

Documento de Diseño y Plan de Pruebas

Título: Generación de Proyecciones 2D en Volúmenes de Imágenes PGM

Autores: Jeison Alfonso, Douglas Diaz

Fecha: 06/03/2025

Curso: Estructuras de Datos

Profesor: John Corredor

1. Resumen

¿Qué problema se plantea?

El problema abordado en este trabajo es la necesidad de generar proyecciones en 2D a partir de un volumen de imágenes en formato PGM, permitiendo extraer información relevante en distintos planos (X, Y y Z) con diferentes criterios (máximo, mínimo, promedio y mediana).

¿Qué se propone?

Se propone un sistema basado en C++ que permita la carga, manipulación y procesamiento de imágenes en formato PGM, incluyendo la funcionalidad de generar proyecciones 2D aplicando distintos criterios.

¿Cómo se soluciona?

El sistema implementa estructuras de datos eficientes, como `std::vector`, para manejar las imágenes y sus volúmenes. Se desarrollaron funciones para cargar imágenes, gestionar volúmenes y calcular proyecciones en los ejes X, Y y Z, aplicando los criterios mencionados.

¿Qué resultados surgen?

Los resultados muestran que cada criterio de proyección resalta diferentes características del volumen de imágenes, permitiendo visualizar de manera más efectiva la información contenida en ellas. Se identificaron mejoras potenciales, como la optimización del manejo de imágenes de diferentes dimensiones.

2. Introducción

Este documento describe el diseño del **Sistema de Procesamiento de Imágenes**, detallando los **Tipos Abstractos de Datos (TADs)** utilizados, las operaciones implementadas y su estructura. También se incluye un **plan de pruebas** para validar el funcionamiento del comando `proyeccion2D`.

3. Desarrollo

3.1 Descripción del Sistema

El sistema implementado permite la manipulación y procesamiento de imágenes en formato **PGM (P2)**, con funcionalidades como:

- Carga de imágenes individuales (`cargar_imagen`).
- Carga de volúmenes de imágenes (`cargar_volumen`).
- Consulta de información (`info_imagen` y `info_volumen`).
- Generación de proyecciones en 2D (`proyeccion2D`).

El código está desarrollado en **C++**, utilizando estructuras de datos como **vectores** (`std::vector`) para almacenar las imágenes y el volumen de datos.

3.2 Tipos Abstractos de Datos (TADs)

TAD: Imagen

Descripción: Representa una imagen en formato **PGM** cargada en memoria.

Datos:

- `imageData`: Matriz de enteros que almacena los valores de los píxeles.
- `width`: Ancho de la imagen.
- `height`: Alto de la imagen.
- `maxPixelValue`: Valor máximo de píxel permitido.
- `imageFilename`: Nombre del archivo cargado.

Operaciones:

- `loadImage(nombre_archivo)`: Carga una imagen en memoria.
- `infoImage()`: Muestra información de la imagen cargada.

TAD: Volumen de Imágenes

Descripción: Representa un conjunto de imágenes de la misma dimensión cargadas como un volumen tridimensional.

Datos:

- `volumeData`: Vector 3D que almacena todas las imágenes del volumen.
- `volume`: Nombre base del volumen cargado.

- `width, height`: Dimensiones de las imágenes.
- `maxPixelValue`: Valor máximo de píxel.

Operaciones:

- `loadVolume(base_nombre, n_im)`: Carga un volumen de `n_im` imágenes.
- `infoVolume()`: Muestra información del volumen.
- `proyeccion2D(direccion, criterio, archivo_salida)`: Genera una proyección en 2D del volumen.

4. Diseño del Sistema y Funcionalidades

4.1. Estructura General

El sistema sigue un modelo **basado en comandos**, donde el usuario puede interactuar mediante la terminal.

