una aplicación computacional, desarrollada enteramente en Java (netbeans versión 6.8 “GNU”), que captura, edita (bordes, suavizados, realces, distancias, inversión de colores, escala de grises, brillo, zoom móvil y fijo), y guarda imágenes metalográficas, proporcionadas por el microscopio OLYMPUS PME3®.

¹ Ingeniero en Control e Instrumentación Electrónica. Docente Universidad Distrital Francisco José De Caldas. Correo electrónico: mperezp@udistrital.edu.co.

² Tecnólogo en electrónica, Universidad Distrital Francisco José de Caldas. Correo electrónico: haguit2@hotmail.com

³ Tecnólogo en electrónica, universidad Distrital Francisco José de Caldas. Correo electrónico: fénix_720@hotmail.com

Palabras claves:

Fotografía digital, Adquisición, S-video, Interface, Metalografía, PME3.

Abstract:

The information in the form of data and measurements supplied by metallographic images is important in the characterization of materials. Efficiency in the treatment of such information is crucial for successful engineering laboratory management. This article shows the analysis, development,

and implementation of a computer application, developed entirely in Java (version 6.8 netbeans “GNU”), capturing, editing (borders, softened, enhancements, distance, color inversion, grayscale, brightness, zoom mobile and fixed), and save images provided by the metallographic microscope OLYMPUS PME3®.

Key words:

Digital Photography, Purchase, S-video, Interface, Metallography, PME3.

1. Introducción

La Universidad Distrital Francisco José de Caldas en su Facultad Tecnológica, ubicada en la localidad de Ciudad Bolívar (Bogotá, Colombia), cuenta con un microscopio OLYMPUS PME3, equipado con un módulo RS-232⁴ para abrir o cerrar los juegos de lentes que componen la unidad en la cual se colocan las cámaras para visualizar la muestra. También cuenta con un juego de lentes, los cuales pueden variarse para dar mayor o menor aumento a la zona de la muestra a analizar (su escala mínima es 50x y la máxima 1000x), y con un juego de palancas que permiten el paso de la luz, para visualizar la zona de la muestra iluminada o sin iluminación. Posee, además, una cámara de video - en desuso, ya que el medio por el cual se comunica con el computador es tecnológicamente obsoleto -.

Además de la obsolescencia del sistema de comunicación entre la cámara de video y el computador, no se cuenta con disponibilidad para almacenar las imágenes en medio magnético, la única forma de adquirir una muestra es colocando una cámara digital en el lente del visor o utilizar la cámara de rollo, ocasionando pérdida de información de la muestra, baja calidad en la fotografía, un gasto adicional en la compra del rollo y pérdida de tiempo en la toma de las muestras.

La anterior problemática es abordada en la investigación, entendiéndose que se debe recondicionar la interface de comunicación de la cámara con un computador y debe diseñarse una aplicación computacional que permita el manejo de las imágenes adquiridas. Esta alternativa, debe considerar la edición de las imágenes, proceso que se

aborda desde la aplicación de filtros pasa bajos, pasa altos y filtro media, permitiendo suavizar la imagen, mejorar su contraste, nitidez y eliminar varios tipos de ruido (sal, pimienta, gaussiano, entre otros). Las imágenes modificadas tienen que estar disponibles para el usuario del microscopio, por lo que se deben almacenar en una base de imágenes.

Finalmente, la disponibilidad de la galería de imágenes debe permitir, en trabajos futuros, hacer comparaciones entre muestras vistas por el microscopio y adquiridas por la aplicación así como llevar un control documental para mejorar sustancialmente la gestión del laboratorio.

El contenido de este escrito se dispone así: estado del arte en procesamiento digital, modelo matemático, características de Hardware, características de software, perspectivas, resultados y conclusiones.

2. Estado del Arte

El procesamiento digital de imágenes es una disciplina que se ha desarrollado desde la década de los sesenta, inicialmente por la NASA, buscando aplicar algoritmos sobre una imagen para reconocer datos implícitos, lo cual permite resaltar, en particular, patrones de la micro estructura de materiales metalográficos. Con la explosión de microprocesadores, como DSP (digital signal processor), de bajo costo; se impulsó el procesamiento de imágenes con altas tasas de eficiencia. Por lo anterior, hoy en día se dispone de programas dedicados exclusivamente para el procesamiento de imágenes, la gran mayoría, sin embargo, no de código abierto, lo que restringe críticamente su uso.

⁴ RS-232: Es una interfaz que designa una norma para el intercambio serie de datos binarios entre: Un DTE (Equipo terminal de datos) y un DCE (Data Communication Equipment, Equipo de Comunicación de datos).

No obstante, con el desarrollo del lenguaje de programación Java⁵, caracterizado por su portabilidad y accesibilidad, se han venido desarrollando múltiples aplicaciones en diferentes campos, entre ellos el procesamiento de imágenes. Existe, entre muchos, un software llamado “Títtere” [1], herramienta open source bajo licencia GPL⁶ destinado a la transformación de imágenes digitales, desarrollado en su totalidad en Java, siendo un programa didáctico para el procesamiento de imágenes, donde se encuentran algoritmos ya diseñados, pero para ser aplicados genéricamente. También podemos encontrar otra que se denomina “Paint.Net” [2], Hace gran parte de lo que hace Photoshop, y lo hace más sencillo, más rápido, más ligero y es software libre. Además, funciona bien en máquinas con pocos recursos, como los netbooks.

5 Java: Es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. Entre diciembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre.

