TDA Imagen

Generado por Doxygen 1.8.17

Chapter 1

Índice de clases

1.1 Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

Chapter 2

Indice de archivos

2.1 Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

Chapter 3

Documentación de las clases

3.1 Referencia de la Clase Imagen

clase que representa una imagen digital en escala de grises

`#include <imagenES.h>`

Métodos públicos

- Imagen (int filas, int columnas)

  constructor de la clase Imagen.
- Imagen (const Imagen &otra)

  constructor de copia de la clase Imagen
- ~Imagen ()

  destructor de la clase Imagen.
- Imagen & operator= (const Imagen &otra)

  operador de asignación entre imagenes.
- int num_filas () const

  Calcula el número de filas de la imagen.
- int num_columnas () const

  Calcula el número de columnas de la imagen.
- void asigna_pixel (int fila, int columna, byte valor)

  Asigna el valor valor al pixel indicado por fila y columna de la imagen.
- byte valor_pixel (int fila, int columna) const

  Consulta el valor de un pixel de la imagen.

3.1.1 Descripción detallada

clase que representa una imagen digital en escala de grises

3.1.2 operaciones

Se definen una serie de operaciones:

1. Creación de una imagen
2. Destrucción de una imagen
3. Consultar el número de filas
4. Consultar el número de columnas
5. Asignar un valor a un punto de la imagen
6. Consultar el valor de un punto de la imagen

3.1.3 Documentación del constructor y destructor

3.1.3.1 Imagen() [1/2]

```
Imagen::Imagen (
            int filas,
            int columnas )
```

constructor de la clase Imagen.

Parámetros

filas	el número de filas que tendrá la imagen
columnas	el número de columnas que tendrá la imagen

Devuelve

el objeto nuevo de imagen esta inicializado a una imagen en negro

MÉTODOS DE LA CLASE IMAGEN

3.1.3.2 Imagen() [2/2]

```
Imagen::Imagen (
            const Imagen & otra )
```

constructor de copia de la clase Imagen

Parámetros

otra	la imagen que copiamos

Devuelve

el nuevo objeto, copia de *otra*

3.1.3.3 ~Imagen()

```
Imagen::~Imagen ( )
```

destructor de la clase Imagen.

Postcondición

destruye la imagen, libera la memoria, volverá a usarse con una llamada al constructor

3.1.4 Documentación de las funciones miembro

3.1.4.1 asigna_pixel()

```
void Imagen::asigna_pixel (
            int fila,
            int columna,
            byte valor )
```

Asigna el valor *valor* al pixel indicado por *fila* y *columna* de la imagen.

Parámetros

fila	la fila del pixel a modificar
columna	la columna del pixel a modificar
valor	el valor que se asigna a la posicion (*fila*, *columna*)

Precondición

fila y *columna* deben ser valores válidos [0, num_filas()] para *fila* y [0,num_columnas()] para *columnas valor* debe ser [0,255]

Postcondición

Imagen(*fila*, *columna*) = *valor* . El resto de pixeles no se modifica

3.1.4.2 num_columnas()

```
int Imagen::num_columnas ( ) const
```

Calcula el número de columnas de la imagen.

Devuelve

el número de columnas que tienen la imagen

Postcondición

no se modifica la imagen

3.1.4.3 num_filas()

```
int Imagen::num_filas ( ) const
```

Calcula el número de filas de la imagen.

Devuelve

el número de filas de la imagen

Postcondición

la imagen no se modifica

3.1.4.4 operator=()

```
Imagen & Imagen::operator= (
            const Imagen & otra )
```

operador de asignación entre imagenes.

Parámetros

otra	la imagen que asignamos a la que llama al operador

Devuelve

una referencia al objeto que llama el operador

Postcondición

el objeto es una copia del pasado por referencia

3.1.4.5 valor_pixel()

```
byte Imagen::valor_pixel (
             int fila,
             int columna ) const
```

Consulta el valor de un pixel de la imagen.

Parámetros

fila	la fila del pixel a consultar
columna	la columna del pixel a consultar

Precondición

fila y *columna* deben ser valores válidos. [0, num_filas()] para *fila* y [0,num_columnas()] para *columnas*.

