Diseño de *software* para el procesamiento y análisis de imágenes biomédicas en 2D utilizando librerías ITK

Software design for processing and analysis of biomedical images in 2D using ITK libraries

Wilmer Antonio Blanco Riaño
Tecnólogo en Electrónica. Estudiante de Ingeniería en Control de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia.seralejo2004@yahoo.es

Sergio Alejandro Rojas Barbosa
Tecnólogo en Electrónica. Estudiante de Ingeniería en Control de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia.wilmerbros@gmail.com

Fernando Martínez Santa
Ingeniero en Control ymagister en Ingeniería Electrónica. Docente de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia.fmartinezs@udistrital.edu.co

Clasificación del artículo: investigación (Recreaciones)

Fecha de recepción: agosto 26 de 2009 Fecha de aceptación: febrero 2 de 2010

Palabras clave: Imágenes, ITK, Pipeline, Procesamiento, Segmentación, Umbral, Varianza.

Key words: Images, ITK, Pipeline, Processing, Segmentation, Threshold, Variance.

Resumen

Las librerías ITK (InsightSegmentation and RegistrationToolkit) son en conjunto una herramienta de código abierto, multiplataforma, que proporcionan un sistema con una amplia gama de herramientas de *software* para el análisis y el procesamiento de imágenes.ITK emplea algoritmos de vanguardia para el registro y segmentación de datos multidimensional; algoritmos orientados a la exploración y manipulación de diferentes características de las imágenes, lo que permite interactuar con diferentes funciones de las imágenes tales como umbralización, detección de bordes, gradiente, curvatura, entre otros filtros de imagen, los cuales se encargan de destacar particularidades en una imagen dependiendo del uso que se quiera brindar. La aplicación ITKUD fue implementada con el fin de hacer uso de algunos de estos algoritmos y así mismo de crear un ambiente de *software* totalmente libre permitiendo su portabilidad, es decir, su compilación en cualquier sistema que soporte C++.

Abstract
The libraries ITK (InsightSegmentation and RegistrationToolkit) make up one open source, cross platform, providing a system with a wide range of software tools for analyzing and processing images. ITK employs leading-edge algorithms for the registration and segmentation of multidimensional data, algorithms aimed at the exploration and manipulation of different characteristics of images, allowing different functions to interact with images such as thresholding, edge detection, gradient, curvature and other filters image, which are responsible for particular highlight in an image depending on the application you want to provide. ITKUD application was implemented to make use of some of these algorithms and also to create a completely free software environment allowing for portability, ie, the compilation on any system that supports C++.

* * *

1. Introducción

Este proyecto se desarrolla en un campo que se ha ido ampliando en los últimos años: la Visión Artificial. La visualización se puede definir como el proceso de explorar, transformar y mostrar datos en forma de imágenes para comprender y apreciar adecuadamente las características de los mismos. Se entiende por procesamiento digital de imágenes la manipulación de las mismas a través de un computador, de modo que la entrada y la salida del proceso sean imágenes. En la actualidad se encuentran con mayor facilidad equipos que permiten una captura de imágenes de forma más detallada, lo cual permite la elaboración de aplicativos que tratan señales bidimensionales y tridimensionales, dedicados a mejorar en ciertos aspectos visuales la imagen con el fin de proveer al usuario de herramientas para interpretarlas[2].

Existen diversas herramientas destinadas a la visualización de imágenes en un entorno informático, comprendidas entre ellas las librerías ITK, las cuales son una herramienta relativamente nueva en el procesamiento digital de señales e imágenes, tomando como base la segmentación proceso que se encarga de clasificar e identificar los datos encontrados en una representación muestreada digitalmente [2].

Es de gran importancia mencionar que esta herramienta nace con el fin de apoyar el Proyecto Humano Visible, que se está llevando a cabo en la Biblioteca Nacional de Medicina de EE.UU, el cual consiste en hacer una recolección y análisis de imágenes transversales del cuerpo humano de un hombre y una mujer por medio de TAC, IRM y criosercción, con el fin de lograr muestras de un milímetro. La aplicabilidad de las ITK va enfocada al procesamiento y al análisis de estas muestras, ya su vez generar la base teórica y tecnológica de un proyecto de esta magnitud.

En otras ciencias como la biomedicina actualmente se requiere el uso de nuevos métodos de investigación, automatizando procedimientos y detección de anormalidades en estudios que normalmente son invasivos. Las librerías ITK son aplicables en técnicas como las angiografías, donde se requiere la detección y el detalle de contornos en arterias, en estudios de estructuras cerebrales en los casos donde es necesario identificar y clasificar las concentraciones de masa gris y blanca, en análisis de diferenciación de tejidos, en detección de anormalidades en colonoscopias y en muchos otros procedimientos en esta misma rama.

De esta herramienta se pueden destacar algunas características:

- Permite la implementación de funciones o filtros básicos usando cada librería aparte.
- Permite hacer *pipeline* (conexión entre los resultados de la ejecución de algún algoritmo de procesamiento, a la entrada de otro, con el fin de

utilizar una secuencia de diferentes algoritmos y obtener resultados más elaborados).

- Se puede interactuar con los tonos de intensidad de la imagen y de umbral permitiendo aplicar funciones matemáticas, binarias o físicas a la imagen.