6 Licencia GPL: Está orientada principalmente a proteger la libre distribución, modificación y uso de software. Declarando que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

En el campo de la metalografía, hay empresas que se circunscriben a: México (SyMIP®, SIMEX®, TOTVS®, PETROX®) y Colombia (VIDCOL®); que producen software propietario para gestión de imágenes, pero no particularmente en microscopios metalográficos. Existe el Software de Análisis Metalográfico X PLUS – REMET®, pero limitado a microscopios Carl Zeiss®; o programas de LECO® como IA44, o, PAX-IT, completamente en inglés. El software Scentis®, es una opción para adquisición, tratamiento y gestión de imágenes metalográficas, sin embargo, además de ser propietario, se dispone sólo para cámaras digitales QImaging o video cámaras, generalmente de alto costo. En el campo de la didáctica de la metalografía, hay simuladores de microscopios, ver [3], que no obstante dejan sin contacto con el aparato a los estudiantes, por efecto de los altos costos de mantenimiento.

3. Procesamiento Digital de Imágenes

Es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

3.1. Proceso de Filtrado

Es el conjunto de técnicas englobadas dentro del preprocesamiento de imágenes 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.

Los principales objetivos que se persiguen con la aplicación de filtros son:

- Suavizar la imagen: reducir la cantidad de variaciones de intensidad entre píxeles vecinos.
- Eliminar ruido: eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión.
- Realzar bordes: destacar los bordes que se localizan en una imagen.
- Detectar bordes: detectar los píxeles donde se produce un cambio brusco en la función intensidad.

Por tanto, se consideran los filtros como operaciones que se aplican a los píxeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella.

El proceso de filtrado puede llevarse a cabo sobre los dominios de frecuencia y/o espacio.

Filtrado en el dominio de la frecuencia

Los filtros de frecuencia procesan una imagen trabajando sobre el dominio de la frecuencia en la Transformada de Fourier de la imagen. Para ello, ésta se modifica siguiendo el Teorema de la Convolución correspondiente:

1. se aplica la Transformada de Fourier,
2. se multiplica posteriormente por la función del filtro que ha sido escogido,
3. para concluir re-transformándola al dominio espacial empleando la Transformada Inversa de Fourier.

Teorema de la Convulación (frecuencia):

$$G(u,v) = F(u,v) * H(u,v)$$

$F(u,v)$: transformada de Fourier de la imagen original

$H(u,v)$: filtro atenuador de frecuencias

Como la multiplicación en el espacio de Fourier es idéntica a la convolución en el dominio espacial, todos los filtros podrían, en teoría, ser implementados como un filtro espacial.

4. Modelo Matemático

Generalmente, en las imágenes reales existen relaciones entre los valores asociados a píxeles vecinos. Para poder modelar este tipo de interacciones desde el punto de vista estadístico, se han propuesto numerosos

modelos, la mayoría de los cuales fueron a su vez utilizados en la Física ya que puede establecerse una analogía entre partículas en un sistema físico con píxeles en una imagen.

Una imagen digital se puede definir como una colección de puntos definidos en una región C , las regiones pueden en general tener cualquier forma y existe una función $I(x, y, z)$, donde alguno de los parámetros es función continua de los otros dos, que nos permite recorrer los puntos que componen la colección. Digamos que $z = h(x, y)$, entonces una representación de la imagen digital es, ver [4],[5],[6] o [7]:

$$I(x, y, z) = I(x, y, h(x, y)) = J(x, y) \quad (1)$$

Donde las propiedades de cada punto o *pixel* que conforma la imagen están contenidas en $J(x, y)$.

Estas consideran por ejemplo: el color, el brillo, la reflectividad, etc.

Se define una transformación T de la imagen $J(x,y)$ como una función tal que,

$$C: T [J(x, y)] = J'(x, y), \quad (2)$$

Los efectos de T sobre J pueden ser diversos, entre otros podemos citar;

- Modificación del dominio C .
- Cambio en una o varias de sus propiedades.
- Cambio en la posición de los puntos.
- Cambio en la representación de J .

Si T actúa sobre un punto diremos que la transformación es puntual, si actúa sobre

una región será zonal y si actúa sobre toda la imagen se dice que es global.

Diremos que T es una transformación de mapeo isomórfico de un pixel (uno a uno) si cada pixel de la imagen de origen nos genera un nuevo pixel en otro espacio.

Muchas de éstas transformaciones se pueden definir a través de la relación:

$$F(u, v) = \langle g(x, y) | J(x, y) \rangle \quad (3)$$

Donde $g(x, y)$ se denomina el *núcleo* o *kernel* de la transformación, $J(x, y)$ es un punto de la imagen y $\langle | \rangle$ define el mecanismo de transformación.

Una clase bastante grande de éstas transformaciones se realiza mediante un *producto escalar*, si la magnitud de $f(x, y)$ tiene magnitud uno la transformación se denomina *unitaria*.

Algunas formas usuales del producto escalar están definidas por una integrales definidas o una sumatorias y el kernel de la transformación es típicamente una base ortogonal y puede ser utilizada para la reconstrucción de la imagen original. Si el producto escalar de dos funciones es cero, se dice que éstas son ortogonales.

4.1. Modelo de Ising

Los campos aleatorios, [8], proveen una estructura matemática para modelar imágenes capturando las dependencias entre píxeles. Una de las propiedades usualmente requeridas para un modelo matemático de imágenes es proporcionar probabilidades de los valores de los píxeles condicionados a los píxeles vecinos.

Este modelo tuvo su origen en la física para explicar el comportamiento de materiales ferro- magnéticos. Dado un sistema de n variables binarias o spins s_i , que pueden tomar dos estados opuestos $s_i = -1$ o $s_i = +1$, la energía del sistema E esta dada por los estados de los spins, de la siguiente manera:

$$E_i \{S_i\} = -\sum_{ij} \epsilon_{ij} S_i S_j - H \sum_{i=1} S_i \quad (4)$$

Donde el símbolo $\langle ij \rangle$ indica pares de vecinos de primer orden (o vecinos más cercanos), ϵ_{ij} indica el potencial de interacción entre los spins S_i y S_j , y H es un posible campo externo que induce a los spins a polarizarse, ya sea positivamente $H > 0$ o negativamente $H < 0$.

