

# **Sistema de Visualización de Imágenes Médicas:**

## **SVIM**

### *Especificación de Requerimientos del Sistema (ERS)*

## **1 Introducción**

Este documento presenta la especificación de requerimientos del sistema SVIM en su versión 2015-C2. Contiene una introducción general que describe el dominio del problema y las especificaciones de cada uno de los componentes del sistema.

Las imágenes digitales tienen un amplio uso en medicina. Sus fuentes de creación son diversas: por ultrasonido, por reflexión de luz, por transmisión de rayos-x, etc. Así, por ejemplo, se tiene la imagen de tomografía axial computada, la cual se obtiene haciendo pasar un conjunto de rayos-x a través del cuerpo humano y midiendo el nivel de transmisión de cada rayo. Esto permite construir una imagen en dos dimensiones con la estructura de una sección interna del cuerpo.

En este sistema se busca poder visualizar imágenes de diferentes tipos, a partir de información almacenada en archivos con un formato unificado para todos los tipos, para luego aplicar algoritmos de procesamiento que permitan obtener características particulares de la imagen.

## **2 Requerimientos de la imagen**

El sistema deberá poder manipular imágenes en la memoria de la computadora, accediendo y modificando cada uno de los atributos que contenga. Estas imágenes se representan mediante una matriz de píxeles. Un píxel es un rectángulo que tiene un determinado color. El color en la computadora se puede modelar como una combinación de tres colores elementales: rojo, verde y azul (RGB). El color de un píxel se obtiene mediante una proporción de estos tres colores. Así, por ejemplo, el color amarillo se obtiene combinando el rojo con el verde.

Para que el procesamiento de la imagen sea más sencillo se puede modelar a cada color como un valor de punto flotante con un rango de cero a uno, donde 0 significa que ese tono no aporta color y 1 significa que ese tono aporta un

100% del color. Con esa forma de representación, el valor RGB para el color amarillo sería de (1.0, 1.0, 0.0), es decir 100% de rojo, 100% de verde y 0% de azul. La intensidad de un pixel da una idea del brillo que tiene, y se puede estimar sencillamente como el promedio de los valores de sus componentes de color. Sobre los pixeles deben poder realizarse las operaciones aritméticas básicas para poder modificarlos con los algoritmos de procesamiento.

Las imágenes médicas tienen además otros datos que son importantes a la hora de examinarla. Éstos se denominan Metadatos. Los metadatos tienen un código y un valor asociado al código, cada uno representado como una cadena de caracteres. Por ejemplo, se podría asociar a una imagen el nombre de la persona a la cual se le está haciendo el estudio; el metadato tendría como clave “nombre” y como valor asociado “Juan Pérez”. Como restricción, los metadatos no pueden tener códigos repetidos. Además, se debe poder buscar fácilmente un valor a partir de su código.

### 3 Requerimientos de los archivos

El sistema debe contemplar la carga en memoria de los datos de una imagen almacenados en un archivo. De la misma forma, deberá poder guardarse en un archivo una imagen que esté en la memoria.

El sistema debe estar preparado para los cambios de los formatos de archivos. En un principio se requiere que se pueda trabajar con archivos de formato MIF (Medical Image Format), el cual fue diseñado exclusivamente para este sistema. No se descarta la opción de poder trabajar con otros formatos, como por ejemplo JPG, PNG o TIFF.

