

ClaseImagenyFunciones

1.0

Generado por Doxygen 1.5.6

Tue Jun 23 12:09:50 2009

Índice general

1. Índice de clases	1
1.1. Lista de clases	1
2. Índice de archivos	3
2.1. Lista de archivos	3
3. Documentación de las clases	5
3.1. Referencia de la Clase Imagen	5
3.1.1. Descripción detallada	6
3.1.2. Documentación del constructor y destructor	6
3.1.2.1. Imagen	6
3.1.2.2. Imagen	6
3.1.2.3. Imagen	6
3.1.2.4. ~Imagen	7
3.1.3. Documentación de las funciones miembro	7
3.1.3.1. operator=	7
3.1.3.2. Reserva	7
3.1.3.3. guardarPGM	7
3.1.3.4. cargarPGM	8
3.1.3.5. num_filas	8
3.1.3.6. num_columnas	8
3.1.3.7. asigna_pixel	8
3.1.3.8. valor_pixel	9
4. Documentación de archivos	11
4.1. Referencia del Archivo include/funciones.h	11
4.1.1. Descripción detallada	11
4.1.2. Documentación de las funciones	12
4.1.2.1. crearIcono	12

4.1.2.2.	ecualizar	12
4.1.2.3.	mejorarContraste	13
4.1.2.4.	morphing	13
4.1.2.5.	RGB2Gris	13
4.1.2.6.	umbralizar	14
4.2.	Referencia del Archivo include/Imagen.h	15
4.2.1.	Descripción detallada	15
4.2.2.	Documentación de los tipos definidos	15
4.2.2.1.	byte	15
4.3.	Referencia del Archivo include/imagenES.h	16
4.3.1.	Descripción detallada	16
4.3.2.	Documentación de las enumeraciones	16
4.3.2.1.	TipoImagen	16
4.3.3.	Documentación de las funciones	17
4.3.3.1.	EscribirImagenPGM	17
4.3.3.2.	EscribirImagenPPM	17
4.3.3.3.	LeerImagenPGM	18
4.3.3.4.	LeerImagenPPM	18
4.3.3.5.	LeerTipoImagen	19
4.4.	Referencia del Archivo src/contraste.cpp	20
4.4.1.	Descripción detallada	20
4.4.2.	Documentación de las funciones	20
4.4.2.1.	main	20
4.5.	Referencia del Archivo src/crearicono.cpp	21
4.5.1.	Descripción detallada	21
4.5.2.	Documentación de las funciones	21
4.5.2.1.	main	21
4.6.	Referencia del Archivo src/ecualizar.cpp	22
4.6.1.	Descripción detallada	22
4.6.2.	Documentación de las funciones	22
4.6.2.1.	main	22
4.7.	Referencia del Archivo src/funciones.cpp	23
4.7.1.	Descripción detallada	23
4.7.2.	Documentación de las funciones	24
4.7.2.1.	calcularUmbral	24
4.7.2.2.	crearIcono	24

4.7.2.3.	ecualizar	24
4.7.2.4.	mejorarContraste	25
4.7.2.5.	morphing	25
4.7.2.6.	RGB2Gris	25
4.7.2.7.	umbralizar	26
4.8.	Referencia del Archivo src/Imagen.cpp	27
4.8.1.	Descripción detallada	27
4.9.	Referencia del Archivo src/imagenES.cpp	28
4.9.1.	Descripción detallada	28
4.9.2.	Documentación de las funciones	28
4.9.2.1.	EscribirImagenPGM	28
4.9.2.2.	EscribirImagenPPM	29
4.9.2.3.	LeerCabecera	29
4.9.2.4.	LeerImagenPGM	29
4.9.2.5.	LeerImagenPPM	30
4.9.2.6.	LeerTipo	30
4.9.2.7.	LeerTipoImagen	30
4.9.2.8.	SaltarSeparadores	31
4.10.	Referencia del Archivo src/morphing.cpp	32
4.10.1.	Descripción detallada	32
4.10.2.	Documentación de las funciones	32
4.10.2.1.	main	32
4.11.	Referencia del Archivo src/negativo.cpp	33
4.11.1.	Descripción detallada	33
4.11.2.	Documentación de las funciones	33
4.11.2.1.	main	33
4.12.	Referencia del Archivo src/rgb2gris.cpp	34
4.12.1.	Descripción detallada	34
4.12.2.	Documentación de las funciones	34
4.12.2.1.	main	34
4.13.	Referencia del Archivo src/umbralizar.cpp	35
4.13.1.	Descripción detallada	35
4.13.2.	Documentación de las funciones	35
4.13.2.1.	main	35

Capítulo 1

Índice de clases

1.1. Lista de clases

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

[Imagen](#) (Clase que almacena la información de una imagen y se encarga de su gestión) 5