Este modelo se calcula a partir de la Formula de Gibbs, ver [9]:

$$P\{S_i\} = \frac{1}{z} e^{\left(\left(\frac{-1}{kT}\right)E_i\{S_i\}\right)} \quad (5)$$

Donde z es una constante de normalización conocida como función de partición en mecánica estadística, k es la constante de Boltzmann y T es la temperatura del sistema. Se observa que, para el caso de ausencia de campo externo, y considerando interacciones homogéneas de primeros vecinos ($\epsilon_{ij} = 1$ si s_i y s_j son vecinos de primer orden), la probabilidad de una configuración esta dada por:

$$P\{S_i\} = \frac{1}{z} e^{\left(\beta \sum_{ij} S_i S_j\right)} \quad (6)$$

Con $\beta = \frac{1}{kT}$ siendo el único parámetro que define el modelo y que es inversamente proporcional a la temperatura del sistema. Este sencillo modelo permite representar imágenes en blanco y negro, donde el parámetro β regula las interacciones entre píxeles.

4.2. Expresión General de un Filtro. [10], [11], o, [12].

Hay muchas formas de representar un filtro para imágenes. Una opción es hacerlo en el dominio espacial. Su fundamentación es la convolución discreta. Este operador transforma dos funciones $x[n]$ y $h[n]$ en una tercera función:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] * h[n-k] \quad (7)$$

La función $x[n]$ es definida para la creación del filtro como una matriz $m \times n$, la cual contiene la imagen a ser tratada, la función $h[n]$ es una matriz $n \times m$ (kernel), cuyos valores son asignados dependiendo de los vecinos de la matriz, el tamaño de la matriz designa qué tantos vecinos tomar para luego hacer un promedio entre todos ellos. La operación resultante es la imagen filtrada.

Otra opción, es representar el filtro en función de w (frecuencia digital), en función de z y en función de n (número de muestra). Todas son equivalentes. Como regla general se suele dejar el término $a_0=1$.

Si se expresa en función de z y en forma de fracción:

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} \quad (8)$$

Y en dominio de n :

$$y(n) = \sum_{k=0}^N b_k x(n-k) - \sum_{k=1}^M a_k y(n-k) \quad (9)$$

Los coeficientes son la a_k y el b_k y son los que definen el filtro, por lo tanto el diseño consiste en calcularlos.

5. Características del Hardware.

El PME3 (Ver Figura 1) tiene un juego de cinco lentes (Ver Figura 2), que permiten

Figura 1. PME3



enfocar las muestras pues cuenta con diferentes aumentos con un zoom de 50x a 1000x. Para mover la muestra en los ejes (x,y) , dispone de un mango con dos diales para desplazar estas libremente.

Figura 2. Juego de lentes

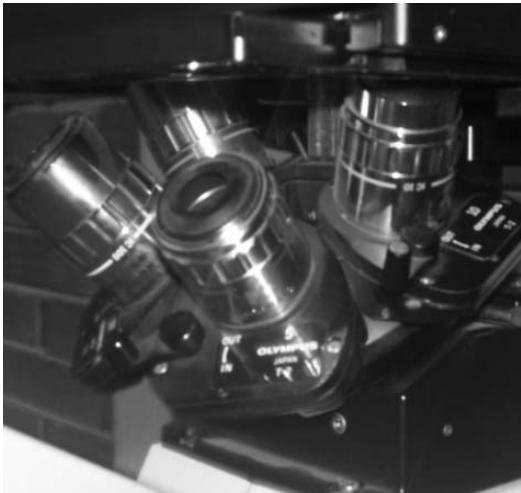


Figura 3. Cámara digital



Figura 4. Activador



Figura 5. Cámara de Rollo



El foco del lente de la cámara (Ver Figura 3) tiene un diámetro de 35 mm, permitiendo la visualización de una parte de las muestras, para visualizar toda la muestra, el PME3 tiene un lente de enfoque que permite hacer un barrido sobre la muestra para acentuar u omitir detalles.

Para activar o desactivar la adquisición de imágenes desde la video cámara o la cámara de rollo, cuenta con un modulo RS-232 (Ver Figura 4) que permite abrir o cerrar el juego de lentes para visualizar la imagen (en el caso de la cámara fotográfica también permite el disparo del flash y avance del rollo (Ver Figura 5)).

La cámara de video se alimenta a 12Vdc-650-mA, la parte delantera tiene un sensor CCD⁷ y un orificio roscado para su conexión con el PME3, en la parte trasera tiene la salida de video (S-Video o Video Compuesto), la cual puede ser visualizada en diferentes dispositivos (TV, VHS, DVD, Pc's, entre otros).

5.1 Comunicación del Pme3 con el Computador

Para tomar las muestras de los materiales, el microscopio utiliza una cámara que proporciona una señal de S-Video [S-Video tiene más calidad que el video compuesto⁸, ya que la pantalla dispone por separado de la información de brillo y la de color, mientras que en el video compuesto se encuentran juntas, [13]. Esta separación hace que el cable S-Video tenga más ancho de banda para la luminancia⁹ y consiga más trabajo efectivo del decodificador de crominancia¹⁰. Con una salida de 4 pines, muchos computadores en el comercio no poseen una entrada para este conector, para poder comunicar el computador y el microscopio fue necesario usar una capturadora de video compatible con una señal de súper video mostrada en la figura 6.

7 CCD: (dispositivo de carga acoplada) Es un circuito integrado que contiene un número determinado de condensadores enlazados o acoplados. Es un sensor con diminutas células fotoeléctricas que registran la imagen. Desde allí la imagen es procesada por la cámara y registrada en la tarjeta de memoria.

8 Video Compuesto: Es una señal de video analógica que se utiliza en la producción de televisión y en los equipos audiovisuales domésticos. Esta señal eléctrica es una señal compleja en la que se codifica la imagen en sus diferentes componentes de luz y color añadiendo los sincronismos necesarios para su posterior reconstrucción.