4.2. Entradas y Salidas

Función	Entrada	Salida
<code>cargar_imagen</code>	Nombre de un archivo <code>.pgm</code>	Imagen cargada en memoria.
<code>cargar_volumen</code>	Base del nombre y número de imágenes	Volumen de imágenes cargado.
<code>info_imagen</code>	Ninguna	Información de la imagen cargada.
<code>info_volumen</code>	Ninguna	Información del volumen cargado.
<code>proyeccion2D</code>	Dirección (x, y, z), criterio, salida	Archivo PGM con la proyección 2D.

5. Plan de Pruebas para `proyeccion2D`

5.1. Casos de Prueba

ID	Descripción	Entrada	Salida Esperada
P1	Proyección en el eje Z (máximo)	<code>proyeccion2D z max salida.pgm</code>	Imagen con valores máximos en Z.

ID	Descripción	Entrada	Salida Esperada
P2	Proyección en el eje Z (mínimo)	proyeccion2D z min salida.pgm	Imagen con valores mínimos en Z.
P3	Proyección en el eje Z (promedio)	proyeccion2D z prom salida.pgm	Imagen con valores promedio en Z.
P4	Proyección en el eje Z (mediana)	proyeccion2D z med salida.pgm	Imagen con la mediana de valores en Z.
P5	Proyección en el eje Y (máximo)	proyeccion2D y max salida.pgm	Imagen con valores máximos en Y.
P6	Proyección en el eje Y (mínimo)	proyeccion2D y min salida.pgm	Imagen con valores mínimos en Y.
P7	Proyección en el eje Y (promedio)	proyeccion2D y prom salida.pgm	Imagen con valores promedio en Y.
P8	Proyección en el eje Y (mediana)	proyeccion2D y med salida.pgm	Imagen con la mediana de valores en Y.
P9	Proyección en el eje X (máximo)	proyeccion2D x max salida.pgm	Imagen con valores máximos en X.
P10	Proyección en el eje X (mínimo)	proyeccion2D x min salida.pgm	Imagen con valores mínimos en X.
P11	Proyección en el eje X (promedio)	proyeccion2D x prom salida.pgm	Imagen con valores promedio en X.
P12	Proyección en el eje X (mediana)	proyeccion2D x med salida.pgm	Imagen con la mediana de valores en X.
P13	Proyección sin volumen cargado	proyeccion2D x max salida.pgm	Mensaje de error: <i>No hay volumen cargado.</i>

5.2. Criterios de Aceptación

- La proyección debe generarse correctamente según el criterio seleccionado.
 - El tamaño de la imagen proyectada debe coincidir con la resolución esperada.
 - Si no hay volumen cargado, el sistema debe mostrar un mensaje de error.
-

Guía para Clonar, Compilar y Ejecutar el Programa desde GitHub

1. Clonar el Repositorio

Para obtener el código fuente del programa, debe clonar el repositorio desde GitHub.

Instrucciones:

1. **Abrir la terminal** (Linux/macOS) o **Git Bash** (Windows).
2. **Ejecutar el siguiente comando:**

```
git clone https://github.com/jeisonAlfonso/entrega1-ED.git
```

3. **Entrar a la carpeta del repositorio clonado:**

```
cd entrega1-ED
```

2. Compilar el Programa con Makefile

Una vez dentro de la carpeta del repositorio, debe compilar el código utilizando `make`.

Ejecutar el siguiente comando:

```
make
```

Esto generará un **ejecutable** llamado `programa`.

3. Ejecutar el Programa

Después de compilar, puede ejecutar el programa con:

En Linux/macOS:

```
make run
```

En Windows (con MinGW):

```
make run
```

Esto iniciará el sistema de procesamiento de imágenes en la terminal.

4. Limpiar Archivos Compilados (Opcional)

Si desea eliminar los archivos compilados (programa y *.o), puede ejecutar:

```
make clean
```

Esto **no borra** el código fuente ni los archivos .pgm, solo los archivos generados durante la compilación.

