

***DIP Lab*: Herramienta de apoyo pedagógico en el área de procesamiento de imágenes digitales.**

Andrés Córdoba M.

Departamento. de Ingeniería de Sistemas y Computación
Universidad de los Andes, Bogotá, Colombia
an-cordo@uniandes.edu.co

y

William A. Romero R.

Departamento. de Ingeniería de Sistemas y Computación
Universidad de los Andes, Bogotá, Colombia
wil-rome@uniandes.edu.co

Resumen

Este artículo describe *DIP Lab* (*Digital Image Processing Laboratory*), una herramienta Open Source de apoyo pedagógico en el área de procesamiento de imágenes digitales. La originalidad de la herramienta radica en 3 características principales: 1) *DIP Lab* maneja un modelo gráfico para la visualización y persistencia de la secuencia de operaciones de un proceso, a través de un gráfico. 2) Gestión de proyectos. 3) Integración de unidades temáticas con actividades y enlaces teóricos como material pedagógico dentro de la enseñanza práctica en el área. Este desarrollo se realizó en conjunto con el grupo de visualización y procesamiento imágenes: *IMAGINE*, del departamento de Ingeniería de Sistemas y Computación de la Universidad de los Andes.

Palabras claves: Procesamiento de imágenes, Grafo de procesamiento, Material de apoyo pedagógico

1. INTRODUCCIÓN

El procesamiento digital de imágenes comprende el análisis, diseño y desarrollo de algoritmos para la manipulación de imágenes digitalizadas, con el propósito de poder realizar transformaciones tales como: correcciones y/o ajustes de brillo, contraste, intensidad; transformaciones geométricas como escalamiento y rotación; aplicación de filtros para resaltar información o reducción de ruido en la imagen; segmentación en regiones y detección de bordes, entre otras. Un factor importante en el procesamiento de una imagen digital es el software utilizado para este fin. Para ciertos problemas, utilizar una aplicación que ofrezca un conjunto básico de operaciones es suficiente. Pero en la mayoría de los casos, es necesario desarrollar software a la medida para realizar operaciones específicas de acuerdo a los requerimientos del problema.

Estos requerimientos se definen en términos de la información que se desea obtener de la imagen, dejando en un plano secundario, en el caso de aplicaciones genéricas, requerimientos como: la interfaz de usuario, la persistencia de la secuencia de operaciones de un proceso, la administración de los archivos de acuerdo a un proyecto, entre otros. En el caso de los desarrollos personalizados este tipo de requerimientos son de vital importancia. Por ejemplo, las aplicaciones desarrolladas para el análisis de imágenes médicas son diseñadas para que puedan ser fácilmente utilizadas por los médicos como herramienta de soporte dentro un diagnóstico.

A nivel de enseñanza en esta área, de igual manera, es importante contar con una aplicación que permita afianzar los conocimientos teóricos a través de la práctica para desarrollar destrezas propias en la materia. Dentro de la revisión de aplicaciones para el procesamiento de imágenes, no se encontró ninguna diseñada como herramienta de soporte a este nivel, por el contrario se encontró software especializado en diferentes campos de aplicación tales como la medicina, la astronomía, etc. La necesidad de ofrecer un ambiente en donde se integren las funcionalidades de procesamiento de imágenes con elementos de soporte al aprendizaje fue la motivación para desarrollar esta herramienta.

Este documento describe *DIP Lab*, una aplicación desarrollada como herramienta de apoyo pedagógico en el área de procesamiento de imágenes digitales. La originalidad de la herramienta radica en 3 características principales: 1) *DIP Lab* maneja un modelo gráfico para la visualización y persistencia de la secuencia de operaciones de un proceso, a través de un grafo. La representación visual del proceso permite un mejor seguimiento y comprensión; docente y alumno convergen al conocer el orden y los detalles del proceso en el grafo. 2) Gestión de proyectos. 3) Integración de unidades temáticas con actividades y enlaces teóricos como material pedagógico dentro de la enseñanza práctica en el área.