9 Luminancia: Es el componente de la señal de video que contiene las informaciones de la luz o brillo.

10 Crominancia: Es el componente de la señal de video que contiene las informaciones del color.

Figura 6. Capturador de S-video.



Dicho dispositivo es externo, y permite entradas de video/audio por medio del puerto USB del computador.

6. Características del Software.

La aplicación fue desarrollada enteramente en Java, por medio de la interfaz compuesta por: adquisición, base de imágenes y el editor de imágenes. En la adquisición se puede

visualizar la imagen en tiempo real, para ser fotografiada en el momento indicado; la base de imágenes es un apartado donde son almacenadas todas las imágenes de apoyo para el estudio y análisis por parte del usuario. El editor permite modificar la imagen.

En la figura 7 se muestra el procesamiento del programa escrito en Java, para su interacción con el hardware (Capturador de Video), al lado izquierdo se encuentran los pasos para la compilación y al lado derecho, se encuentran los pasos para su ejecución:

6.1. Parámetros de Diseño.

En el desarrollo de la aplicación, se tuvo en cuenta el diseño arquitectónico, que se enfocó en el desarrollo de estructuras, para generar interfaces que facilitan el flujo de datos a lo largo del programa. Manejando este modelo se genera un programa con varios módulos, los cuales están enlazados entre

Figura 7 Procesamiento de algoritmos en Java

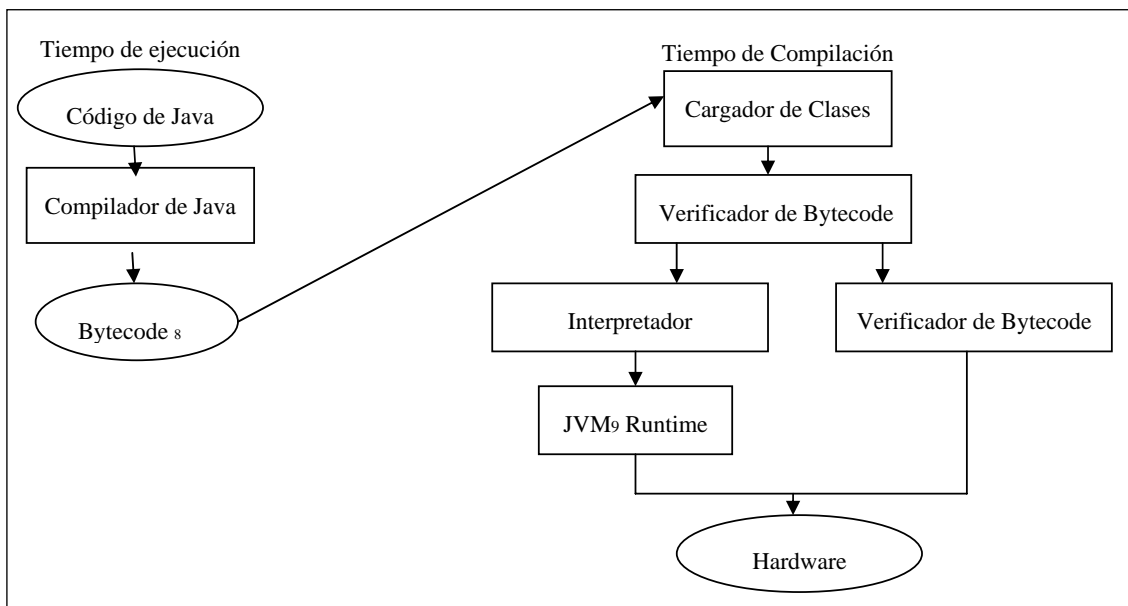
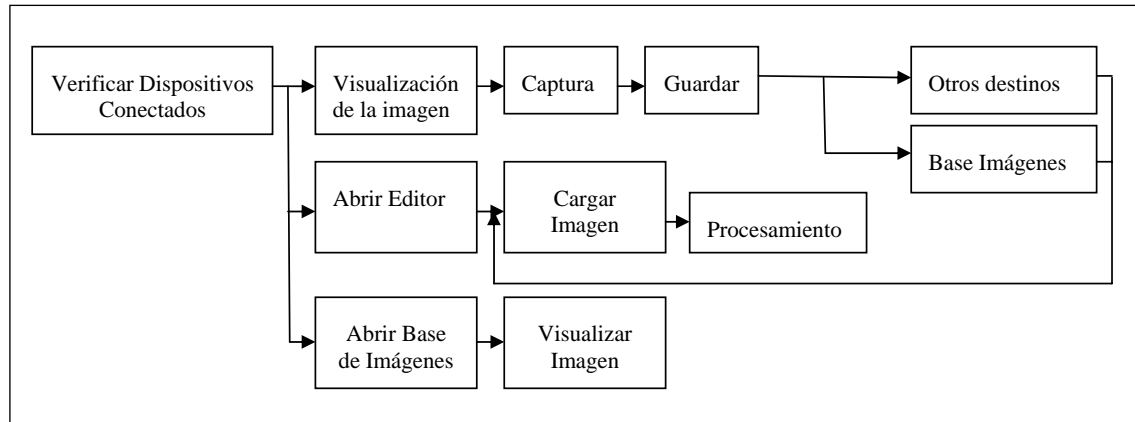


Figura 8. Funcionamiento de la aplicación.



sí para facilitar el manejo de la información (ver figura 8).

6.1.1. Diseño Arquitectónico

Su objetivo principal es desarrollar una estructura de programa modular y representar las relaciones de control entre los módulos. Además, el diseño arquitectónico mezcla la estructura de programas y la estructura de datos y define las interfaces que facilitan el flujo de datos a lo largo del programa, ver [14].