5. Uso de los Comandos en el Programa

Una vez ejecutado, el sistema mostrará un prompt (\$) donde podrá ingresar comandos.

Ejemplo de comandos en el programa:

```
cargar_imagen imagen1.pgm
info_imagen
proyeccion2D z max salida.pgm
salir
```

Estos comandos permiten cargar imágenes, obtener información y generar proyecciones 2D.

7. Resultados de Pruebas

Se realizaron pruebas utilizando cuatro imágenes en formato PGM, aplicando la función de proyección en 2D sobre el volumen de imágenes según los criterios de máximo, mínimo, promedio y mediana en los ejes X, Y y Z.

Las imágenes generadas muestran las siguientes características:

- En la proyección con criterio de máximo, las regiones más brillantes se conservan, lo que resalta las áreas de mayor intensidad en el volumen.
- En la proyección con criterio de mínimo, se observan las zonas más oscuras de cada capa del volumen, destacando las regiones con menor intensidad.
- las imágenes convertidas a formato png, permiten ver la proyección2D demostrando los valores máximos y mínimos de intensidad. Agregando que todas se estandarizaron a un mismo tamaño



Imagen 1. Primer archivo. pgm ilustrado en formato png



Imagen 2. segundo archivo. pgm ilustrado en formato png



Imagen 3. tercer archivo. pgm ilustrado en formato png

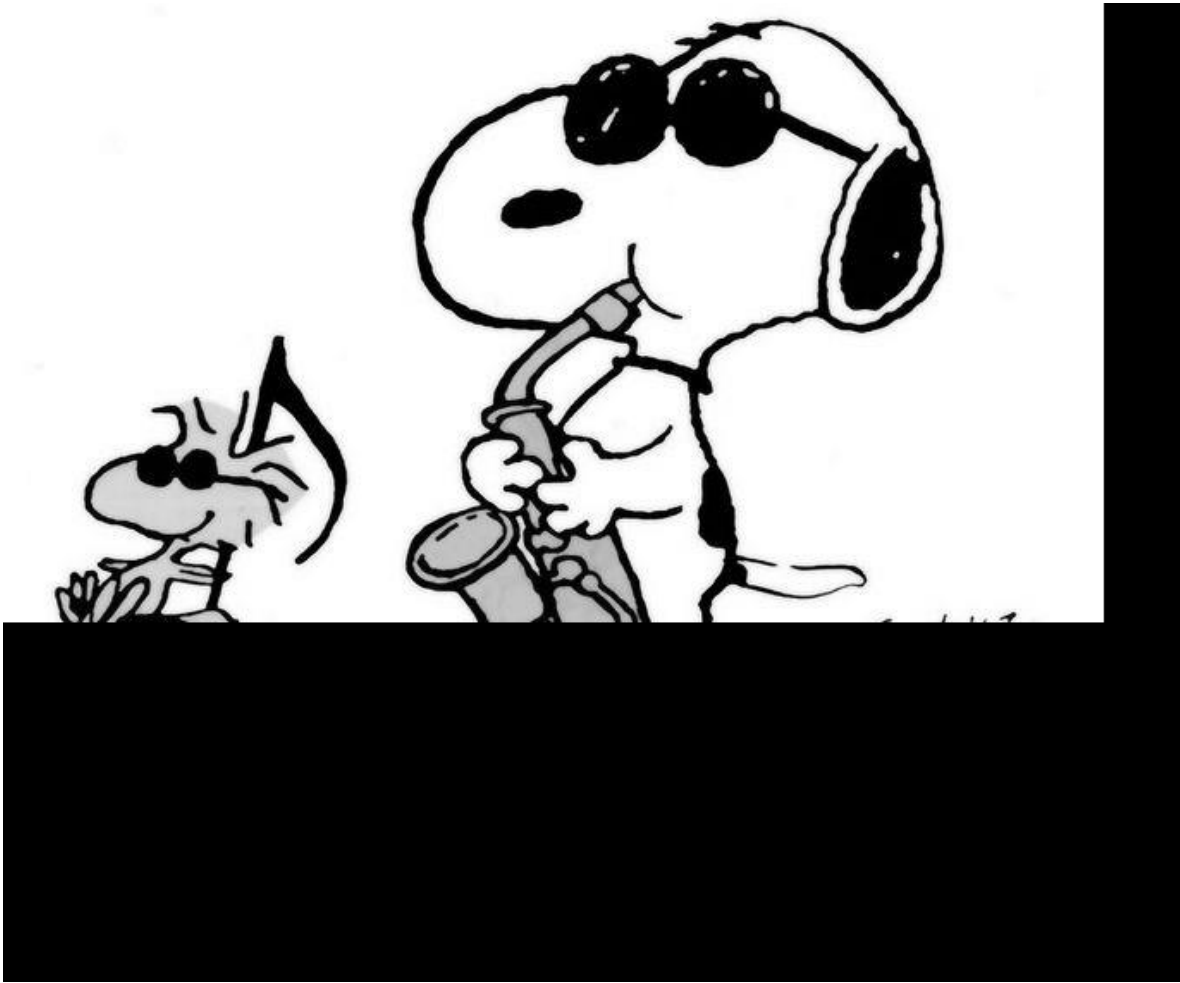


Imagen 4. Primer archivo. pgm ilustrado en formato png



Imagen 5. Proyección 2D con el criterio máximo en el eje z.