El objetivo principal es diseñar e implementar una aplicación en la cual se puedan desarrollar y visualizar algunas de las funciones realizables con las librerías ITK, además sentar una base teórica de la herramienta para cualquier investigación futura del grupo Digiti en este mismo campo.

Figura 1. Logo ITK [1].

2. Metodología

El *software* para el procesamiento y análisis de imágenes, utilizando librerías ITK, se desarrolló en tres fases: construcción del entorno de visualización, selección de imágenes y la implementación de filtros y funciones con las ITK.

2.1. Construcción del entorno de visualización

Se elaboró un programa en el cual se pueden visualizar todas las opciones a las que el usuario tiene acceso.Se recurrió al programa de interfaz gráfica wxWidgets V. 2.8.9, el cual se describe como una herramienta de C++ que proporciona una GUI (interfaz gráfica de usuario) multiplataforma y permite a los desarrolladores crear aplicaciones para diversos tipos de sistemas operativos, es una herramienta amplia, libre y de código abierto [3]. Esta interfaz contiene un conjunto de librerías que nos permite hacer la interacción de datos entre el entorno de visualización y el procesamiento de la imagen a cargo de las librerías ITK.Cabe destacar la importancia de crear un entorno fiable y sin errores y permitir así una visualización de los resultados más armónica y completa, además de permitirle al usuario detallar los resultados e interactuar con la imagen.

Inicialmente se construyó una ventana de 400 x 300 píxeles, de color negro, que va a aparecer en el centro de la pantalla. En la parte superior de la aplicación se implementó un barra en el cual se puede tener acceso a los diferentes menús, el menú Archivo (Abrir, Guardar, Guardar como, Salir), el menú Editar (Deshacer, Repetir, Zoom, Rotar), el menú Filtros (Umbralizar, Detectar bordes, Suavizar, Gradiente, Erosión, Dilatación, Promedio, Mediana, Sigmoid, Curvatura), el menú Funciones (Matemáticas y Sobel) y el menú Ayuda.

El tamaño de la ventana puede variar dependiendo del tamaño de la imagen como tal. En la barra inferior "barra de estado" se describen atributos de la imagen en tamaño y porcentaje de escalamiento; adicionalmente, indicaciones al usuario acerca de cómo utilizar la aplicación cuando se necesite ingresar valores a la función por utilizar.

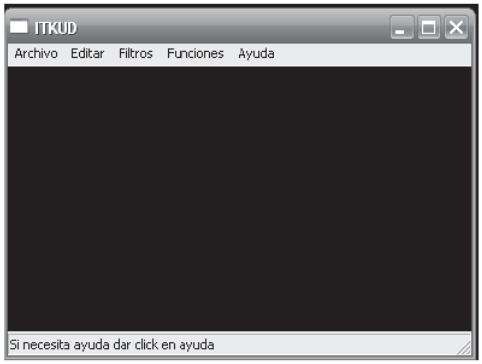

Figura 2. Entorno de Visualización.

En esta fase cabe resaltar la compilación de estas librerías wxWidgets y las librerías ITK mediante el

programa Cmake. El CMake es una herramienta multiplataforma de código libre empleada paraconfigurar y dirigir el proceso de construcción de aplicaciones. Los ficheros independientes llamados CMakeLists.txt se usan para describir el proceso deconstrucción y establecer las dependencias. Cuando ejecutamos CMake se generan los ficheros necesarios, dependiendo del compilador y sistema operativo que se estéutilizando. Esto sirve para compilar ITK fácilmente y hacer uso de las herramientaspropias de la plataforma en la que se esté trabajando.

El CMake necesita tres datos que deben ser indicados antes de su ejecución: el compilador que va a ser empleado, la dirección del código fuente y la direcciónde se generarán el código objeto, las librerías y los binarios producidos en la compilación. Finalmente, se producirán los *workspaces*, *makefiles* y todo lo necesario para controlar laconstrucción de los procesos por parte del compilador. Una vez hecho esto ya se puede cargar el fichero *workspace* generado (dsw) y crear nuestra aplicación [2].

2.2. Selección de la imagen

La imagen inicialmente es seleccionada por el usuario y admite imágenes de entrada de diferentes tipos (Jpeg, Png, Gif, Bmp, Pcx), de cualquier tamaño. Lo que se hace posteriormente con el programa de entorno es reescalar la imagen, es decir, acoplarla a un tamaño justo para poderla visualizar sin inconvenientes.

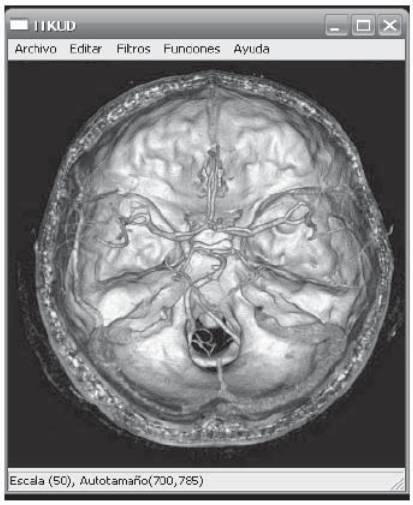

Figura 3. Imagen cargada en la ventana.

