

CREATOOLS: UNA PLATAFORMA PARA EL DESARROLLO DE SOFTWARE DE PROCESAMIENTO DE IMÁGENES MÉDICAS. APLICACIÓN A LA SEGMENTACIÓN DE ARTERIAS CORONARIAS Y A LA DETECCIÓN AUTOMÁTICA DE LESIONES EN IMÁGENES DE ANGIOTAC

CREATOOLS: A FRAMEWORK FOR THE DEVELOPMENT OF MEDICAL IMAGE PROCESSING SOFTWARE. APPLICATION FOR THE SEGMENTATION OF CORONARY ARTERIES AND FOR THE AUTOMATIC DETECTION OF LESIONS IN CT IMAGES

PALABRAS CLAVE (DeCS)

Procesamiento de imagen
asistida por computador
Anomalías cardiovasculares

KEY WORDS (MeSH)

Computer-assisted image
processing
Coronary vessel anomalies

Marcela Hernández Hoyos¹
Eduardo E. Dávila Serrano²
Esteban M. Correa Agudelo¹
Juan Sebastián Torres González¹
Ricardo A. Corredor Jerez²
Leonardo Flórez Valencia³
Claire Mouton²
Maciej Orkisz²

RESUMEN

En este artículo se presenta un *software* de código abierto, llamado CreaTools, cuyo principal objetivo es el procesar y facilitar la visualización de imágenes médicas. Este *software* flexible funciona en diferentes sistemas operativos (Linux, Mac OS X, Windows), se desarrolla en el lenguaje de programación C++ para asegurar una fácil integración de módulos C++ y proporciona a los usuarios herramientas computacionales para construir interfaces gráficas de usuario (GUI), incluidos los datos de entrada/salida (manejo de archivos), la visualización, la interacción y el procesamiento de datos. Este artículo muestra también la utilidad de CreaTools mediante un proyecto de investigación que consiste en la detección automática de lesiones arteriales. Los algoritmos desarrollados han sido implementados en una interfaz gráfica amigable con visualización 3D e interacción. Ejemplos de tales algoritmos incluyen la extracción de ejes de arterias y la generación de modelos descriptivos de arterias con lesiones y sin lesiones.

SUMMARY

This paper presents an open source software, called CreaTools, which main objective is the processing and visualization of medical images.. This flexible software works in different Operating Systems (Linux, Mac OS X, Windows), is developed in C++ programming language to ensure easy integration of C++ modules, and provides users with computational tools to

¹Grupo Imagine, Grupo de Ingeniería Biomédica, Universidad de los Andes, Bogotá, Colombia.

²CREATIS; Université de Lyon; Université Lyon 1; INSA-Lyon; CNRS UMR5220; INSERM U630, 69621, Villeurbanne, France.

³Grupo Takina, Pontificia Universidad Javeriana, Bogotá, Colombia.

build Graphical User Interface, (GUI) including data input/output (management of files), visualization, interaction, data processing, etc. This paper also shows the utility of CreaTools through a research project regarding the automatic detection of artery lesions. The developed algorithms have been implemented in a friendly GUI with 3D visualization and interaction. Examples of such algorithms include axis extraction of arteries and the generation of descriptive models of arteries with and without lesions.

Introducción

Actualmente es cada vez más común contar con imágenes médicas tridimensionales (3D), o, incluso, de dimensiones superiores, si se consideran las secuencias temporales de imágenes 3D o las imágenes 3D multiespectrales. Adicionalmente, publicar nuevos métodos de procesamiento de imágenes médicas requiere una validación cada vez más extensiva de grandes volúmenes de datos, lo cual supone dificultades para los médicos, quienes no son expertos en computación.

Cada etapa del proceso de desarrollo de nuevos métodos (pruebas, depuración visual, evaluación periódica, demostración), así como su validación clínica, requiere el uso de una interfaz gráfica de usuario (GUI, por sus siglas en inglés), apropiada tanto para visualizar las imágenes y los resultados como para interactuar con ellos. Ahora bien, es relativamente sencillo crear GUI para imágenes 2D; sin embargo, los mecanismos de visualización de imágenes 3D e interacción amigable con el usuario constituyen una cuestión para expertos en computación gráfica e interacción hombre-máquina. Si bien algunos especialistas en procesamiento de imágenes poseen dicho conocimiento, para la mayoría de ellos, y especialmente para los estudiantes de posgrado y los aprendices involucrados en proyectos de análisis de imágenes médicas, es altamente preferible enfocarse en el desarrollo de nuevos algoritmos, en vez de gastar tiempo adquiriendo el conocimiento de alto nivel requerido para crear las GUI correspondientes.

Se han propuesto varias iniciativas, tanto de la industria como de la academia, para cerrar esta brecha y proveer herramientas de interacción y visualización de alto nivel fáciles de utilizar. Estas

incluyen MeVisLab (1), GIMIAS o XIP (véase la página ToolKit, de la Red Europea de Excelencia VPH (2), y la tabla 1, para conocer las características). La mayoría de ellas tienen en común el uso de una serie de herramientas subyacentes ampliamente utilizadas en el medio, tales como el Visualization Toolkit (VTK) e Insight Toolkit (ITK) de Kitware (estas son de código abierto) (3), al igual que Qt (4). En el presente artículo se describe una iniciativa reciente llamada CreaTools y se ilustra su utilidad en el proyecto de análisis de arterias coronarias, actualmente en desarrollo.

Presentación de CreaTools

CreaTools (5) es un conjunto de programas y de herramientas de desarrollo de aplicaciones de procesamiento y visualización de imágenes médicas. Es de libre distribución y funciona sobre diversos sistemas operativos (Linux, Mac OS X, Windows). Ha sido desarrollado en el lenguaje de programación C++ (6), y ha facilitado así la integración de diferentes módulos específicos de lectura y de escritura de imágenes, la construcción de interfaces gráficas, la visualización, la interacción y el análisis de imágenes; además, está construido sobre el concepto de *cajas negras*, en el cual se establece un modelo cuyo interés radica en sus entradas, sus salidas y la funcionalidad que presta, más que en conocer cómo está hecho. En el caso de CreaTools, dicho modelo es llamado *Black Box Toolkit (bbtk)*.