En el diagrama N°2 se muestra el funcionamiento de la aplicación desarrollada:

6.2. Interfaz Grafica de Usuario

Como medio de comunicación entre hombre-máquina se realizó una interfaz de usuario, diseñada para facilitar el manejo de la información en ambientes educativos, permitiendo al usuario capturar la imagen, ya sea en la base de imágenes o en una ubicación diferente.

En la figura 9 se observa la interfaz del programa cuando es inicializado, donde se pue-

de visualizar directamente la imagen que se está recibiendo de la cámara de video.

6.2.1 Captura de la imagen, [8], [10], [15]

Para la captura de la imagen, se trabajó en el software con el API JMF (*Java Media Framework*). La JMF es una librería que por defecto no se encuentra instalada en la plataforma netbeans, por lo que fue necesario incorporarla. Cargada la librería en el paquete JDK (*Java Development Kit*), ver

Figura 9. Ventana de inicio



[16],[17], permite la programación de tareas multimedia, es decir, que se puedan capturar, reproducir, transferir y codificar varios formatos, como también la *transmisión* en tiempo real de vídeo y audio a través de la red, mediante el protocolo *RTP (Real-time Transport Protocol)*.

Con el paquete JMF incorporado en Java, se implementa un algoritmo para la captura de la imagen proporcionada por la cámara de vídeo del microscopio; en el diagrama de flujo mostrado en la figura 10 se muestra el algoritmo desarrollado.

6.3. Base De Imágenes

La base de imágenes es el lugar en donde se encuentran almacenadas todas las imágenes necesarias para atender a las necesidades de los usuarios de forma directa. Para su realización, se tomaron fotos de muestras las cuales los usuarios tienen como guía para compararlas con otros tipos de muestras que posean la misma estructura; en la figura 11 se observa la ventana principal.

La base, cuenta con siete carpetas; aglomerantes, ferrosos, no ferrosos, pétreos, plásticos, maderas y una carpeta con el nombre de otros; por cada imagen almacenada en las carpetas, se genera un contador, el cual aumenta cada vez que una nueva imagen es almacenada, el contador es publicado en una tabla, donde están especificados los nombres de cada carpeta y la cantidad de imágenes contenidas por cada una.

6.4 Editor de Imágenes, [8], [10], [15].

Todas las imágenes almacenadas con extensiones JPG, PNG, BMP y GIF, pueden ser tratadas por el software en el editor de imágenes, para garantizar que el programa cargue los formatos indicados, se aplica

Figura 10. Captura de la imagen

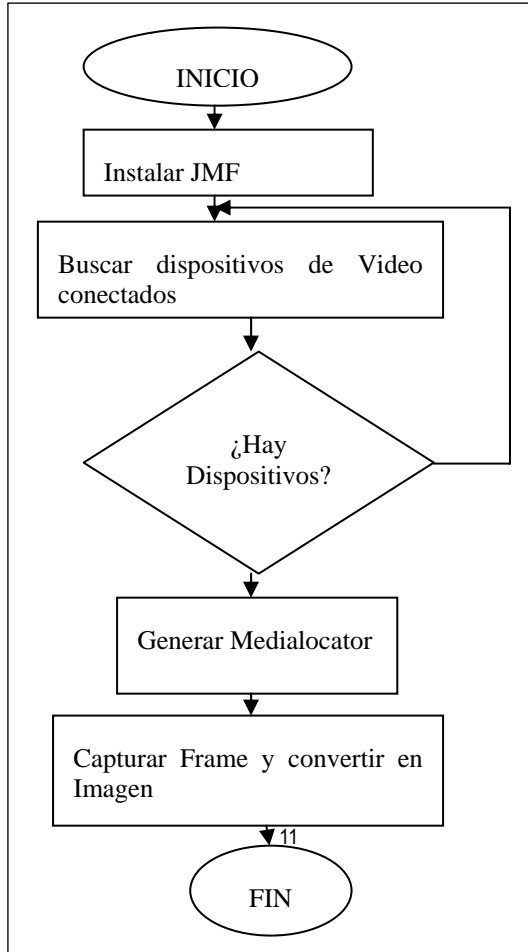
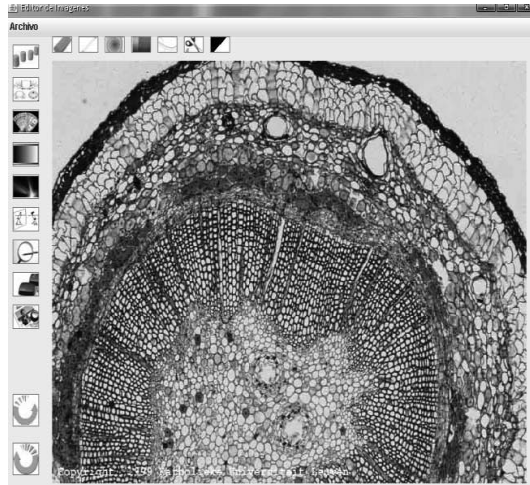


Figura 11. Base de imágenes.



11 Medialocator: describe la ubicación de los contenidos de los medios.

Figura 12. Editor de imágenes.



un filtro que solo abre los formatos de imagen mencionados, lo cual permite rechazar extensiones de texto, video, virus y otros que ocasionan el mal funcionamiento y bloqueo del programa, evitando problemas en el software al momento de abrir un archivo con una extensión diferente.

En la figura 12, se observa la ventana del editor de imágenes, la cual ofrece una barra de herramientas y barra de mascarar, donde el usuario puede modificar y tratar la imagen para obtener datos importantes como; distancias entre dos puntos, que tanto se parece la imagen tratada a la original.

En la barra de herramientas se encuentran las opciones de: borrado, lápiz, texto, color, medir entre un punto y otro de la imagen, recortar y comparar dos imágenes.