2.3. Implementación de filtros y funciones con las librerías ITK

Inicialmente se seleccionaron los filtros por implementar, teniendo en cuenta aquellos que generen resultados más representativos visualmente y que en la misma estructura de código sea más sencillo identificarlos con base a lo que se dedujo de la documentación de la tecnología. De igual manera, se incluyeron filtros que permiten el uso de dos o más librerías y el desarrollo de funciones matemáticas con base a estructuras de código como los iteradores. Cabe resaltar que los filtros y funciones elegidos se visualizan en tonalidad de grises.

2.3.1. Tipos de filtros implementados

- **Umbralización**

La umbralización de una imagen en escala de grises nos permite, dado un umbral definido por algún método, separar lo que es fondo de los que es objeto, siempre y cuando el fondo y el objeto tengan sus niveles de gris agrupados en dos modos dominantes.

Analíticamente, el concepto de umbral de una imagen se define como:

$$T = T[x, y, f(x, y), p(x, y)] \quad (1)$$

Donde f(x, y) es la intensidad del punto (x, y) y p(x, y) representa alguna propiedadlocal medida en un entorno de vecindad de este punto.La aplicación de un umbral a la imagen permite dividir dicha imagen en subconjuntosde píxeles con características afines.

Por medio de las librerías ITK se utilizó la función ThresholdImageFilter, con la cual se puede aplicar un tipo de umbralización. Este filtro realiza una transformación a una imagen binaria cambiando el valor de intensidad del píxel, según se ilustra en la figura 4, se le ingresan valores de umbral alto y umbral bajo, los cuales están limitados por el lower-Thresholdq es cero y upperThreshold el máximo que es 255; se comparan los valores ingresados con los límites superior e inferior y si el valor del píxel está dentro del rango se le asigna un valor de píxel de 255 InsideValue, de lo contrario un valor de OutsideValue, en este caso de cero [4].

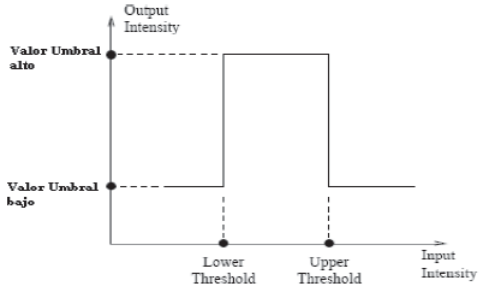

Figura 4. Función de transferencia de ThresholdImageFilter.

En el siguiente caso (figura 5) se ingresan valores de Umbral bajo de 20 y Umbral alto de 190, tratando de resaltar los tonos claros de la figura3 y eliminando el fondo resaltando más el objeto o el elemento que puede ser analizable.

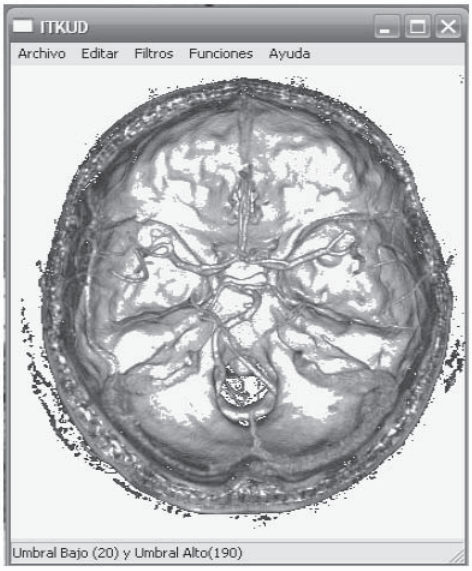

Figura 5. Umbralización.

- ***Suavizado***

El suavizado borra los detalles más finos de una imagen, es decir, conlleva una atenuación de las altas frecuencias mientras se mantienen las bajas y medias frecuencias. Tiene un buen número de aplicaciones, algunas veces se emplea para simular una cámara desenfocada o para restar énfasis a un fondo. Mientras los fotógrafos usan un filtro de cámara para conseguir ese efecto, los artistas informatizados emplean filtros digitales [5].

El suavizado se implementó con la función de las ITKmoothingRecursiveGaussianImageFilter, la cual toma como variable de entrada sigma, que se refiere a una constante con la cual se implementará la máscara de suavizado. Se caracteriza por tener simetría de rotación, es decir, filtran por igual en todas las dimensiones; tienen un único lóbulo central y el peso de los píxeles vecinos disminuye a medida que nos alejamos del centro, es un filtro paso bajo, es decir, conserva una relación 1/sigma (luego a mayor sigma filtra más). Figura 6,suavizado con sigma =2.

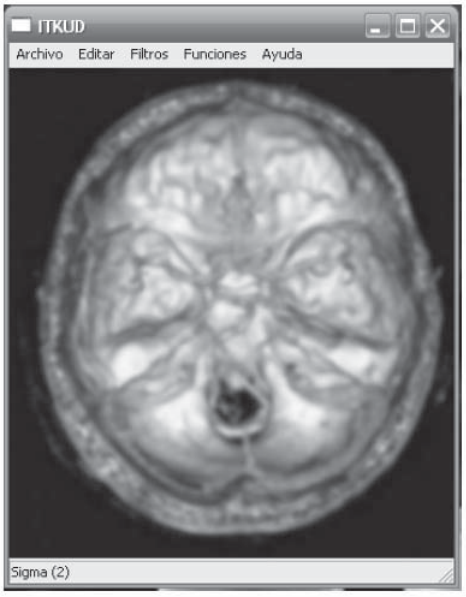

Figura 6. Suavizado.