En este artículo se ilustran la aplicación y la utilidad de CreaTools a través de un proyecto de investigación que aborda la segmentación de arterias coronarias y la detección automática de lesiones en imágenes de tomografía axial computarizada (TAC).

Tabla 1. Características de algunas plataformas de desarrollo de software de procesamiento de imágenes médicas, actualmente disponibles

	CreaTools	MevisLab	GIMIAS	XIP
Software libre	Sí	Sí	Sí	Sí
Código fuente libre	Sí (CeCILL)	No	Sí	Sí
Mac OS X	Sí	Sí	Sí	Sí
Linux	Sí	Sí	Sí	Sí
Windows	Sí	Sí	Sí	Sí
Arquitectura modular	Sí	Sí	Sí	Sí
Objetivo	Análisis y visualización	Análisis y visualización	Análisis y visualización	Análisis y visualización
Lenguajes de desarrollo	C++	C++, Python	C++	C++
Software libre de base	VTK, ITK, wxWidgets, Qt, BOOST	VTK, ITK, Qt, OpenInventor	VTK, ITK, MITK, BOOST, wxWidgets	VTK, ITK, OpenInventor
Formatos de imágenes soportados	DICOM, JPEG, TIFF, PNG, RAW, VTK, STL	DICOM, TIFF, JPEG, PNG, RAW, STL	DICOM, vtk, stl, Nifty, Analyze	DICOM
Instituciones creadoras	INSA de Lyon (Lyon, Francia), Universidad de los Andes y Pontificia Universidad Javeriana (Bogotá, Colombia)	MeVis Medical Solution (Bremen, Alemania)	Universitat Pompeu Fabra (Barcelona, España)	Washington University in St. Louis (St. Louis, USA)

Los algoritmos desarrollados por los investigadores del proyecto son integrados en CreaTools aprovechando sus funcionalidades de interfaz gráfica, visualización 3D e interacción. Tales algoritmos incluyen la extracción de ejes de arterias, la segmentación del lumen y la generación de modelos descriptivos de arterias sanas y patológicas, y su respectiva detección.

Además del proyecto de análisis de arterias coronarias (figura 1), CreaTools ha sido utilizado en múltiples trabajos de investigación (figura 2), tales como: Segmentación de corales; caracterización de tejido óseo en el maxilar inferior; análisis interactivo de imágenes cerebrales de bebés prematuros; cuantificación de tejido adiposo abdominal; y el Proyecto Europeo de Thrombus (7).

Las publicaciones relacionadas con los proyectos científicos que utilizan CreaTools están enumeradas en la página del proyecto PLUME-FEATHER (8). CreaTools puede ser utilizado, modificado y distribuido según los términos de la licencia francesa CeCILL, que rige el software libre, y se encuentra protegido por la Agencia Francesa para la Protección de Programas (IDDN.FR.001.250014.000.S.P.2010.000.20700).

Es posible identificar varios tipos de usuarios de CreaTools; a saber: usuarios finales de aplicaciones independientes; usuarios que construyen prototipos a partir de las cajas negras disponibles; desarrolladores de nuevas cajas y algoritmos. Los usuarios finales tienen acceso a varias miniherramientas, tales como la puesta en correspondencia interactiva de imágenes 2D y 3D, la extracción de subvolúmenes y el remuestreo de imágenes.

El usuario también dispone de aplicaciones completas, como un avanzado visor 3D interactivo que incluye renderización volumétrica (figura 3) y de superficies con múltiples valores de umbral, así como *creaContours*, una herramienta interactiva para trazar contornos en imágenes N-dimensionales. Esta última permite al usuario crear varias formas de contorno (*B-Splines*, círculos, rectángulos, anillos...), y así extraer información de la imagen: Calcular parámetros locales estadísticos dentro de regiones de interés bien definidas y sobre sus contornos, u obtener datos de píxeles de la región de interés, para utilizarlos en cálculos particulares.

Tanto los desarrolladores como quienes crean prototipos pueden beneficiarse de *bbtk*, el cual permite a los usuarios diseñar rápidamente aplicaciones pequeñas con propósitos de demostración que contengan componentes de alto nivel, como combinaciones de ventanas, botones, barras deslizadoras, etc., llamadas *MetaWidgets* de interacción, lectores/escribotes de imágenes, visores de imágenes/mallas, interactores, etc. Sus mecanismos de adición de nuevas funcionalidades a las aplicaciones ya existentes (*plug-in*) y documentación automática permiten compartir y obtener el mayor provecho de nuevas cajas negras.

Una de las principales características de *bbtk* es su flexibilidad: Cualquier código de C++ puede ser encapsulado en una caja negra y conectado con las entradas/salidas adecuadas de otras cajas negras, siempre y cuando cumpla con la definición de un filtro, definido, p.ej., por su función, sus entradas y sus salidas.

Actualmente se dispone de más de 300 cajas negras. Tales cajas se basan, principalmente, en herramientas de software desarrolladas por terceros: VTK, ITK, wxWidgets y Qt. Las últimas son herramientas de código abierto utilizadas en numerosos proyectos de procesamiento de imágenes médicas en todo el mundo. Esto garantiza su calidad y la futura interoperabilidad entre CreaTools y otros productos de software. Uno de ellos, *gdc*, ampliamente utilizado para gestionar (leer/hacer *parsing* y escribir) archivos DICOM, ha sido desarrollado por el mismo equipo de trabajo de CreaTools.

Es posible construir una aplicación (prototipo, aplicación de demostración...) al interconectar cajas negras y formar un proceso secuencial entre estas, llamado *pipeline*. Esto se puede lograr usando *bbs* (*black box script*), un lenguaje sencillo de *scripts* (archivo de texto simple que contiene la sintaxis apropiada para poder ser ejecutado), el cual comprende una docena de comandos para definir y parametrizar las cajas por utilizar, establecer los *pipelines* y ejecutarlos. Los *scripts* pueden ser editados y probados en un ambiente llamado *bbStudio*, que también provee asistencia en línea, ejemplos, y toda la documentación necesaria de las cajas negras existentes.

Recientemente se ha desarrollado un editor gráfico llamado *bbEditor*, que facilita enormemente la creación y la edición de *pipelines* y genera los *scripts* apropiados. En algunos casos, cuando la aplicación involucra componentes de muy alto nivel, el *pipeline* correspondiente puede ser muy simple (figura 4a). Al ejecutar este ejemplo se muestra un cuadro de diálogo para seleccionar un directorio que contenga imágenes médicas; luego, un lector de DICOM abre los archivos de imagen seleccionados y los envía a un visor (figura 4b).