Este artículo comienza con la especificación de los requerimientos planteados para la construcción de ésta aplicación; éstos serán empleados como criterios de comparación con desarrollos similares. Este estudio permitirá evaluar la funcionalidad del programa, tener referentes de los requerimientos genéricos de este tipo de aplicaciones y resaltar las innovaciones propuestas. Posteriormente, se realiza una exposición general de la arquitectura, detalles de implementación. Seguido de la descripción de la herramienta y la interacción con el usuario. Finalmente se plantean trabajos futuros para expandir las funcionalidades y la cobertura de este trabajo.

2. REQUERIMIENTOS

A continuación se presentan los requerimientos que se definieron durante el análisis del problema de la enseñanza y el aprendizaje en el área de procesamiento de imágenes.

2.1. Requerimientos funcionales

- **RF. 1 Proveer los servicios estándar de una aplicación de procesamiento de imágenes:** Leer y escribir en la mayoría de formatos posibles(PCX, JPG, BMP, entre otros) realizar operaciones sobre el histograma de la imagen; utilizar filtros de realce, suavizado, reducción de ruido; segmentación, aplicación de operaciones de morfología matemática, entre otros servicios
- **RF. 2 Administrar la persistencia de los procedimientos realizados por el usuario:** Permitir el almacenamiento y la recuperación de la secuencia de operaciones realizadas, los parámetros empleados y los respectivos resultados. La representación y visualización de este historial se realiza mediante un grafo, que visualiza y modifica el usuario en la interfaz gráfica.
- **RF. 3 Administrar proyectos:** La herramienta debe manejar la noción de proyecto. Un proyecto se compone del conjunto de imágenes fuente, un grafo de procesamiento asociado y un conjunto de imágenes resultado.
- **RF. 4 Integrar un componente para la ejecución de actividades docentes:** Se requiere que esta aplicación sea una herramienta pedagógica y de apoyo en los cursos de procesamiento de imágenes. Para este fin, debe permitir la visualización de actividades prácticas propuestas por el docente para que sean realizadas por los alumnos con la aplicación, que permitan el desarrollo de habilidades en la aplicación de las técnicas y conceptos teóricos en la solución de problemas reales.

2.2. Requerimientos no funcionales

- **RNF. 1 Software independiente de la plataforma:** La aplicación final debe poderse ejecutar en cualquier sistema operativo.
- **RNF. 2 Aplicación de código abierto:** Es indispensable poder contar con un desarrollo que pueda ser enriquecido con los aportes de los profesores, estudiantes y/o demás personal interesado en este trabajo.
- **RNF. 3 Integración con The Visualization ToolKit (VTK):** La herramienta debe proveer integración con esta plataforma estándar Open Source para procesamiento de imágenes[1].
- **RNF 4. Acceso a través de Internet:** Se desea que la herramienta pueda invocarse a través de Internet.

3. ESTUDIO DE APLICACIONES SIMILARES

En este aparte se describen algunas aplicaciones para el procesamiento de imágenes, encontradas en la etapa de estudio del estado del arte.