Postcondición

no se modifica la imagen

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- inc/imagenES.h
- src/imagenES.cpp

Chapter 4

Documentación de archivos

4.1 Referencia del Archivo inc/imagenES.h

Fichero cabecera para la E/S de imágenes.

Gráfico de los archivos que directa o indirectamente incluyen a este archivo:

4.2 Referencia del Archivo src/imagenES.cpp

Fichero con definiciones para la E/S de imágenes.

```
#include <iostream>
#include <fstream>
#include <string>
#include <cassert>
#include <imagenES.h>
#include <cmath>
```

Dependencia gráfica adjunta para imagenES.cpp:

Funciones

- TipoImagen **LeerTipo** (ifstream &f)
- TipoImagen LeerTipoImagen (const char *nombre)

  Devuelve el tipo de imagen del archivo.
- char **SaltarSeparadores** (ifstream &f)
- bool **LeerCabecera** (ifstream &f, int &fils, int &cols)
- unsigned char * LeerImagenPPM (const char *nombre, int &fils, int &cols)

  Lee una imagen de tipo PPM.
- unsigned char * LeerImagenPGM (const char *nombre, int &fils, int &cols)

  Lee una imagen de tipo PGM.
- bool EscribirImagenPPM (const char *nombre, const unsigned char *datos, const int fils, const int cols)

  Escribe una imagen de tipo PPM.
- bool EscribirImagenPGM (const char *nombre, const unsigned char *datos, const int fils, const int cols)

  Escribe una imagen de tipo PGM.
- void error (string mensaje)

- Imagen leerVectorPGM (byte *vector, int filas, int columnas)

  funcion para obtener una imagen a partir de un vector 1d de una imagen PGM
- Imagen leerVectorPPM (byte *vector, int filas, int columnas)

  funcion para obtener una imagen a partir de un vector 1d de una imagen PPM
- void escribirVectorPGM (const Imagen &img, byte *vector, int &filas, int &columnas)

  funcion para transformar un objeto imagen en un vector unidimensional
- void colorAGris (const char *nombre_ppm, const char *nombre_pgm)

  función que permite convertir una imagen PPM a una imagen PGM

Variables

- const double **ROJO_GRIS** = 0.2989
- const double **VERDE_GRIS** = 0.587
- const double **AZUL_GRIS** = 0.114

4.2.1 Descripción detallada

Fichero con definiciones para la E/S de imágenes.

Autor

Salvador Romero Cortés Permite la E/S de archivos de tipo PGM,PPM

4.2.2 Documentación de las funciones

4.2.2.1 colorAGris()

```
void colorAGris (
            const char * nombre_ppm,
            const char * nombre_pgm )
```

función que permite convertir una imagen PPM a una imagen PGM

Parámetros

nombre_ppm	la imagen a color
nombre_pgm	la imagen en escala de grises

Precondición

debe existir el archivo con el nombre *nombre_ppm*

Postcondición

se escribe directamente el archivo desde esta funcion

4.2.2.2 error()

```
void error (
            string mensaje )
```

FUNCIONES AUXILIARES

4.2.2.3 EscribirImagenPGM()

```
bool EscribirImagenPGM (
            const char * nombre,
            const unsigned char * datos,
            const int filas,
            const int columnas )
```

Escribe una imagen de tipo PGM.

Parámetros

nombre	archivo a escribir
datos	punteros a los *f* x *c* bytes que corresponden a los valores de los píxeles de la imagen de grises.
filas	filas de la imagen
columnas	columnas de la imagen

Devuelve

si ha tenido éxito en la escritura.

4.2.2.4 EscribirImagenPPM()

```
bool EscribirImagenPPM (
            const char * nombre,
            const unsigned char * datos,
            const int filas,
            const int columnas )
```

Escribe una imagen de tipo PPM.

Parámetros

nombre	archivo a escribir
datos	punteros a los *f* x *c* x 3 bytes que corresponden a los valores de los píxeles de la imagen en formato RGB.
filas	filas de la imagen
columnas	columnas de la imagen

Devuelve

si ha tenido éxito en la escritura.