Otra funcionalidad muy importante y útil de *bbtk* es la posibilidad de crear cajas complejas. Una caja compleja es una caja de cajas negras que, en realidad, está compuesta por un *pipeline* cuya utilidad ha sido comprobada, y que es susceptible de ser reutilizado en otras aplicaciones. Un mecanismo muy simple transforma tal *pipeline* en una caja negra que puede ser utilizada como cualquier otra caja, para así simplificar las redes y capitalizar el esfuerzo de los desarrolladores.

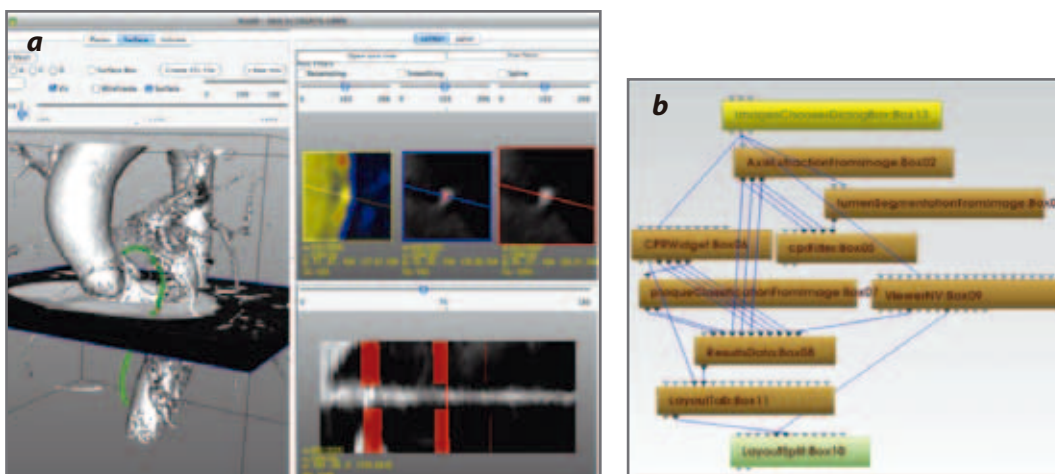


Figura 1. CreaTools utilizado en trabajos de investigación. a) Interfaz gráfica de la aplicación de estudio de arterias coronarias (segmentación y detección de lesiones). b) Red *bbtk* correspondiente a la aplicación.

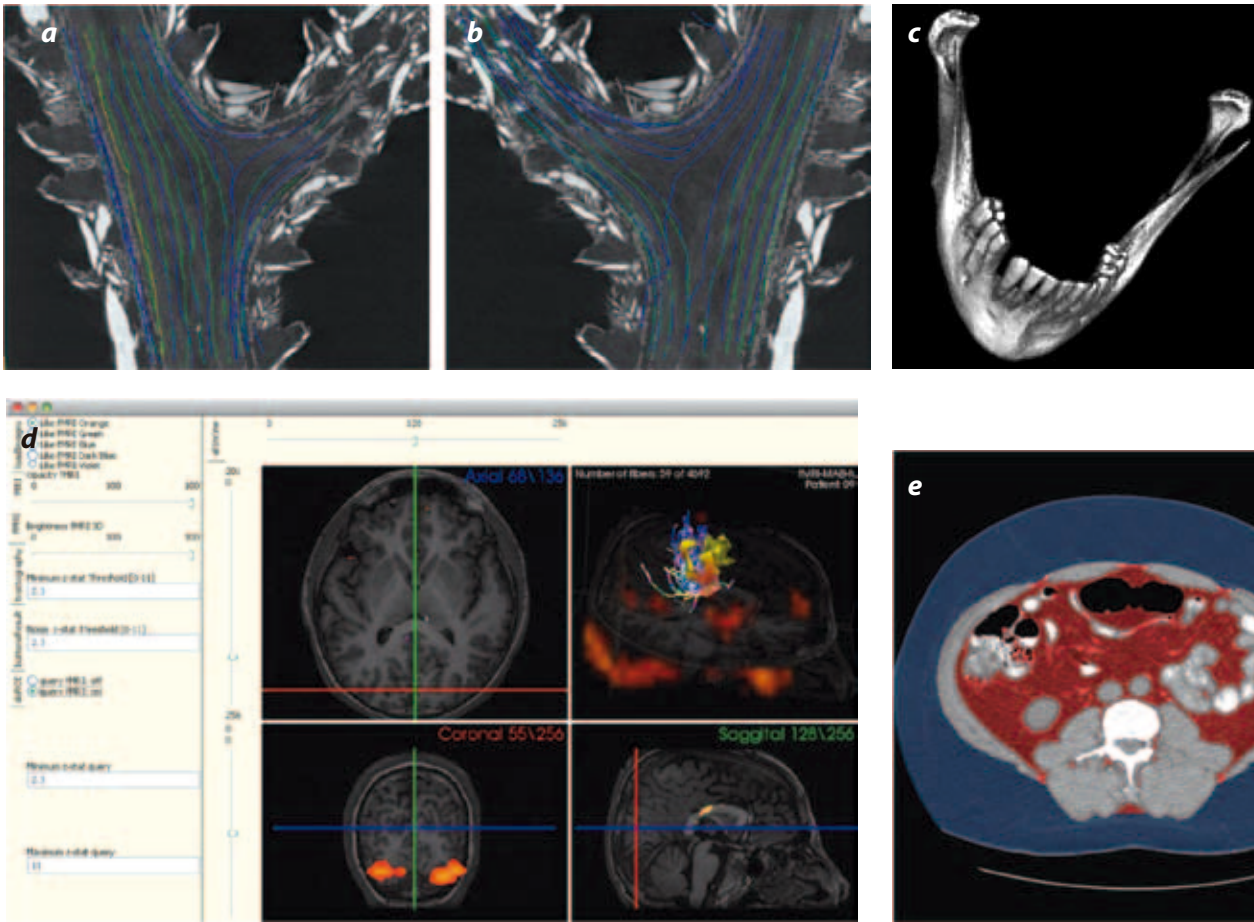


Figura 2. Ejemplos de proyectos hechos con CreaTools: a y b) Extracción y visualización de canales internos de octocorales en imágenes de microTAC. c) Segmentación 3D del maxilar inferior. d) Proyecto KABAf: Análisis interactivo de imágenes diagnósticas cerebrales de bebés prematuros procedentes de distintas fuentes (RMI, RMf, DTI). e) Proyecto CAAVAT: Segmentación y diferenciación del tejido adiposo abdominal (en azul: Tejido adiposo subcutáneo; en rojo: Tejido adiposo intravisceral).