- **Madena:** Es una herramienta *freeware* para plataforma *MAC OS* con énfasis en imágenes médicas. Puede trabajar con todos los archivos de imagen compatibles con QuickTime, DICOM y ACR/NEMA. Todas las funcionalidades de esta aplicación fueron diseñadas para realizar diagnósticos médicos[2].
- **Lucis:** Esta aplicación fue desarrollada por la compañía *Image Content Technology* para plataformas Windows. El potencial de esta aplicación es el conjunto de filtros configurables, dependiendo de la técnicas y tecnologías de adquisición de imágenes digitales: Microscopia electrónica, rayos X, CT, Adquisición desde telescopios etc[3].
- **ImageJ:** Software de uso libre implementado en JAVA por el *National Institutes of Health* (NIH). Para su desarrollo se usó como referencia la aplicación *NIH Image*. NIH Image es un programa de procesamiento y análisis de imágenes de dominio público para Macintosh, desarrollado en *Research Services Branch* (RSB) del *National Institute of Mental Health* (NIMH), que hace parte del NHI[9].
ImageJ a diferencia de *NIH Image* puede ejecutarse en cualquier plataforma y además puede incorporarse parte de sus funcionalidades en Applets y Servlets, permitiendo su invocación desde páginas web[3].
- **IrfanView:** Esta herramienta *freeware* desarrollada por *Irfan Skiljan* para plataformas Windows y destinada a uso no comercial. Se caracteriza por ser una aplicación pequeña, compacta y rápida. IrfanView se concentra en la presentación de imágenes más que en su procesamiento[6].
- **vtkGUI** Programa de uso libre escrito en *TCL/IncrTCL*. Su desarrollo fue inspirado por *vtkPipeline* y el *vtkShow* con el fin de permitir el montaje de una tubería de VTK con VTK. Es una interface gráfica y sencilla que manipula el concepto de tubería[7].

El siguiente cuadro muestra una comparación entre las funcionalidades que cubre cada una de las aplicaciones descritas, teniendo en cuenta los requerimientos mencionados en el numeral 2.

Puede observarse que ninguna aplicación cubre en su totalidad los requerimientos deseados. *DIP Lab* es una propuesta alternativa a las aplicaciones presentadas.

4. LA HERRAMIENTA

En esta sección se presentan los conceptos en los cuales se fundamenta la herramienta, el desarrollo de la misma y la interacción del usuario con la aplicación.

	Malena	Lucis	ImageJ	InfranView	vtkGUI
RF. 1	x	x	x	x	x
RF. 2	-	-	-	-	x
RF. 3	-	-	-	-	-
RNF. 1	-	-	x	-	x
RNF. 2	-	-	x	-	x
RNF. 3	-	-	-	-	x
RNF. 4	-	-	x	-	-

Cuadro 1: Evaluación de aplicaciones similares.

4.1. Conceptos

4.1.1. Grafo de procesamiento

El grafo de procesamiento es la estructura y el modelo visual que emplea *DIP Lab* para representar el orden y la secuencia del procesamiento definido por el usuario, mediante un grafo dirigido. Este grafo se compone de 3 tipos de nodos:

- **Nodo fuente:** Origen de los datos, en este caso, una imagen digital;
- **Nodo operación:** Representa la acción a realizar con el nodo fuente. Este nodo tiene parámetros determinados por el tipo de operación. Por ejemplo, en el caso de una umbralización el usuario de introducir el valor del nivel de gris (umbral).
- **Nodo resultado:** Es el producto de la acción sobre el nodo fuente.

En resumen, los nodos representan los objetos y las operaciones del procesamiento y los arcos indican el flujo de los datos dentro de este (figura 1). Esta representación permite conocer rápida y detalladamente un proceso, no sólo para quien esta realizando el procesamiento, si no también para un tercero al que se le presente el grafo ya construido.

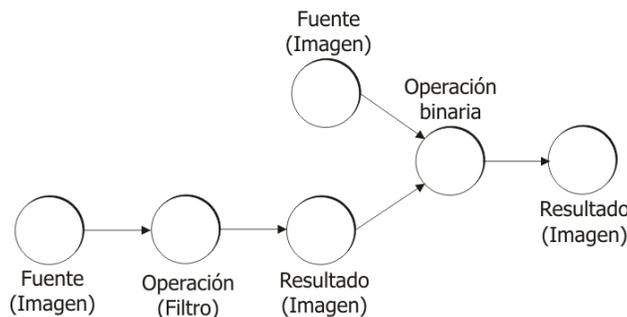


Figura 1: Grafo de procesamiento

4.1.2. Proyecto

DIP Lab se concibió para manejar proyectos. Un proyecto es la unidad de procesamiento que puede ser una actividad docente simple, talleres prácticos o proyectos de mayor complejidad. El usuario crea un proyecto, al cual incorpora un conjunto de imágenes fuente, que forman la materia prima del procesamiento. Este conjunto tiene asociado el grafo de procesamiento, que representa el orden y las operaciones realizadas con las fuentes. El usuario puede compartir proyectos. En este caso sólo es necesario el conjunto de imágenes fuente y el grafo de procesamiento, dado que el conjunto de imágenes resultado se puede obtener procesando el grafo.