Los archivos MIF contienen al inicio una cabecera, escrita en modo texto, la cual contiene sólo dos renglones. El primer renglón contiene información de la dimensión de la imagen: cantidad de pixeles a lo ancho, cantidad de pixeles a lo alto, medida física del ancho de la imagen, medida física del alto de la imagen y, por último, la unidad de la medida física. Estos cinco valores se encuentran separados por un caracter de punto y coma (;). El segundo renglón tiene los metadatos de la imagen, también separados por un caracter de punto y coma (;). Cada metadato, a su vez, tiene la clave, el caracter de dos puntos (:) y luego el valor asociado a la clave. La cantidad de metadatos es variable y no hay un formato fijo definido. En caso que no existan metadatos este renglón estará vacío. El siguiente párrafo es un ejemplo de la cabecera de un archivo MIF:

```
210;300;42.2;60.3;mm  
fecha:2015/07/04;nombre:Bruce Willis;equipo:T1000;tipo:XRay
```

En este ejemplo, la imagen tiene 210 pixeles de ancho, 300 pixeles de alto y mide 42,2 por 60,3 milímetros, respectivamente. Además, la fecha de captura es del 4 de Julio de 2015, el paciente es Bruce Willis, el equipo con el que se capturó la imagen es el T1000 y el tipo de imagen es de Rayos-X.

Luego de la cabecera, están codificados los píxeles de la imagen en un formato binario. Cada píxel está codificado como una terna RGB, donde cada componente de color es un valor entero de 16 bits sin signo. La disposición de los píxeles es la siguiente: el primer píxel corresponde al de la esquina superior izquierda de la imagen, siguiéndole el segundo píxel de la primer fila y así sucesivamente hasta completar la primer fila. Luego están los píxeles de la segunda fila de la misma forma, y luego las filas restantes hasta el final del archivo.

## **4 Requerimientos de visualización**

El sistema deberá permitir la graficación de una imagen utilizando la librería OpenGL. La ventana de graficación podrá ser flotante o estar inmersa en otra ventana con otros componentes gráficos. La imagen nunca deberá perder la relación de aspecto (no deberá deformarse), por más que cambie el tamaño de la ventana.

El sistema sólo deberá graficar una única imagen, pero debe contemplarse que este requerimiento podría cambiar en el futuro y podría ser necesario graficar más de una imagen.

## **5 Requerimientos de procesamiento**

El sistema debe permitir que se apliquen algoritmos de procesamiento en imágenes. Un procesamiento cambia los valores de los píxeles de la imagen produciendo una nueva imagen. Una vez aplicado el procesamiento, se deberá reemplazar la imagen original por la procesada.

Los algoritmos de procesamiento pueden ser diversos. En este momento se requieren sólo dos: un filtro de color y un detector de estructuras.

### **Filtro de color:**

Este procesamiento reemplaza cada componente de color RGB por el valor de la intensidad del píxel, exceptuando los píxeles que tienen una similitud con un color base definido. En este caso, la similitud entre dos colores se da cuando cada componente homóloga tiene una diferencia en valor absoluto no mayor a 0,05.

### **Detector de estructuras:**

El procesamiento de detección de estructuras parte de un píxel inicial que pertenezca a la estructura a detectar y sobre él aplica el algoritmo del pintor. La estructura debe ser uniforme (los píxeles deben estar contiguos) y de una intensidad similar. La imagen resultante para este proceso debe tener un fondo negro y la estructura detectada con color verde.

## **6 Requerimientos no funcionales del sistema**

El sistema tendrá una interfaz de consola, excepto para la visualización de la

imagen, en la cual se utilizarán las librerías de OpenGL y Qt para su implementación.

La implementación del sistema deberá realizarse utilizando el lenguaje de programación C++ y la metodología de la Programación Orientada a Objetos.

El sistema deberá lograr cada operación en un tiempo no mayor a 2 segundos.

El sistema deberá contemplar los casos en que el archivo de imagen no se encuentre disponible para cargarlo en memoria.

El sistema deberá funcionar en un entorno Linux sobre procesadores de la familia x86 o similar.

## 7 Casos de uso

La siguiente lista presenta los casos de uso fundamentales que deberá tener el sistema:

<b>CU</b>	<b>Descripción</b>
01	Cargar imagen en memoria.
02	Guardar imagen en archivo.
03	Mostrar atributos de la imagen.
04	Modificar atributos de la imagen.
05	Visualizar la imagen.
06	Aplicar procesamiento a la imagen.