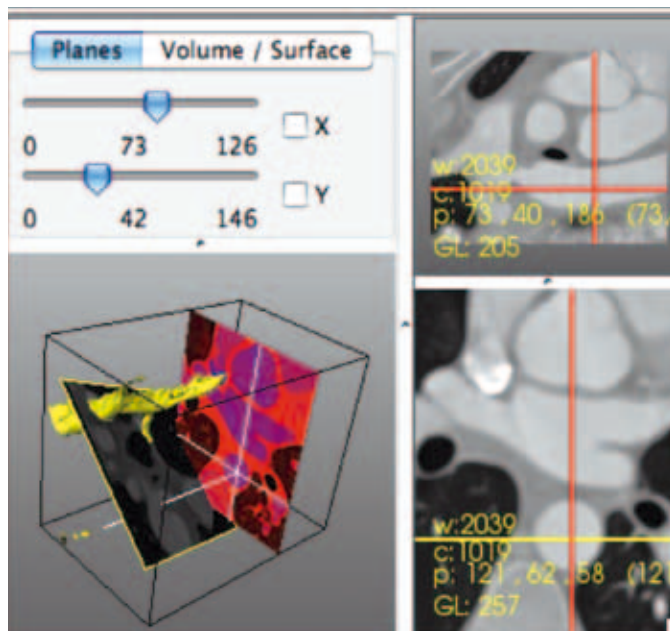


Figura 3. Ejemplo de renderización volumétrica utilizando CreaTools. A la izquierda, abajo, se puede observar el volumen generado de una arteria (en amarillo). A la derecha, arriba, se observa un plano diagonal (corresponde al plano en niveles de gris de la imagen de la izquierda), y a la derecha, abajo, se observa un plano axial del corazón (corresponde al plano con paleta de colores rojo en la imagen de la izquierda).

Lo anterior se puede ilustrar con la caja *Cropping*, desarrollada para el proyecto Thrombus. Su propósito es la selección interactiva de un subvolumen de imagen de interés (VOI, por sus siglas en inglés), recortarlo y crear la estructura de datos necesaria tal que el VOI pueda ser utilizado en la continuación del *workflow*. Esto involucra la sincronización de un cierto número de herramientas de visualización e interacción, y así da lugar a una compleja red con muchas cajas negras atómicas (figura 5). Tras definir las entradas y las salidas de este *pipeline*, la recién creada caja de *Cropping* se agrega al banco de cajas existentes. A partir de entonces puede ser utilizada como cualquier otra caja atómica, lo cual simplifica en gran medida la creación, la comprensión y la edición de *pipelines*.

Además de las cajas negras existentes, los desarrolladores tienen la oportunidad de crear nuevas cajas. También se ponen a disposición del desarrollador las herramientas necesarias para encapsular el código en C++ en nuevas cajas negras.

Además de los principales componentes de CreaTools previamente mencionados, cabe considerar ahora otros desarrollos importantes que se han integrado a CreaTools. Dentro de estos se encuentran: *creaImageIO*, la cual es una herramienta que maneja e indexa archivos de imagen; *Gimmick (Give me my medical images quick)*, o aplicación que lee formatos de imagen estándar (jpg, tif...) y formatos médicos (mhd, hdr, dcm, etc.), muestra los archivos seleccionados en un visor y permite al usuario organizar las imágenes en varios árboles (por

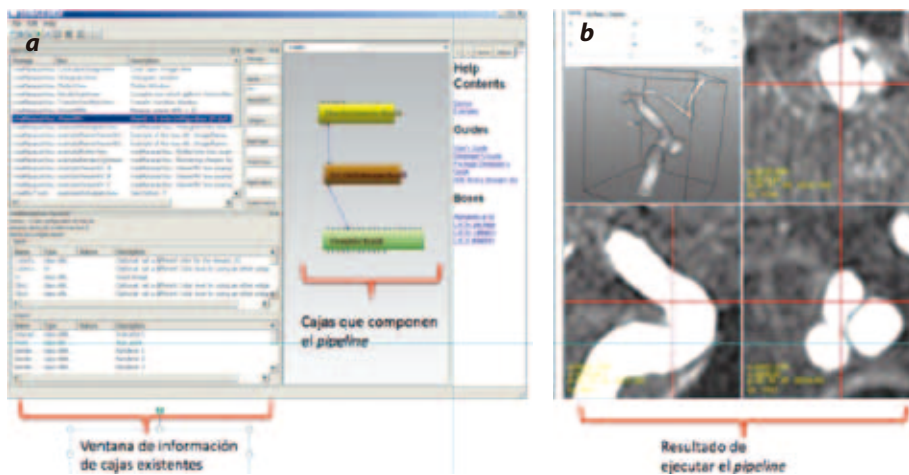


Figura 4. Ambiente para la creación de prototipos bbEditor. a) Ejemplo de una red simple, incluyendo un seleccionador de directorios (caja amarilla), un lector DICOM (caja café) y un visor (caja verde). b) Visualización de un aneurisma usando el visor: Representación 3D de la superficie y cortes ortogonales de la imagen.