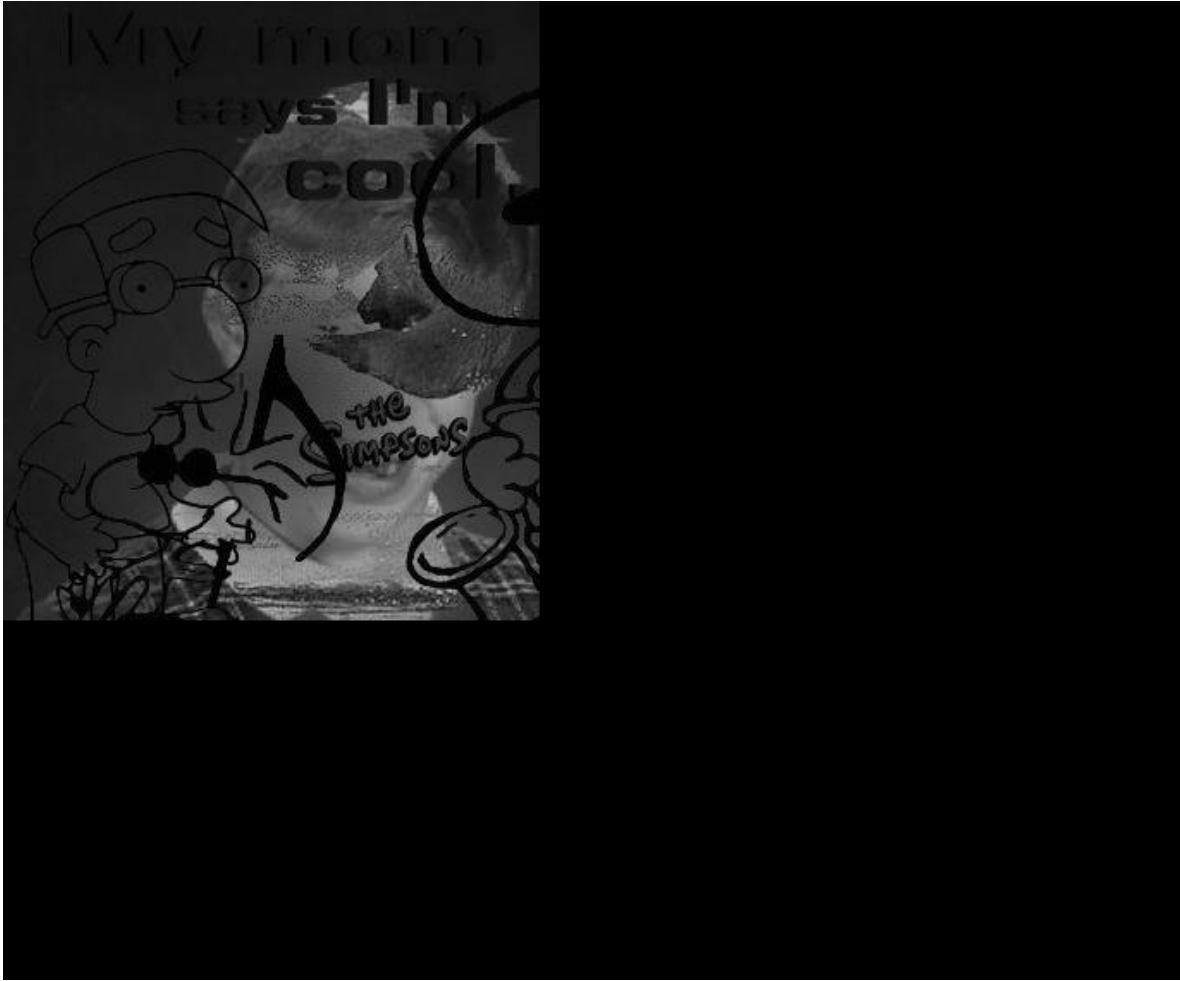


Imagen 5. Proyección 2D con el criterio mínimo en el eje z

8. Conclusiones

Los resultados de las proyecciones 2D permiten analizar la información del volumen de imágenes desde diferentes perspectivas. Se evidenció que cada criterio de proyección ofrece una visualización distinta de los datos, permitiendo resaltar diferentes características de las imágenes originales.

El sistema demostró ser funcional para la generación de proyecciones en 2D a partir de volúmenes de imágenes en formato PGM. No obstante, la precisión de los resultados depende de la calidad y homogeneidad de las imágenes de entrada.

Para futuras mejoras, se podría implementar un mecanismo para ajustar la escala de los valores de salida y mejorar la interpolación en volúmenes con imágenes de diferentes resoluciones.

Para este caso los ejemplos mas ilustrativos son los del criterio máximo y mínimo y por eso se hizo la conversión a formato png para evidenciar que el programa hace su trabajo correctamente con las imágenes redimensionadas y demás características, sin embargo, se puede probar con cualquier método como los de mediana o promedio y de igual forma hará el trabajo correctamente de la proyección.