- ***Detectar bordes (Canny)***

Consiste en la determinación de los puntos en que se produce una variación de intensidad. Para detectar los bordes se emplean operadores derivada, y es necesaria la segunda derivada de la intensidad. Algunas de las transformaciones que permiten detectar bordes son: el operador gradiente, el gradiente de Roberts, el detector horizontal, el detector vertical, el operador Prewitt, el operador Sobel y el operador Laplaciana.

En este caso se utilizó el operador gradiente de Roberts para la detección de bordes. Presenta la desventaja de que, dependiendo de la dirección, ciertos bordes son más realzados que otros, inclusive teniendo igual magnitud. Como resultado de su aplicación se obtiene una imagen con altos valores de niveles de gris, en regiones de límites bien definidos y valores bajos en regiones de límites suaves, siendo 0 para regiones de nivel de gris constante. El operador consiste en la siguientefunción:

$$(a') = (a - d)2 + (c - b)2 \qquad (2)$$

Donde a' es el nivel de gris correspondiente a la localización a, que será substituido, y a, b, c, d son las localizaciones cuyos valores serán computados para la operación [6].

El detector de bordes se implementó con la función ITKCannyEdgeDetectionImageFilter, el cual en su funcionamiento se observa que soluciona las limitaciones de sensibilidad, localización del ruido y es robusto (ver figura. 7).

Figura 7. Detector de Bordes.

- ***Gradiente***

Es usado como una técnica adicional de detección de bordes, pero se considera como una derivada de segundo orden; los detectores de bordes de la derivada

de segundo orden proporcionan una localización mejor del borde. Otra ventaja de los operadores de la derivada de segundo orden es que los contornos del borde detectados son curvas cerradas, lo cual es muy importante en la segmentación de imagen. El operador Laplaciano se define como una derivada de segundo orden, por lo cual obtiene resultados superiores a los anteriores y puede trabajar con imágenes donde las variaciones de intensidad no sean suficientemente abruptas para ellos. No obstante, presenta una sensibilidad más grande frente al ruido y una ligera incapacidad para determinar la dirección de los bordes.

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \qquad (3)$$

El Laplaciano es un buen ejemplo de un operador de derivada de segundo orden y se distingue de los otros operadores porque es omnidireccional, es decir, destacará los bordes en todas las direcciones. El operador Laplaciano producirá bordes más agudos que la mayoría de las otras técnicas. Estos toques de luz incluyen pendientes positivas y negativas de la intensidad. El borde Laplaciano de una imagen puede ser encontrado convolucionando con máscaras, como lo muestra la figura 8.

1	-2	1		-1	-1	-1		0	-1	0
-2	4	-2		-1	8	-1		-1	4	-1
1	-2	1		-1	-1	-1		0	-1	0
Laplaciano1				Laplaciano2				Laplaciano3		

Figura 8. Convolución.

La imagen resultante exhibe un cambio del signo en los bordes de la imagen. Estos cambios de signo son referidos como pasos cero. Después del operador de convolución la imagen se debe procesar para encontrar estos pasos cero y para fijar, por consiguiente, los píxeles de la salida.

Como se puede ver, las máscaras del operador Laplaciano coinciden con las máscaras de filtro paso alto vistas anteriormente. Esto se debe a que el Laplaciano detecta los bordes, es decir, las altas frecuencias de la imagen, sin considerar la orientación, por lo que además de utilizarse para la detección de bordes sirve para el filtrado paso alto de imágenes [5].

Para realizar la visualización se implementó la función ITKGradientMagnitudeImageFilter. Este filtro calcula la magnitud de gradiente de la imagen en cada píxel. El proceso de cálculo es equivalente a suavizar la imagen con la funciónconvolución Gaussiana y, a continuación, la aplicación de un operador diferencial, en este caso el operador Laplaciano. Esto se puede ver de mejor forma en la figura 9, en la cual aplicamos nuestro filtro llamado "Gradiente" a la imagen de entrada, como se ha hecho con los filtros anteriores.

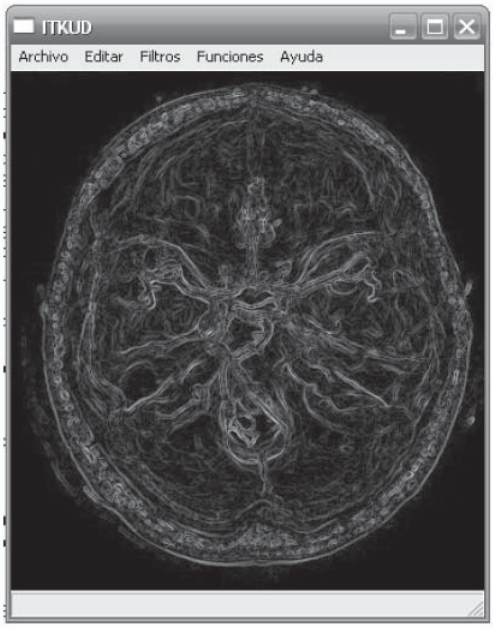

Figura 9. Gradiente.