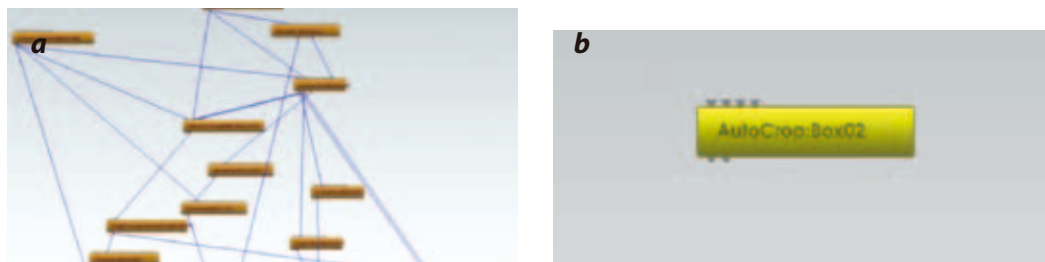


Figura 5. a) Estructura interna de la caja negra compleja de Cropping. Los cuadrados (rojo y verde) son, respectivamente, la entrada y la salida de la caja negra compleja. b) Todos los cuadros de la imagen en a) se reducen a una caja compleja con las entradas y las salidas especificadas por los cuadros rojo y verde.

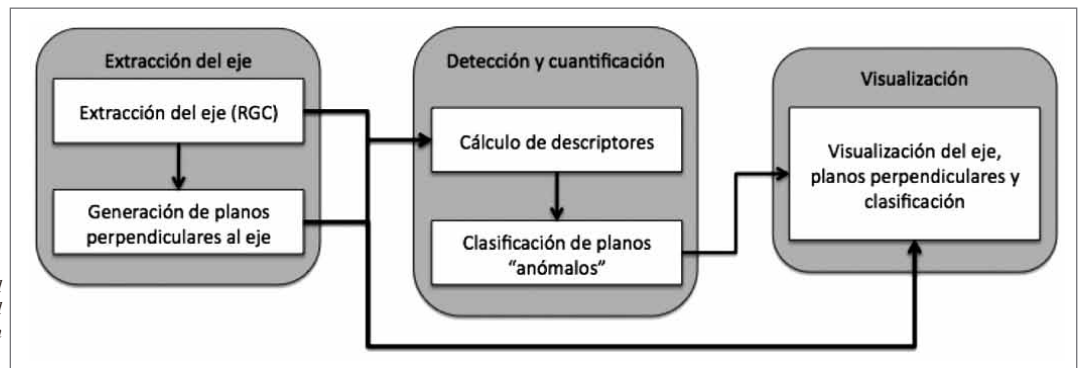


Figura 6. Proceso general del proyecto CreaCoro para extraer el eje, y detectar y visualizar lesiones en las arterias coronarias.

defecto, se tiene el árbol DICOM: paciente | series | imágenes), que se almacenan en bases de datos (SQL), lo cual permite su rápida recuperación.

CreaMaracasVisu, que gestiona la visualización (2D o 3D) de imágenes médicas ofreciendo varios visores y técnicas de interacción que ayudan a analizar las imágenes y los resultados de su procesamiento (por ejemplo, es posible utilizar transformaciones geométricas (rotación, traslación y escalamiento), *rendering* de volumen y de superficie, entre otras, para extraer información relevante del objeto estudiado (adicionalmente, se pueden mostrar diferentes cortes 2D y vistas 3D del mismo objeto, y se pueden combinar objetos binarios [típicamente, los resultados de la segmentación] con la imagen original dentro de la misma escena).

Proyecto CreaCoro: CreaTools para estudio de las arterias coronarias

El proyecto CreaCoro, que hace uso de la plataforma CreaTools, tiene como objetivos la extracción del eje del árbol coronario y la detección de anomalías a lo largo de este para asistir el diagnóstico de enfermedades de las arterias coronarias (EAC) en imágenes de tomografía axial computarizada (TAC). A continuación, se describe el uso de CreaTools para obtener los objetivos mencionados; además, se explica el proceso llevado a cabo en tres etapas: extracción del eje, detección de anomalías y su respectiva visualización (figura 6).

El eje central es extraído usando una estrategia basada en el algoritmo de Dijkstra para encontrar árboles de recubrimiento mínimo (9). Este algoritmo es útil en problemas de grafos donde es necesario

el cálculo del camino más corto entre cualquier par de puntos (camino de costo mínimo).

En el caso de la extracción del eje de una arteria a partir de imágenes de TAC, el grafo está definido por los vóxeles de la imagen (por ejemplo, los nodos o los vértices del grafo) y por una función que establece qué tan centrado se halla un punto dentro de un vaso (criterio *vesselness*). Para cada vóxel se define el costo de pasar sobre este; se espera un costo bajo en el centro de las arterias y un costo alto en los bordes. Un costo alto restringe la extracción del eje en el interior de la arteria. Dicha estrategia básica se ha utilizado en varios trabajos preliminares (10,11). El reto principal es la definición de la función de costo. En este caso se usa la función de *vesselness* propuesta en Gülsün (12), donde se hace un análisis del gradiente de la imagen a lo largo de rayos concéntricos de la arteria para encontrar el radio del cilindro que mejor la modele.

Específicamente, el algoritmo para extraer los ejes comienza con la ubicación de dos puntos por parte del usuario (los puntos deben estar en cualquier parte en el interior de la arteria), desde los cuales dos árboles de recubrimiento mínimo, usando la función de costo descrita anteriormente, son extraídos paralelamente, hasta que entran en colisión; entonces, el camino más corto entre los dos puntos iniciales representa el eje central de la arteria (figura 7).

Con los puntos del eje extraído se calculan planos perpendiculares al eje, que permiten visualizar y analizar una arteria a través de cortes transversales a lo largo de esta. A partir de dichos cortes se genera un nuevo volumen que contiene la arteria “desenrollada”. Esta técnica es llamada *Curved Planar Reformation (CPR)*.

Los algoritmos para la extracción del eje y la obtención de los planos perpendiculares son desarrollados en C++ haciendo uso de las herramientas ITK y VTK. Posteriormente, tales algoritmos son integrados en CreaTools como cajas negras, las cuales pueden ser parametrizables por el usuario. Adicionalmente, se desarrolló una caja negra, llamada CPRWidget, que integra la visualización del eje, los planos perpendiculares a este, la imagen original (la cual puede mostrarse a través de planos multiplanares, como un volumen o una superficie renderizada) y el CPR (figura 8).

Una vez extraído el eje y obtenido el CPR, se procede a realizar un análisis 2D sobre cada uno de los cortes del CPR y a buscar propiedades particulares sobre la región de interés (ROI, por sus siglas en inglés) que permitan diferenciar los cortes sanos y los cortes patológicos. Dichas características diferenciadoras son llamadas *descriptores*.