La barra de mascarar genera algoritmos que son aplicados a la imagen para dar: suavizado, bordes, escala de grises, brillo, espejo, zoom, deshacer, vista anterior y giros. Cada una de las opciones descritas se desarrolló de la siguiente forma:

- **Borrado y medición entre puntos.**

Para borrar una parte de la imagen el programa dibuja un cuadrado, el cual tiene cinco tamaños dependiendo del que sea escogido, este es pintado de color blanco, para ser aplicado sobre la imagen, lo que hace es utilizar la clase “Repaint” para dejar un rastro sobre el Canvas¹², dando la forma de borrado a la imagen.

Para realizar la medida de un punto a otro en la imagen, el software procesa la imagen como una matriz, donde cada número representa un píxel, para obtener la distancia entre un píxel a otro, el programa captura las coordenadas de los puntos a medir, por medio de la función mouseClicked, la cual permite capturar el evento del mouse al hacer clic sobre la imagen, teniendo la posición de los dos vectores, se hace una resta entre los dos vectores, se obtiene la magnitud de la diferencia, este se multiplica por el factor de escala del microscopio.

La respuesta se gradúa automáticamente dependiendo de la distancia y de la escala, permitiendo obtener las medidas en nanómetros, micrómetros e inclusive picómetros, mostrando sobre la imagen una recta la cual representa la distancia medida y en uno de sus extremos el resultado.

- **Comparación entre dos imágenes.**

Al ser cargada la imagen, la clase de Java que permite capturar la matriz de componentes RGB de la imagen a analizar, carga la matriz de la imagen de comparación. Con el proceso, realizado componente a componente, se verifica tamaño y, según la cantidad de aciertos, el programa muestra un resultado, que indica qué el grado de semejanza de una con respecto a la otra.

¹²Canvas: Elemento que permite la generación de gráficos en forma dinámica por medio de programación.

Figura 12.a imagen original.

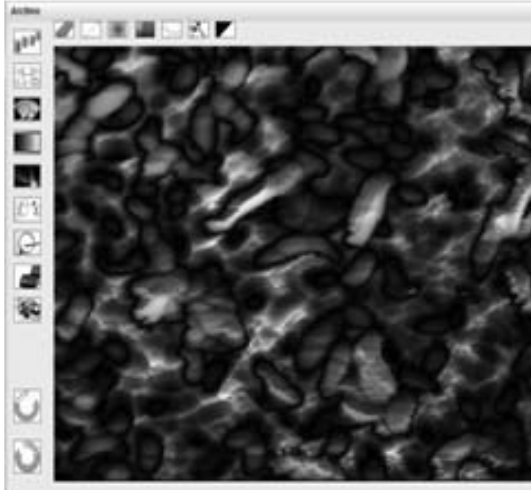
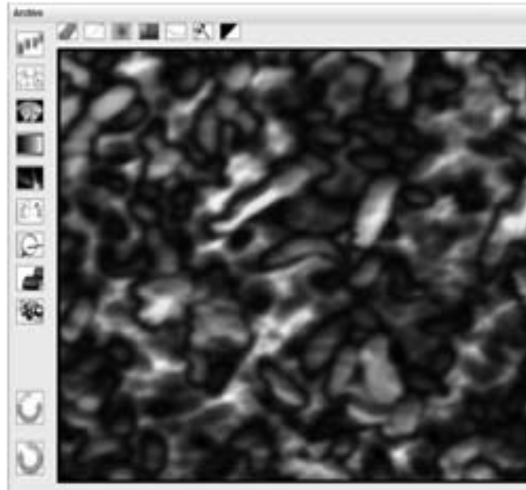


Figura 12.b Imagen suavizada.



- **Suavizamiento.**

Bajo la denominación general de filtros de suavizado se engloba un conjunto de ellos cuya característica común es que disminuye las diferencias de valores de los píxeles respecto a sus vecinos. La aplicación de estos filtros produce la degradación de la imagen, un efecto que se asemeja al desenfoque de la imagen debido a la difuminación de las líneas de contorno de los objetos.

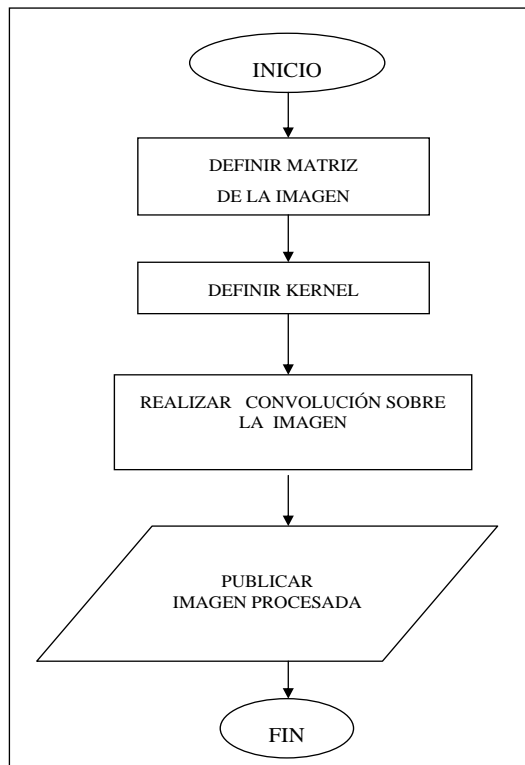
Cada operación de filtrado tiende a reducir el valor de gris promedio de la imagen, lo que le ha valido, también con el nombre de filtros pasa bajo (*low-pass*). Los tres principales filtros de suavizado son: filtro media, filtro gaussiano y filtro de la mediana, el software aplica a la imagen el filtro media ya que permite buenos suavizados como se muestra en la figura 12.b, la cual se le ha aplicado el filtro media, en la figura 12.a se muestra la imagen original, en la figura 13 se muestra la estructura para la programación del filtro y su aplicación.