Esto se debe hacer para validar la proyección en todos los ejes y con los todos los métodos, sin embargo, no se adjunta la imagen ya que como son solo 4 que concuerdan con el formato pgm, al momento de transformarlas a png, solo muestra 4 líneas con los colores del criterio correspondiente en los ejes que no se ilustran, que son el x y el y. para esta validación, hay que ver el archivo generado por la proyección, y validar los números de intensidades para que se vea su correcto funcionamiento.

Cabe destacar, que también se debe intentar usar el archivo t1_icbm_5mm_03 y t1_icbm_5mm_04 siguiendo las instrucciones del programa, para validar los métodos de cargar_imagen, cargar_volumen y proyeccion2D, sin embargo, para cumplir con los requisitos del entregable, tanto para hacer un volumen como para la proyección, estos archivos no cumplen con el requisito del formato (.pgm P2), ni con la información requerida para el formato, por lo tanto en la pantalla se mostraran los diferentes errores los diferentes errores.

9. Bibliografía

- Netpbm - PGM Format Specification. Disponible en: <https://netpbm.sourceforge.net/doc/pgm.html>
- OpenCV Documentation: Image Processing. Disponible en: <https://docs.opencv.org/>
- Stack Overflow: Implementación de operaciones con imágenes PGM en C++. Disponible en: <https://stackoverflow.com/>