Se espera que las regiones “sanas” vistas en los planos perpendiculares al eje presenten una geometría cercana a la de un círculo (muy simétrica) y que su intensidad (en nivel de gris) tenga un comportamiento gaussiano (intensidad alta en el centro y que decae hacia las paredes de la arteria), mientras que las regiones “patológicas” son menos simétricas y su distribución de intensidad no tiene una forma gaussiana.

Entre los descriptores implementados se encuentran Core (13), Flux y MFlux (14), los cuales buscan obtener medidas sobre qué tan centrada está la respuesta de gradientes direccionales de rayos lanzados desde el punto del eje a la pared arterial. En cortes sanos se espera una respuesta alta, mientras que en cortes patológicos se espera una respuesta baja.

Otro descriptor implementado es el *de anillos concéntricos* (15), el cual busca capturar los perfiles de intensidad y la simetría axial de cortes normales, e integra la respuesta de varios anillos de diferentes radios alrededor del punto central (figura 9). También se utilizan como descriptores:

- Los valores propios de la matriz hessiana: Método basado en la derivada de segundo orden que provee información sobre

la geometría de los vasos realizando un análisis de la simetría de la región transversal.

- La métrica de Ribbon, que, combinando integración y diferenciación, busca establecer un contraste entre el lumen y el fondo.
- La medida de Balón, que pretende determinar la circularidad del vaso.
- Los momentos de inercia, para establecer la simetría de la forma a través de los valores propios de la matriz de inercia.
- Los descriptores de direccionalidad, que proporcionan información sobre la orientación y la escala de la arteria.

Con la información proporcionada por los mencionados descriptores se procede a clasificar los cortes en dos clases: “Sanos” y “patológicos”. La clasificación se hace a través de técnicas de *Aprendizaje de máquinas*, las cuales se basan en modelos, métodos y algoritmos que aprenden a partir de su experiencia utilizando como base herramientas estadísticas, probabilísticas y matemáticas.

Los modelos de aprendizaje usados en este proyecto son el DLD (*Density Level Detection*) (15) y el LPU (*Learning from Positive Unlabeled Data*) (16), basados, a su vez, en *Support Vector Machines (SVM)*. El modelo LPU se basa en realizar la clasificación de los cortes “sanos” a través de ejemplos etiquetados (se etiquetan solo algunos cortes “sanos” de arterias); también, teniendo en cuenta la distribución entre cortes “sanos” (mayor densidad) y “patológicos” (muy poca densidad) que se presentan en las imágenes. El modelo DLD realiza la clasificación sin etiquetas, utilizando la información de dicha distribución.

Para validar el modelo propuesto se tomaron 15 series de imágenes TAC de arterias coronarias, para un total de 53 arterias obtenidas. Estas imágenes fueron adquiridas en dos sitios diferentes: el Erasmus Medical Center (Rotterdam, Holanda) y el Hospital Louis Pradel (Bron, Francia). El tamaño de la matriz en todas las imágenes es de 512×512 . La resolución de las imágenes de Erasmus MC es $0,32 \times 0,32 \times 0,4 \text{ mm}^3$, y la de las imágenes del Hospital Louis Pradel es de $0,37 \times 0,37 \times 0,9 \text{ mm}^3$.

Una vez extraído el eje, se obtuvieron las imágenes de los planos perpendiculares a este, las cuales fueron etiquetadas como normales y patológicas por un radiólogo experto, en el caso de las imágenes provenientes de Francia, y por dos expertos, en el caso de las imágenes provenientes de Holanda. Sobre las mismas imágenes fueron calculados los descriptores, y así fueron construidos los modelos de clasificación.

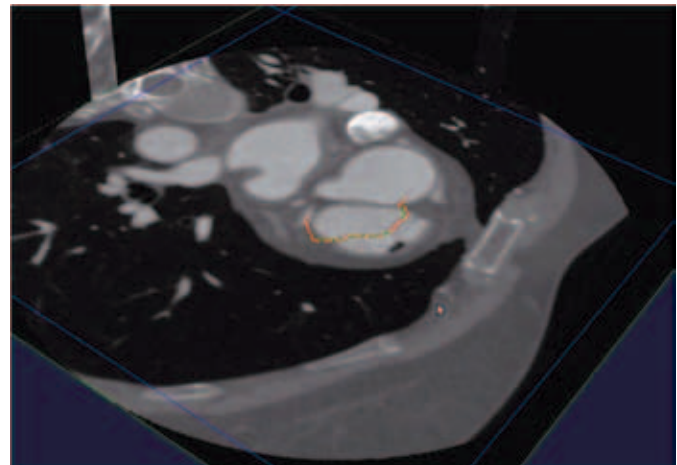


Figura 7. Ejemplo de un eje extraído de una arteria coronaria a partir del algoritmo implementado en CreaTools.