4.1.3. Material pedagógico

El material pedagógico corresponde a los contenidos propuestos por el docente para la realización de talleres, prácticas o proyectos. De acuerdo a la experiencia adquirida dentro del curso de procesamiento de imágenes de la Universidad de los Andes, se propone la siguiente estructura básica para los contenidos del material pedagógico:

- Conjunto de imágenes fuente.
- Actividades propuestas para realizar con la herramienta sobre el conjunto de imágenes fuente.
- Conceptos teóricos relacionados y los respectivos enlaces desde las actividades, para que el estudiante durante el desarrollo pueda acceder a esta información.
- Ayuda funcional del programa
- Ejercicio(s) de síntesis. Este es un ejercicio propuesto en donde se integran todos los conceptos practicados en las actividades previas.

La edición y producción se realiza en un documento LaTeX, que permite una manipulación y visualización sencilla de los contenidos matemáticos relacionados. Adicionalmente permite exportar a documentos de hipertexto, ejemplo archivos pdf y html. Aunque *DIP Lab* no ofrece un componente para la edición y producción de este material, este puede realizarse mediante editores como TeXnicCenter o Lyx. Este material es desplegado por *DIP Lab* dentro del mismo ambiente de trabajo del usuario, evitando el uso de aplicaciones externas.

5. DESARROLLO DE *DIP Lab*

Para el desarrollo de *DIP Lab* se adoptó la metodología Extreme Programming que permite un monitoreo constante por parte del grupo IMAGINE, así como la obtención de resultados funcionales en corto tiempo.

5.1. Aspectos de diseño

DIP Lab se diseñó modularmente con propósito de hacer un sistema extensible, de fácil actualización y mantenimiento. Se definió la implementación en *JAVA*, para cumplir con **RNF. 1**. Para la selección del componente de visualización de gráficos se evaluarón los paquetes: Visualizing Graphs with Java (VGJ) [10], Java Universal Network/Graph Framework (JUNG) [11], The Graph Visualization Framework (GVF) [12], JGraphT [13] y Java [14] Graph Visualization and Layout (JGraph); adoptando *JGraph* como la librería a usar, por ser la de mayor flexibilidad e independencia de otras librerías.

5.2. Arquitectura

En la figura 2, se ilustran los componentes principales de *DIP Lab*:

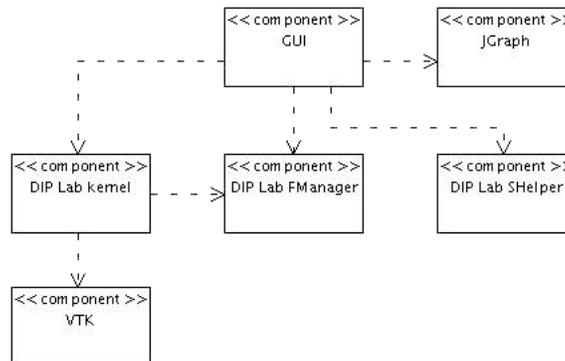


Figura 2: Componentes *DIP Lab*

- **GUI:** Interface gráfica de usuario; Hace uso de la librería JGraph.
- **DIPLab Kernel:** Módulo encargado de la lógica del grafo de procesos y su relación con VTK.
- **DIPLab FManager:** Componente para la persistencia y administración de archivos de un proyecto.
- **DIPLab SHelper:** Publicador de actividades y material de ayuda.
- **VTK:** Librería que conoce y define las estructuras que representan las imágenes y los procesos con estas.