- **Bordes**

Con la aplicación de filtros en el dominio espacial se trata la imagen, para ver com-

ponentes no visibles por el ojo humano, empleando filtros pasa altas, lo cual permitió intensificar detalles y bordes, mientras que

Figura 13. Procesamiento de imagen.



se atenúan las zonas de tonalidad uniforme. En la creación de filtros pasa altas, el programa utiliza la misma estructura que los filtros pasa bajos. Para realizar el kernel, se realizó una matriz de 3x3, la cual se llena con valores positivos y negativos.

Cuando se selecciona en el editor la opción borde, se despliega un menú que contiene las opciones de: afilado, repujado, desenfoque, realce negro, realce negro suave y relieve negro; para cada opción de borde se genera un kernel diferente, permitiendo tratar la imagen capturada. En la figura 14.a se muestra una imagen con bordes afilados, la figura 14.b muestra la imagen con una máscara de repujado, la figura 14.c se le aplicó un desenfoque.

- **Escala de grises.**

Toda imagen en blanco y negro, maneja enteros de ocho bits, lo que permite que sea una imagen monocromática, manejando 255 tonos diferentes. Si se varían los componentes de la matriz que están representado los pixeles de la imagen, se obtiene una escala de grises, dando a la imagen un color de blanco y negro, el programa asigna el color dependiendo si es 0 o 255, cuando reconoce un cero cambia de color el pixel a negro, con 255 cambia el pixel de la imagen a blanco, para poder realizar la escala de grises en el programa utiliza los siguiente parámetros que permiten su programación en la plataforma netbeans en el lenguaje de programación Java:

1. Usar la clase `ColorConverterOp` la cual permite pasar el color de la imagen, al color que queramos, escogiendo `ColorSpace.CS_GRAY`.
2. Llamamos el método `filter()`, para poder realizar la conversión.

Figura 14.a Imagen afilada en el dominio espacial

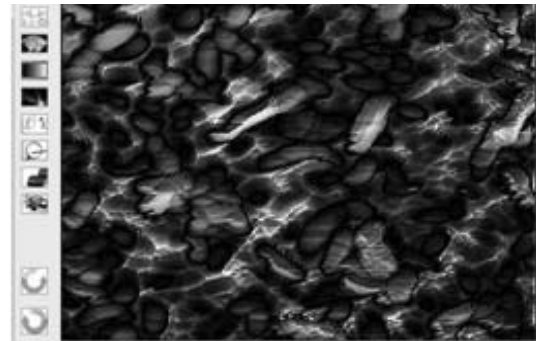


Figura 14.b Imagen repujada

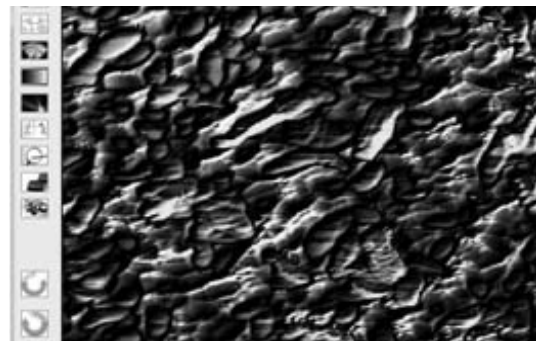
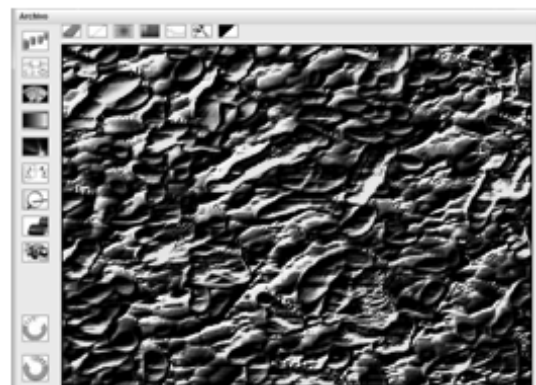


Figura 14.c Imagen con desenfoque



- **Vista original y giros.**

El programa cuenta con una herramienta con la cual permite ver la imagen capturada sin ningún tipo de tratamiento, se genera

una variable tipo `BufferedImage`¹³ asignando a esta variable la imagen original, se publica en un `JInternalFrame`, todo este proceso es inicializado cuando se ubica el puntero sobre el icono que indica vista previa, si por el contrario se ubica el puntero en otra parte, la ventana se cierra automáticamente.

Para realizar los giros derecha e izquierda, cada vez que se pulsan la imagen aumenta o disminuye su sentido de giro 10 grados, con respecto a la coordenada (0,0) que es el centro de la imagen.

- **Espejo, zoom y deshacer.**

Cuando se hace espejo, se crea una secuencia la cual funciona de la siguiente manera; la imagen se carga en un `BufferedImage`, se dibuja la imagen especificando las coordenadas de cada pixel, para el eje Y intercambiando las posiciones Yabajo con Ysuperior, para el eje X se cambian las posiciones Xizquierdo con Xderecho.

El programa maneja dos tipos de zoom, móvil y fijo.

El zoom móvil está formado a partir de un `JPanel`, el cual con ayuda de un menú donde están contenidos los tamaños a los cuales se puede ampliar la imagen. Una vez seleccionado el tamaño, se multiplica la imagen por esta cantidad y a medida que el mouse se desplaza sobre el lienzo se refleja en el `Jpanel`.

El zoom fijo se carga la imagen en una nueva ventana la cual contiene un `Jslider`, a medida que se desplaza se amplía o reduce la imagen dependiendo de su posición.

La opción deshacer reasigna la imagen original a la imagen que se está trabajando.

7. Perspectivas

Con el diseño del programa se deja base para que en un futuro se pueda desarrollar más ampliamente las aplicaciones de análisis de imagen, permitiendo reconocer en el área de la metalografía el tamaño del grano, forma y distribución de varias fases e inclusiones que tienen gran efecto sobre las propiedades mecánicas del metal, así como el reconocimiento de porcentajes de carbono, y muchos más complementos técnicos.