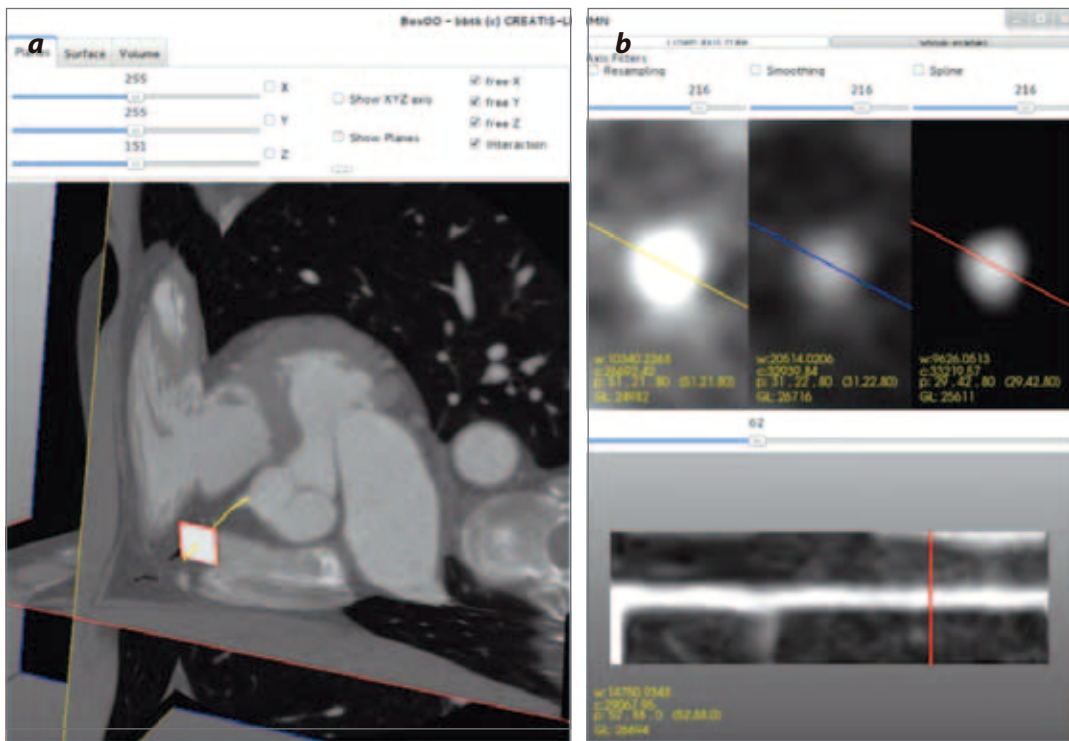


Figura 8. CPRWidget. a) Visualización en 3D de la imagen original a través de planos multiplanares; además, se observan el eje extraído de una arteria coronaria y el plano perpendicular a este. b) En la parte superior se muestran los planos perpendiculares al eje (los tres colores diferentes corresponden a tres posiciones diferentes de los planos a lo largo del eje; en este caso se hallan en el mismo lugar). En la parte inferior se muestra la imagen CPR del eje extraído. Las líneas corresponden a la posición del plano perpendicular en la arteria “desenrollada”.

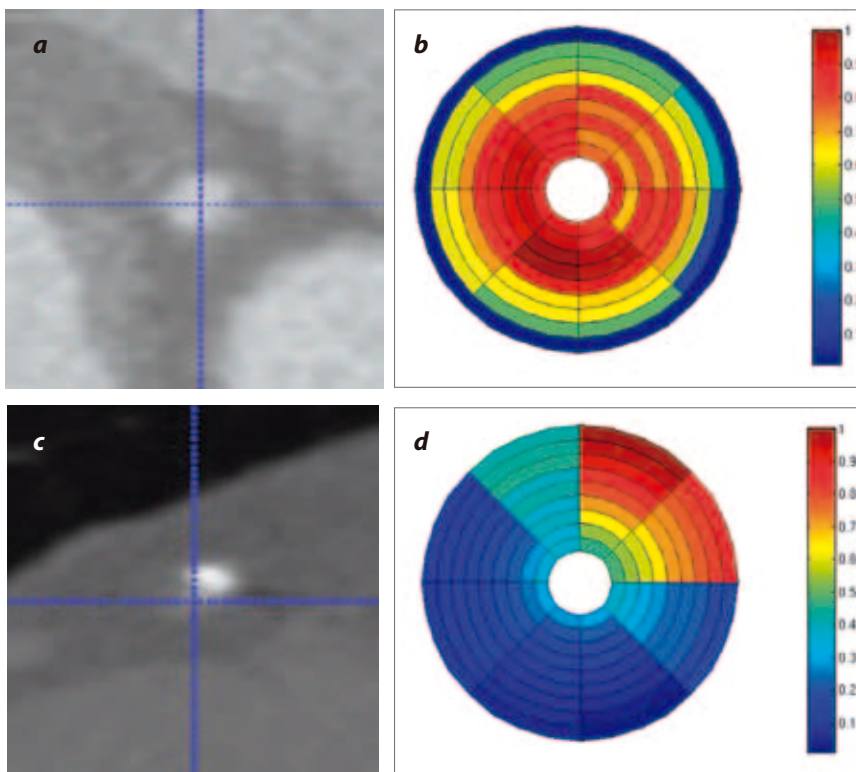


Figura 9. Visualización de la métrica de anillos concéntricos. En el caso de una arteria normal a) los valores más altos son obtenidos en los anillos del centro, y los valores más bajos, en los anillos exteriores b) En el caso de una arteria patológica c) se observa en los anillos d) una diferencia significativa respecto a los valores de b.

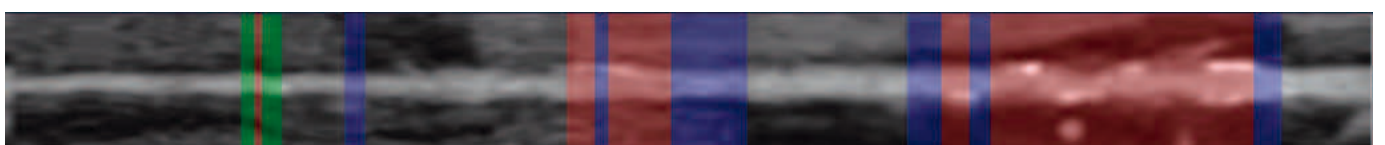


Figura 10. Resultados de la detección de lesiones. Los colores corresponden a los diferentes resultados obtenidos, donde el rojo equivale a VP; el verde, a FN, y el azul, a FP, y donde no hay color equiparable a VN. Los colores se sobrepone en una vista planar curva de las imágenes perpendiculares de diferentes arterias coronarias.

Para evaluar el desempeño de los modelos respecto a la clasificación de anomalías, se especificaron tres medidas:

- Tasa Verdadero Positivo (TVP)= $VP/(VP + FN)$ (1)
- Tasa Verdadero Negativo (TVN)= $VN/(VN + FP)$ (2)
- Tasa Error Balanceado (TEB)= $1 - ((TVP + TVN)/2)$ (3)

Donde: VP (Verdadero Positivo) corresponde al número de imágenes con anomalías clasificadas correctamente; VN (Verdadero Negativo), al de imágenes sin anomalías clasificadas correctamente; FP (Falso Positivo), al de imágenes con anomalías clasificadas incorrectamente; FN (Falso Negativo), al de imágenes sin anomalías clasificadas incorrectamente (figura 10).

Los resultados para el algoritmo DLD fueron: $TPN=86\%$; $TVN=81,2\%$, y $TEB=16,4\%$. Los resultados para el algoritmo LPU fueron: $TPN=83,8\%$; $TVN=86,7\%$, y $TEB=14,2\%$.