6. INTERACCION CON *DIP Lab*

La interfaz gráfica de *DIP Lab* se compone de 5 áreas(figura 3), éstas se describen a continuación:

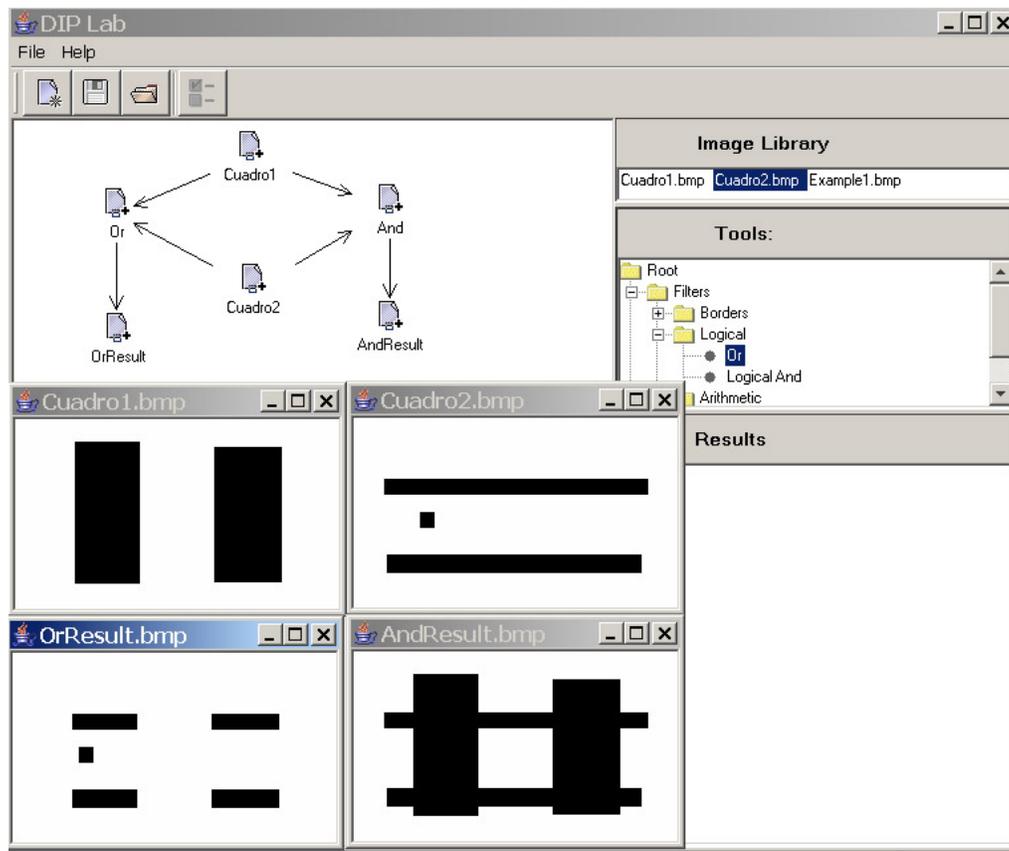


Figura 3: Interfaz gráfica

- **Workspace:** Espacio de trabajo del usuario. Aquí el usuario construye el grafo de procesamiento arrastrando los elementos a incorporar de las áreas laterales.
- **Image Library:** Aquí se listan las imágenes disponibles para ser utilizadas dentro del procesamiento. El usuario debe importar a esta área las imágenes de interés. Estas imágenes corresponden a los nodos fuente en el grafo.
- **Nodes:** Esta caja de dialogo permite la selección de la operación a incorporar dentro del área de trabajo, representada a través de un nodo de operación.
- **Results:** En este espacio se listan los resultados obtenidos después de una operación.

- **Monitor:** Permite la visualización de las imágenes. Se puede invocar el monitor realizando clic sobre el nodo resultado o en la lista de *Results*.