La ayuda del apartado encuestas, busca la opinión de los usuarios para que con sus recomendaciones o sugerencias, se puedan hacer modificaciones para lograr consolidar un programa hecho a la medida, en especial la conformidad con la interfase de usuario para mejorar botones, iconos, formas, textos, entre otras características.

8. Resultados

- El programa diseñado permite hacer un procesamiento de la imagen con las herramientas expuestas anteriormente debido a la resolución de la cámara estas no pueden ser mejores (Ya que la cámara es la que trae el microscopio por defecto), pero todas funcionan correctamente, ya que los algoritmos de procesamiento de imágenes fueron probados con fotografías de materiales del laboratorio y otras externas (retratos, entre otros).
- Este programa está funcionando en el computador del laboratorio donde se han hecho pruebas que han permitido enriquecer la base de imágenes. Tales muestras se usan como patrón para la comparación con otras que sean tomadas por usuarios especializados o en su defecto usuarios del mismo microscopio,

¹³ `BufferedImage`: La clase `BufferedImage` permite crear imágenes y guardarlas en la memoria RAM, este formato dentro de java permite la manipulación de los valores RGB que componen la imagen.

ajenos a profesiones relacionadas con la metalografía.

- Se considera que para una integración total del aplicativo, se acompaña de una fuente para la alimentación de la cámara, un manual de instrucciones para la instalación y manejo, además de un capturador de video.

9. Conclusiones

- Con la ayuda del programa, el usuario pudo ver la muestra directamente en la pantalla del computador, dándole la facilidad de enfocar las imágenes sin tener que observar la muestra por medio de los binoculares.
- La usabilidad del microscopio se ha elevado en un 100%; dada la facilidad de manejo, rapidez, versatilidad y buena definición en las imágenes, proporcionadas por la aplicación.
- Se consideró el software libre como herramienta para elaborar el programa dado que es un software optimo para procesar matrices y, por tanto imágenes digitalizadas (Una imagen digitalizada es una matriz de $M \times N$), para este desarrollo se utilizo una metodología de desarrollo en espiral ya que La atención se centra en la evaluación y reducción del riesgo del proyecto dividiendo el proyecto en segmentos más pequeños y proporcionar más facilidad de cambio durante el proceso de desarrollo, así como ofrecer la forma de ir de inicio al final de la aplicación y viceversa sin que haya interrupciones en la misma.
- Las metodologías de desarrollo de software libre brindan una gran base para la construcción de aplicaciones de software

de pequeño, mediano y gran tamaño, ya que brindan las herramientas necesarias para su correcta planeación, desarrollo e implementación. Como registro de la implementación se enriqueció una base de imágenes con muchas imágenes (alrededor de 100) tomadas de materiales usados normalmente por los estudiantes para sus pruebas y eso descansa en el laboratorio ya que por defecto se instala con el mismo programa.

Referencias

- [1]. V. Martínez, "Software De Tratamiento De Imágenes". Consultado Febrero de 2005. Disponible en: http://ciclope.fi.upm.es/tools/titere/index_es.
- [2]. J. Gonzales. Procesar imágenes con software libre. Consultado en Febrero de 2011. Disponible en: <http://www.getpaint.net/>
- [3]. M. González; B. Del Río, "Programa simuladores de microscopios para el autoaprendizaje de metalografía". Madrid: Universidad Politécnica de Madrid. Consultado Septiembre de 2010. Disponible en <http://www.dii.etsii.upm.es/ntie/pdf/delrio.pdf>
- [4]. Lyon. "Transformaciones de la imagen". Consultado Octubre de 2010. Disponible en: <http://www.cs.buap.mx/~mmartin/pdf>
- [5]. G. Baxes, "Image Processing Principles And Applications"; Jhon Wiley & sons.INC. 1994.
- [6]. M. Galer, "Tratamiento Digital de Imágenes" peñalara S.A. 2003.
- [7]. J. Pertusa Grau, "Técnicas de Análisis de Imagen"; Aldia: Maite Simon. 2003.
- [8]. D. Cuervas, "Procesamiento de imágenes con Matlab". Consultado Octubre de 2010.

Disponible en: <http://www.scribd.com/doc/23371/Procesamiento-de-imagenes-con-Matlab>

- [9]. J. Bernal, "Sistema De Identificación Para Inspección De Piezas Por Medio De Visión Artificial", Bogotá. 2008.
- [10]. O. Bertel; E. Moreno, "Aplicación De Transformada Wavelet Para Reconocimiento De Formas En Vision Artificial", Bogotá. 2008.
- [11]. A. Castillo, "Sistema De Reconocimiento De Señales De Tránsito Por Visión Artificial", Bogotá. 2008.
- [12]. R. Colom, "Transformada Discreta Wavelet 2-D para procesamiento de video en tiempo real". Consultado Octubre de 2010. Disponible en: <http://www.upv.es/dsd/publica/articulos/alicante01.pdf>
- [13]. Curtin, Dennis. (2010). Elección de una cámara digital .Extraído desde http://www.sapiensman.com/Imagen_digital/index.htm. El 14 de Octubre de 2010.
- [14]. Roger. S. Pressman. Ingeniería del Software. Impresos y revistas, S.A (IMPRESA). 1993.
- [15]. González, Rafael. (2009) Procesamiento Digital de la imagen. Tomado de: http://es.wikipedia.org/wiki/Procesamiento_digital_de_im%C3%A1genes. El 14 de Octubre de 2010.
- [16]. Becerra Santamaría. C. Los 600 Principales Metodos Del Java; Bogotá: Kimpers Ltda. 2001.
- [17]. Knudsen, J. Java 2D Graphics; O' Reilly & associates. 1999.