- ***Erosión***

Consiste en la degradación progresiva de uno de los campos 0 o 1. Un elemento del campo por degradar seguirá perteneciendo al mismo si está rodeado de elementos iguales a él;en caso contrario, pasará al otro campo. Se trata de un proceso iterativo, que de ejecutarse cierta cantidad de veces terminará por destruir la imagen [6].

Se implementó la función GrayscaleErodeImage-Filter, en la cual se ingresa un valor de radio asociado a la iteración del filtro alrededor de un punto imaginario en el centro de cada píxel. Se observa en la figura10 el ejemplo con un radio de 2.

Puede verse con la aplicación del filtro cómo las regiones de color más oscuro van aumentando visualmente en tamaño, además de resaltarlas con color más oscuro diferenciándolas de las tonalidades más claras.

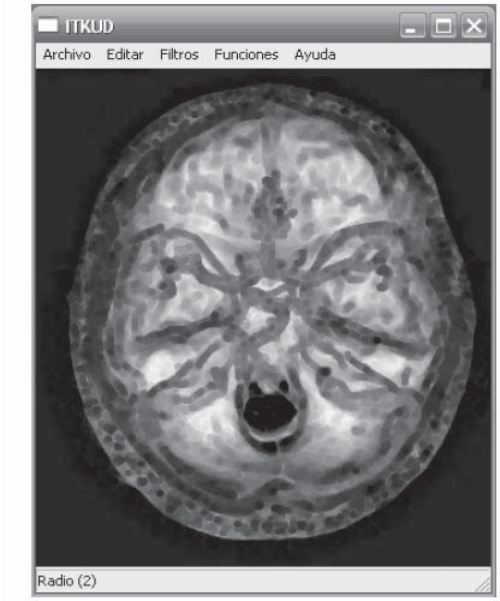

Figura 10. Erosión.

- ***Dilatación***

Es el crecimiento progresivo de uno de los dos campos 0 o 1. Un elemento del campo contrario a crecer será convertido si posee algún vecino perteneciente al campo que se expansiona. En caso contrario, permanecerán igual los elementos pertenecientes al campo por expansionar y que, como es obvio, no se modifican [6].

Se implementó la función GrayscaleDilateImage-Filter, en la cual se ingresa un valor de radio asociado a la iteración del filtro alrededor de un punto imaginario en el centro de cada píxel .Se observa en la figura 10 el ejemplo con un radio de 2. Caso contrario al filtro mencionado anteriormente, en este las partes más claras son detalladas y crecen. Este filtro es ideal en aplicaciones que requieran aumento de brillo.

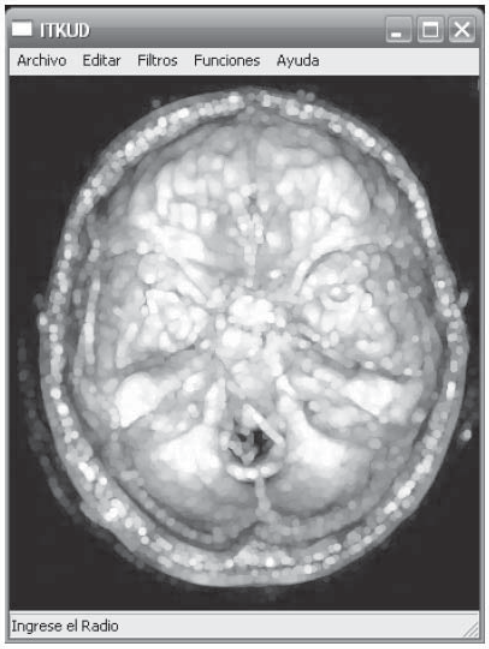

Figura 11. Dilatación.

- ***Mediana***

Los píxeles de la nueva imagen se generan calculando la mediana del conjunto de píxeles del entorno de vecindad del píxel correspondiente a la imagen origen. De esta forma, se homogenizan los píxeles de intensidad, muy diferente con respecto a la de los vecinos. Este tipo de filtro es bastante indicado cuando se tiene ruido aleatorio [7].

Se usó la función de ITKMedianImageFilter, que es un filtro sólido enfocado en eliminar ruido en la imagen y calcula el valor de cada píxel de salida como la estadística media de los valores de entorno de cada píxel de entrada.

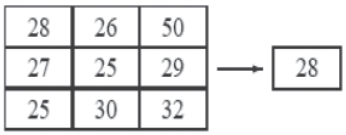

Figura 12. Estadística Media.

Es decir que cada píxel de salida tomará un valor medio de 28, haciendo el "barrido" por todos los píxeles que conforman la imagen (ver figura13).

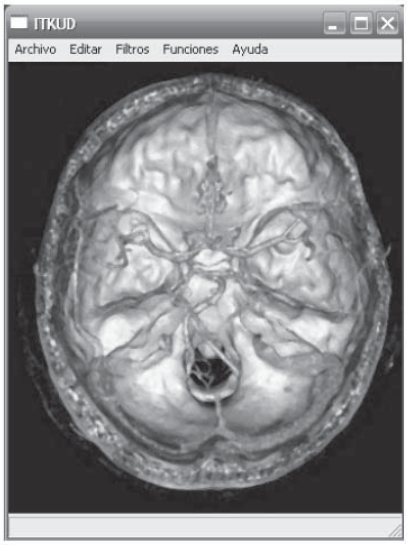

Figura 13. Mediana.