A raíz de estos dos algoritmos se pudo observar que el LPU presenta un mejor comportamiento, en la medida en que este no comete errores de clasificación en las bifurcaciones (el DLD estima que una bifurcación es una anomalía, pues no presenta las propiedades de una arteria normal). Además, el desempeño del LPU podría aumentar en la medida en que exista una base de imágenes mucho más grande durante la etapa de entrenamiento.

El trabajo base del presente proyecto, donde se calculan los descriptores y los modelos de aprendizaje mencionados (16), está desarrollado en diversas herramientas de programación (C++ y Matlab), y actualmente el equipo de desarrollo se halla integrando por cada uno de los descriptores y modelos de aprendizaje mencionados en CreaTools como cajas negras y en un solo *pipeline*, para brindar al usuario la posibilidad de parametrizar los algoritmos. Aprovechando las herramientas de visualización de CreaTools, como es el CPRWidget (ejes, MPR, CPR, volumen, superficie), se observarán los resultados de la clasificación en un ambiente integral donde se pueda realizar un análisis completo sobre la información propuesta por la aplicación.

Conclusiones y perspectivas

El ejemplo del proyecto de análisis de arterias coronarias ilustra varias formas de utilizar CreaTools. Las cajas existentes proporcionan las herramientas necesarias para construir prototipos de GUI y aplicaciones de demostración para visualizar imágenes y tareas de procesamiento. Los algoritmos nuevos se encapsulan en cajas negras atómicas de tal forma que puedan ser probados usando las GUI apropiadas.

Las técnicas de visualización e interacción faltantes, al igual que las herramientas correspondientes, se implementan y se agregan, bien sea como cajas nuevas o enriqueciendo las ya existentes. Los *pipelines* que ya han sido exhaustivamente probados se transforman en cajas complejas, y enriquecen así el *toolkit* entero compartido dentro del VPH NoE, y más allá, de acuerdo con la naturaleza de software libre de CreaTools.

El objetivo de esta presentación del proyecto CreaTools es alentar a los investigadores en procesamiento de imágenes para que descarguen, instalen y usen este ambiente de desarrollo. Se quiere crear una comunidad de usuarios que informen sobre fallas, para contribuir así con el desarrollo de nuevas cajas negras y documentar las ya existentes. Al mismo tiempo que se eliminan fallas, se crea la documentación y se agregan nuevas funcionalidades se está trabajando en la interoperabilidad del sistema.

Inicialmente se desea garantizar la conexión entre bbtck y Slicer3D/CTK, e incluir a Matlab en los *workflows*. El segundo eje de desarrollo se relaciona con la computación distribuida; se desea proveer los *workflows* de procesamiento de imágenes con un acceso directo a recursos computacionales, como *clusters* y *grids*. El tercer eje es la realidad virtual; se intenta facilitar la interacción con imágenes 3D mediante el uso de visión estereoscópica y dispositivos de control remoto.

Agradecimientos

Este trabajo ha sido financiado parcialmente por las subvenciones ECOS Nord C11S01, Uniandes Interfacultades 06-2010 y Colciencias 1204-519-28996, al igual que por la Comunidad Europea, vía VPH STREP Thrombus FP7-ICT-2009-6-269966.

Referencias

1. MeVis M. S. MevisLab-Medical image processing and visualization [internet]. 2012 [citado: 1° de agosto de 2012]. Disponible en: <http://www.mevislab.de/>
2. VPH. Virtual Physiological Human, network of excellence [internet]. 2012 [citado: 1 de agosto de 2012]. Disponible en: <http://toolkit.vph-noe.eu/home/tools/imaging.html>
3. Kitware Inc. Página principal [internet]. 2012 [Citado: 1° de agosto de 2012]. Disponible en: <http://www.kitware.com/>
4. Qt Developer Network. Página principal [internet]. 2012 [citado: 1° de agosto de 2012]. Disponible en: <http://qt-project.org/>
5. CreaTools. Página principal [internet]. 2012 [citado: 1° de agosto de 2012]. Disponible en: http://www.creatis.insa-lyon.fr/site/en/CreaTools_home
6. Harvey M, Deitel PJ. C++ How to Program, 4th ed. s. d.: Prentice Hall. (2003).
7. Thrombus VPH. Página principal [internet]. 2012 [citado: 1° de agosto de 2012]. Disponible en: <http://www.thrombus-vph.eu/>
8. CreaTools. Promoting economical useful and maintained software for the higher education and the research communities [internet]. 2012 [citado: 1° de agosto de 2012]. Disponible en: <https://www.projet-plume.org/en/relier/creatools>
9. Dijkstra EW. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959;1:269-271.
10. Wink O, Frangi AF, Verdonck B, et al. 3D MRA coronary axis determination using a minimum cost approach. *Magn Reson Med*. 2002;47:1169-75.
11. Flórez Valencia L, Vincent F, Orkisz M. Fast 3D pre-segmentation of arteries in computed tomography angiogram. *Computer Vision Graphics*. 2006;32:361-6.
12. Gülsün MA, Tek H. Robust vessel tree modeling. *MICCAI*. 2008;5241:602-11.
13. Fridman Y. Extractin branching object geometry via Cores. PhD thesis, University of North Carolina at Chapel Hill, 2004.
14. Lesage D, Angelini E, Bloch I, et al. Design and study of flux-based features for 3d vascular tracking. Documento presentado en IEEE International Symposium on Biomedical Imaging: From Nano to Macro. 2009, Boston, Estados Unidos.
15. Zuluaga MA, Magnin IE, Hernández Hoyos M, et al. Automatic detection of abnormal vascular cross-sections based on density level detection and support vector machines. *Int J Comput Assist Radiol Surg*. 2011;6:163-74.
16. Zuluaga MA, Hush D, Delgado Leyton EJ, et al. Learning from only positive and unlabeled data to detect lesions in vascular CT images. *MICCAI*. 2011;14:9-16.

Correspondencia

Marcela Hernández Hoyos
Grupo Imagine, Grupo de Ingeniería Biomédica
Universidad de los Andes
Carrera 1ª Este No. 19A-40. Of. ML816.
Bogotá, Colombia
marc-her@uniandes.edu.co

Recibido para evaluación: 19 de julio de 2012

Aprobado para publicación: 15 de agosto de 2012