4.2.2.5 escribirVectorPGM()

```
void escribirVectorPGM (
            const Imagen & img,
            byte * vector,
            int & filas,
            int & columnas )
```

funcion para transformar un objeto imagen en un vector unidimensional

Parámetros

img	la imagen que se va a convertir
vector	el vector donde se escribe la imagen
filas	el número de filas de la imagen
columnas	el número de columnas de la imagen

Precondición

Postcondición

todos los parámetros se modifican menos *img*

4.2.2.6 LeerImagenPGM()

```
unsigned char* LeerImagenPGM (
            const char * nombre,
            int & filas,
            int & columnas )
```

Lee una imagen de tipo PGM.

Parámetros

nombre	archivo a leer
filas	Parámetro de salida con las filas de la imagen.
columnas	Parámetro de salida con las columnas de la imagen.

Devuelve

puntero a una nueva zona de memoria que contiene *filas* x *columnas* bytes que corresponden a los grises de todos los píxeles (desde la esquina superior izqda a la inferior drcha). En caso de que no no se pueda leer, se devuelve cero. (0).

Postcondición

En caso de éxito, el puntero apunta a una zona de memoria reservada en memoria dinámica. Será el usuario el responsable de liberarla.

4.2.2.7 LeerImagenPPM()

```
unsigned char* LeerImagenPPM (
            const char * nombre,
            int & filas,
            int & columnas )
```

Lee una imagen de tipo PPM.

Parámetros

nombre	archivo a leer
filas	Parámetro de salida con las filas de la imagen.
columnas	Parámetro de salida con las columnas de la imagen.

Devuelve

puntero a una nueva zona de memoria que contiene *filas* x *columnas* x 3 bytes que corresponden a los colores de todos los píxeles en formato RGB (desde la esquina superior izqda a la inferior drcha). En caso de que no no se pueda leer, se devuelve cero. (0).

Postcondición

En caso de éxito, el puntero apunta a una zona de memoria reservada en memoria dinámica. Será el usuario el responsable de liberarla.

4.2.2.8 LeerTipoImagen()

```
TipoImagen LeerTipoImagen (
            const char * nombre )
```

Devuelve el tipo de imagen del archivo.

Parámetros

nombre	indica el archivo de disco que consultar

Devuelve

Devuelve el tipo de la imagen en el archivo

Ver también

TipoImagen

4.2.2.9 leerVectorPGM()

```
Imagen leerVectorPGM (
            byte * vector,
            int filas,
            int columnas )
```

funcion para obtener una imagen a partir de un vector 1d de una imagen PGM

Parámetros

vector	el vector unidimensional que contiene los pixeles de la imagen
filas	el número de filas que tendrá la imagen
columnas	el número de columnas que tendrá la imagen

Devuelve

objeto imagen con los datos del vector

4.2.2.10 leerVectorPPM()

```
Imagen leerVectorPPM (
            byte * vector,
            int filas,
            int columnas )
```

funcion para obtener una imagen a partir de un vector 1d de una imagen PPM

Parámetros

vector	el vector unidimensional que contiene los pixeles de la imagen
filas	el número de filas que tendrá la imagen
columnas	el número de columnas que tendrá la imagen

Devuelve

objeto imagen con los datos del vector (en escala de grises)

4.3 Referencia del Archivo src/main.cpp

archivo que ejecuta el programa principal

```
#include <iostream>
#include <imagenES.h>
```

Dependencia gráfica adjunta para main.cpp:

Funciones

- int main (int argc, char ∗args[ ])

  funcion principal del programa. Se encarga de la ejecución del programa

4.3.1 Descripción detallada

archivo que ejecuta el programa principal

Autor

Salvador Romero Cortés

4.3.2 Documentación de las funciones

4.3.2.1 main()

```
int main (
            int argc,
            char * args[] )
```

funcion principal del programa. Se encarga de la ejecución del programa

Parámetros

argc	el número de parametros pasados en la ejecucion
args	el vector de los parametros pasados en la ejecucion