- ***Promedio***

Se obtiene la nueva imagen sumando N veces la imagen obtenida en sucesivas captaciones de la imagen. De este modo se conservan las formas manteniéndose los contornos, pero para ello es necesario que durante la captación de las sucesivas imágenes los objetos por capturar se encuentren en reposo [7].

Se utilizó la función ITKMeanImageFilter, la cual es muy similar al anterior filtro, pero en este caso se realiza un promedio de los valores de píxel de entrada, interactuando sobre los vecinos y haciendo el barrido por todo el tamaño de la imagen; se logra también la eliminación del nivel de ruido.

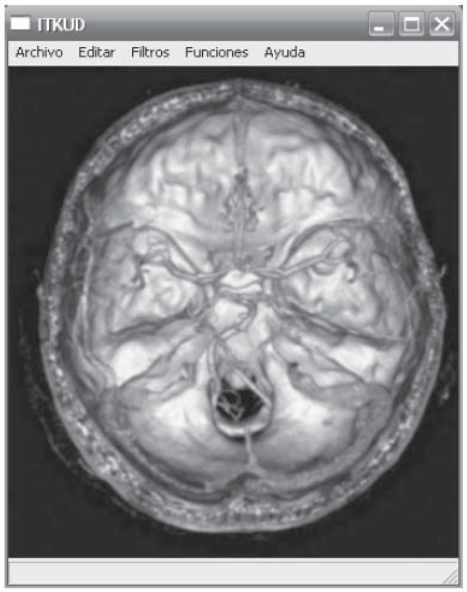

Figura 14. Promedio.

- ***Función Sigmoid***

Este tipo de filtro es comúnmente usado cuando una intensidad se transforma, traza un mapa específico del rango de la intensidad, valora el rango y elabora una nueva intensidad muy suave e interrumpida. Los Sigmoid son ampliamente usados como un mecanismo para enfocar la atención en un conjunto especial de valores y atenuar los valores progresi-

vamente fuera de esa gama. Para ser implementado en ITK incluye cuatro parámetros que pueden ser determinados en las variables de entrada; uno es la variable alfa, otro la variable beta y los dos valores más son el mínimo y el valor máximo, todos ellos destacados en la Ec. (4) que representa el filtro [4].

$$I' = (Max - Min) * \frac{1}{(1 + e^{\left(\frac{1-\beta}{\alpha}\right)})} + Min \qquad (4)$$

Estos valores fueron determinados por defecto en el algoritmo de la función, ya que una leve variación puede llegar a modificar la visualización del filtro, asumiendo valores como: alfa = 10; beta = 170; mín = 10 y máx = 240, y se da como resultado lo visualizado en la figura 13.

Con este filtro se logra apreciar la intensidad de la imagen, se permite diferenciar de forma más clara las tonalidades en cuanto a niveles claros y oscuros; es ideal en aplicaciones que requieren diferenciación de dos elementos con diferentes características, sin la necesidad de eliminar una propiamente, más bien la caracterización de cada una de forma más clara y contrastada.

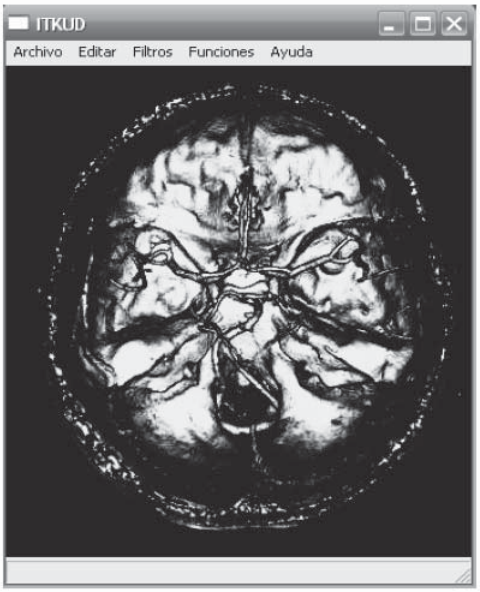

Figura 15. Función Sigmoid.

- ***Operador Sobel***

El operador gradiente de Sobel tiene la propiedad de realzar líneas verticales y horizontales más oscuras que el fondo, sin realzar puntos aislados. Consiste en la aplicación de dos máscaras que componen un único resultado, las cuales se describen a continuación:

a)

-1	2	-1
0	0	0
1	2	1

Figura 16.

b)

-1	0	1
-2	0	2
-1	0	2

Figura 17.

La máscara (a) detecta las variaciones en sentido horizontal y la máscara (b), en sentido vertical. El

resultado de esta aplicación, en cada "píxel", es dado por a'= a2 + b2. Donde a' es el valor de nivel de gris correspondiente a la localización del elemento central de la máscara. Las figuras siguientes ilustran el efecto de su aplicación [6].

Con la implementación de este tipo de filtro se quiere mostrar la interacción que se puede realizar entre diferentes funciones de ITK. En este caso se utilizó ImageRegionIterator y la función ConstNeighborhoodIterator y se dan como datos de entrada dos valores de Sobel que no están delimitados por valor alguno.