El usuario puede adicionalmente invocar el navegador de actividades y material de ayuda. Este componente permite la visualización de hipertextos en el mismo ambiente de trabajo de la aplicación. Los nodos de operación presentan una opción de ayuda, que invoca al navegador visualizando los contenidos teóricos de la operación al igual que los pasos para su ejecución en la herramienta. Esta funcionalidad permite, a nivel de enseñanza, la publicación de material para que el alumno desarrolle con la aplicación en un mismo ambiente.

7. CONCLUSIONES

DIP Lab es una aplicación desarrollada como herramienta de apoyo pedagógico en el área de procesamiento de imágenes digitales. Cumpliendo con los requerimientos especificados para su construcción, presenta una arquitectura modular que permite fácil actualización y mantenimiento, incorporando una plataforma estándar, VTK. *DIP Lab* administra un modelo gráfico para la visualización y persistencia de la secuencia de operaciones de un proceso, a través de un gráfico; También ofrece la posibilidad de incorporar unidades temáticas con actividades y enlaces teóricos que permiten que sea utilizado como material pedagógico dentro de la enseñanza en el área. Frente a otras aplicaciones, ofrece todo el soporte de VTK, ampliando la cobertura de la herramienta. De igual manera la interface de usuario y el modelo visual sobre la secuencia de operación, proveen un ambiente sencillo y práctico que puede ser utilizado como una aplicación de alto nivel. Una aplicación secundaria pero de gran aporte a la comunidad de procesadores de imágenes es la interfaz gráfica a VTK. Como trabajo futuro se tiene planeado el desarrollo un paquete básico de actividades que formen parte de la distribución estándar de *DIP Lab* y la validación del requerimiento de invocación de la herramienta a través de Internet.

Agradecimientos

A Marcela Hernandez profesora del departamento de Ingeniería de Sistemas y Computación por la concepción de la aplicación, el apoyo logístico para su desarrollo y la elaboración de este documento.

Referencias

- [1] *The Visualization ToolKit (VTK)* [Online] Disponible: <http://public.kitware.com/VTK/>
- [2] *Madena* [Online] Disponible: <http://radonc.usc.edu/USCRadOnc/Madena/Madena.html>
- [3] *Lucis* [Online] Disponible: <http://www.imagecontent.com/>
- [4] *ImageJ Image Processing and Analysis in Java* [Online] Disponible: <http://rsb.info.nih.gov/ij/index.html>
- [5] *ImageJ Disclaimer* [Online] Disponible: <http://rsb.info.nih.gov/ij/disclaimer.html>
- [6] *IrfanView* [Online] Disponible: <http://www.irfanview.com/>
- [7] *VtkGui Visual Programming Tool for The visualization Toolkit* [Online] Disponible: <http://visit.cineca.it/vtkGUI/>
- [8] *Web site of the Center for Image Processing in Education* [Online] Disponible: <http://www.evisual.org/homepage.html>
- [9] *About NIH Image* [Online] Disponible: <http://rsb.info.nih.gov/nih-image/about.html>
- [10] *Visualizing Graphs with Java (VGJ)* [Online] Disponible: <http://www.eng.auburn.edu/departament/cse/research/graphdr>
- [11] *Java Universal Network/Graph Framework (JUNG)* [Online] Disponible: <http://jung.sourceforge.net/>
- [12] *The Graph Visualization Framework (GVF)* [Online] Disponible: <http://gvf.sourceforge.net/>

- [13] *JGraphT* [Online] Disponible: <http://jgrapht.sourceforge.net/>
- [14] *JGraph* [Online] Disponible: <http://jgraph.sourceforge.net/>
- [15] R. González, P. Wintz. *Digital Image Processing*. 2da. edición, Addison-Wesley, 2002.
- [16] M. Sonka, V. Hlavac, R. Boyle. *Image processing, analysis and machine vision*. Brook/Cole Publishing Co., 1999.