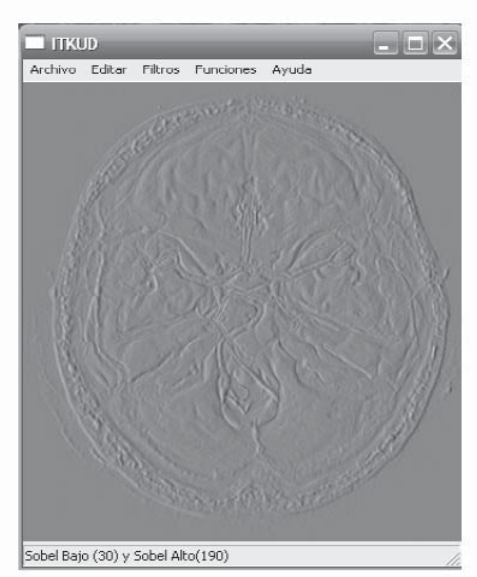

Figura 18. Operador Sobel.

- ***Operaciones matemáticas***

Se implementaron operaciones matemáticas básicas como la suma, la resta, la multiplicación y la división, donde este tipo de operación se realiza con una constante definida por el usuario. Se implementó de forma manual creando un algoritmo de iteradores con estructuras "for", de la siguiente manera:

```
for(iter1.GoToBegin(),iter2.GoToBegin();!iter1.IsAtEnd(); ++iter1,++iter2)
{
iter2.Set(iter1.Get()+ Cons);
}
```

Figura 19. Operaciones matemáticas.

Aquí se puede identificar el iterador 1 e iterador 2, los cuales van realizando un barrido por toda la imagen a lo largo de los dos ejes x e y con la inclusión de la constante que modifica directamente la apariencia del brillo de la imagen.

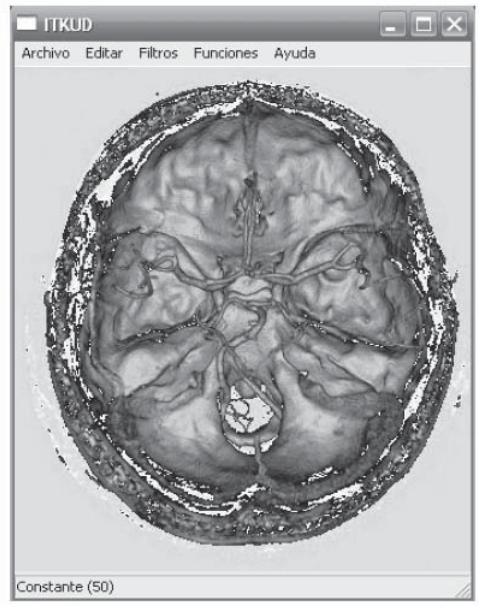

Figura 20. Ejemplo Resta Constante = 50.

3. Diagramas del *software*

En el primer diagrama se puede observar con facilidad y a grandes rasgos cuál es la lógica del programa. Se incluyen, además, todas las funciones integradas en la aplicación.

En el segundo diagrama se muestran las diferentes opciones que el usuario del programa podrá acceder a través de una ventana principal (diagrama de uso).

En el tercer diagrama se plasma la interacción entre objetos y clases de la herramienta que nos proporciona el GUI (Wxwidgets), y de igual forma las correspondientes estructuras de ITK. Esto se realiza por medio de diagramas UML.

En el último diagrama se puede observar un funcionamiento mas detallado del uso de archivos utilizados por el programa.

4. Visión del proyecto

El *software* para el procesamiento y análisis de imágenes (ITKUD) es una excelente herramienta para aquellos que apenas se inician en el mundo de la Visión Artificial, ya que ITK cuenta con innumerables funciones y filtros, que van desde los que implementamos hasta filtros con resultados muy precisos a ciertas necesidades del mundo científico.

Se pretende dar a conocer la utilidad de esta herramienta de una manera demostrativa, sembrando una base teórico – práctica de la implementación de los algoritmos. Así mismo, se podrá contar con el código fuente del *software*, los manuales de instalación de las librerías ITK y la interfaz gráfica en el grupo de investigación Digiti, continuando con la premisa de extender la aplicación en su ámbito de *software* libre y portable.

En una fase posterior del proyecto se podrían integrar imágenes a nivel 3D, tomando en cuenta una escala de color más amplia, ya que aunque ITK permite esta opción la mayoría de las funciones están dirigidas a imágenes de 16 u 8 bits, es decir, en escala de grises o blanco y negro. Actualmente las muestras se pueden obtener a partir de los equipos para estudios como tomografías computarizadas o resonancias magnéticas. En ciencias como la biomedicina se está evolucionando en la automatización de procesos de detección de anormalidades a un nivel de precisión muy alto, exigiendo márgenes de error mínimos.

En la fase posterior se podría implementar un *software* con muchas más opciones al analista, de forma dinámica, y luego la integración total del elemento estudiado, partiendo de las muestras y reconstruyéndolas en 3D. Actualmente se puede utilizar la tecnología del mismo fabricante en las librerías VTK, las cuales son un sistema de visualización que no solo permite visualizar geometría, sino que también incluye métodos vectoriales, escalares, tensores, de textura y volumétricos, aumentando más los métodos por implementar en futuras aplicaciones [2].

5. Resultados

Se desarrolló el *software* de procesamiento de imágenes ITKUD, el cual dispone de las siguientes características:

- Permite cargar distintos tipos de imágenes (Jpeg, Png, Gif, Bmp, Pcx).
- Realiza una modificación al tamaño visual de la imagen para poder visualizar los resultados y en la barra de estado se le informa al usuario el tamaño en porcentaje al que está siendo mostrada, todo sin modificar la imagen original.
- El usuario selecciona los filtros de imagen y funciones por desarrollar. La aplicación permite la visualización, además de funciones como el Zoom y el rotar herramientas útiles para mayores detalles sobre la imagen procesada.
- Los resultados de la aplicación de cada algoritmo de procesamiento se muestran dinámicamente al usuario.
- Existe la posibilidad de deshacer el último algoritmo aplicado sobre la imagen.
- Permite al usuario guardar los cambios, ya sea en el mismo archivo o en otro, conservando los tamaños originales de la imagen inicial y los procedimientos realizados.

Se apropiaron los conocimientos de implementación y uso de las librerías ITK, abriendo una nueva puerta en el tema de procesamiento de imágenes, en el entorno de la Facultad Tecnológica.

Se implementaron las versiones más actualizadas de ITK y WxWidgets, ya que permiten compatibilidad completa entre los algoritmos de procesamiento y la interfaz.

La documentación generada sobre el *software* realizado contiene el código fuente, los manuales de instalación y uso de las librerías utilizadas, diagramas UML, diagramas de bloques y manual de uso del *software*. Dicha documentación se podrá utilizar como base de investigación para futuros desarrollos en el mismo campo, al servicio del grupo de investigación y de la comunidad en general.

Se crearon manuales de uso de las librerías ITK, parte importante en la utilización de esta herramienta, en la cual se compilan dichas librerías. Con la ayuda del *software* CMake el anexo permite seguir paso a paso el proceso de documentación cuando se empiece a utilizarlas. También se generó la documentación de instalación de la interfaz gráficawxWidgets v2.8.9, desde el descargue en la página del fabricante hasta el uso en el compilador.

Se creó un documento que muestra de cómo se hace la implementación de una función de las librerías ITK, detalladas desde la documentación en la página web del autor hasta la inclusión de esta función en el código fuente. Por medio de la documentación creada se proporciona un medio para realizar ampliaciones al *software* diseñado, dotándolo de la funcionalidad extra que se requiera.

6. Conclusiones

El uso de las librerías ITK permite manipular, identificar y resaltar características básicas de la imagen, como el contraste, el brillo y destacar las formas, eliminación de ruido, resalto de tonalidades, entre otras. La abstracción de variables en una imagen, como el tamaño y la escala, es una base para fijar nuestro entorno de visualización, y nos permite trabajar con imágenes de cualquier tamaño y poder ver los resultados a una mejor escala.

La detección de bordes es una técnica muy útil en la ciencia y la industria de hoy; ITK permite hacer gran cantidad de filtros, los cuales hacen esta identificación de maneras distintas utilizando diferentes métodos. Las funciones en ITK pueden utilizarse simultáneamente, y permiten generar el *pipeline* pasando la imagen de un filtro a otro cuando la necesidad sea un resultado específico y muy exacto.

Se implementaron las versiones más recientes de las librerías ITK V. 3.10 y wxWidgets v. 2.8.9 con el fin de evitar inconvenientes de compatibilidades entre los programas al implementar la interfaz y con el fin de que la aplicación sea lo más actualizada posible. La implementación de los algoritmos de las librerías ITK es relativamente sencilla si se cuenta con la documentación necesaria. Las librerías ITK cuentan con una gran cantidad de algoritmos que pueden ser combinados, lo cual puede ser una solución para variadas necesidades del procesamiento de imágenes.

Referencias bibliográficas

[1] L. Ibánez, W. Schroeder, L. Ng, J. Cates "The ITK Software Guide" Kitware, Copyright 2008-09. [En línea]. Disponible: www.itk.org/

[2] I. Moreno, "Desarrollo de algoritmos de pro-cesamiento de imágenes con VTK". [En línea].Disponible: http://www.elai.upm.es/spain/Investiga/GCII/personal/iberzal/PFC_I_Berzal.pdf

[3] K. Ollivier, "wxWidgets".[En línea]. Disponible: www.wxwidgets.org/

[4]. ITKSoftware, "User guide". [En línea]. Disponible: http://www.itk.org/ITK/help/documentation.html

[5] A. Dapena, "Técnicas de procesado de imagen". [En línea]. Disponible:enhttp://www.des.udc.es/~adriana/TercerCiclo/CursoImagen/curso/web/Indice.html

[6] D. Rodríguez, "Implementación multiplataforma de procesamiento de imágenes 2D mediante librerías ITK. [En línea]. Disponible: http://www.elai.upm.es/spain/Investiga/GCII/personal/dllanos/DLlanos.htm

[7] F. J. Muñoz Rodríguez, "Reducción del ruido en una imagen digital". [En línea]. Disponible: http://www4.ujaen.es/~satorres/practicas/practica2.pdf