Capítulo 2

Indice de archivos

2.1. Lista de archivos

Lista de todos los archivos con descripciones breves:

include/funciones.h (Fichero cabecera de funciones extra para la modificación de imagenes) . . .	11
include/Imagen.h (Fichero de cabecera asociado a la biblioteca libImagen.a)	15
include/imagenES.h (Fichero cabecera para la E/S de imágenes)	16
src/contraste.cpp (Fichero de aplicación de la función de mejora de contraste)	20
src/crearicono.cpp (Fichero de aplicación de la función de reducción de imagen)	21
src/ecualizar.cpp (Fichero de aplicación de la función de ecualización automática)	22
src/funciones.cpp (Fichero con definiciones de funciones extra para la modificación de imagenes)	23
src/Imagen.cpp (Fichero con las definiciones de la clase Imagen asociada a la biblioteca libIma- gen.a)	27
src/imagenES.cpp (Fichero con definiciones para la E/S de imágenes)	28
src/morphing.cpp (Fichero de aplicación de la función morphing)	32
src/negativo.cpp (Calcula el negativo de una imagen PGM)	33
src/rgb2gris.cpp (Fichero de aplicación de la función conversión a escala de grises)	34
src/umbralizar.cpp (Fichero de aplicación de la función de umbralización automática)	35

Capítulo 3

Documentación de las clases

3.1. Referencia de la Clase Imagen

Clase que almacena la información de una imagen y se encarga de su gestión.

```
#include <Imagen.h>
```

Métodos públicos

- `Imagen ()`
Constructor por defecto.
- `Imagen (const Imagen &imagen)`
Constructor de copias.
- `Imagen (int fils, int columnas)`
Crear una imagen en memoria con fils filas y columnas columnas.
- `~Imagen ()`
Destructor.
- `Imagen & operator= (const Imagen &orig)`
Operador de asignación.
- `void Reserva (int fils, int columnas)`
Liberar los recursos ocupados por la imagen y crear una imagen en memoria con fils filas y columnas columnas.
- `bool guardarPGM (const char *salida) const`
Almacena la imagen en un fichero PGM.
- `bool cargarPGM (const char *entrada)`
Carga la imagen de un fichero PGM.
- `int num_filas () const`
Calcular el número de filas de la imagen.

- `int num_columnas () const`
Calcular el número de columnas de la imagen.
- `void asigna_pixel (int fil, int col, byte valor)`
Asignar el valor valor al píxel (fil, col) de la imagen.
- `byte valor_pixel (int fil, int col) const`
Consultar el valor de la casilla (fil, col) de la imagen.

3.1.1. Descripción detallada

Clase que almacena la información de una imagen y se encarga de su gestión.

Definición en la línea 19 del archivo Imagen.h.

3.1.2. Documentación del constructor y destructor

3.1.2.1. `Imagen::Imagen ()`

Constructor por defecto.

Definición en la línea 15 del archivo Imagen.cpp.

Referenciado por `Imagen()`.

3.1.2.2. `Imagen::Imagen (const Imagen & imagen)`

Constructor de copias.

Parámetros:

imagen `Imagen` a copiar

Definición en la línea 26 del archivo Imagen.cpp.

Hace referencia a `cols`, `filas`, `Imagen()`, y `valor_pixel()`.

3.1.2.3. `Imagen::Imagen (int fils, int columnas)`

Crear una imagen en memoria con `fils` filas y `columnas` columnas.

Parámetros:

fils filas a reservar

columnas columnas a reservar

Postcondición:

La imagen creada contiene `fils` filas y `columnas` columnas

Definición en la línea 40 del archivo Imagen.cpp.

3.1.2.4. Imagen::~~Imagen ()

Destructor.

Definición en la línea 50 del archivo Imagen.cpp.

3.1.3. Documentación de las funciones miembro

3.1.3.1. Imagen & Imagen::operator= (const Imagen & orig)

Operador de asignación.

Parámetros:

orig Imagen a copiar

Definición en la línea 79 del archivo Imagen.cpp.

Hace referencia a cols, filas, Reserva(), y valor_pixel().

3.1.3.2. void Imagen::Reserva (int *filas*, int *columnas*)

Liberar los recursos ocupados por la imagen y crear una imagen en memoria con *filas* filas y *columnas* columnas.

Parámetros:

filas filas a reservar

columnas columnas a reservar

Definición en la línea 62 del archivo Imagen.cpp.

Referenciado por calcularUmbral(), cargarPGM(), crearIcono(), operator=(), y RGB2Gris().

3.1.3.3. bool Imagen::guardarPGM (const char * *salida*) const

Almacena la imagen en un fichero PGM.

Parámetros:

salida dirección del archivo a crear

Devuelve:

Si ha tenido éxito en la escritura

Definición en la línea 94 del archivo Imagen.cpp.

Hace referencia a EscribirImagenPGM().

Referenciado por crearIcono(), ecualizar(), mejorarContraste(), morphing(), RGB2Gris(), y umbralizar().

3.1.3.4. bool Imagen::cargarPGM (const char * *entrada*)

Carga la imagen de un fichero PGM.

Parámetros:

salida dirección del archivo a crear

Devuelve:

Si ha tenido éxito en la carga

Definición en la línea 114 del archivo Imagen.cpp.

Hace referencia a asigna_pixel(), IMG_PGM, LeerImagenPGM(), LeerTipoImagen(), y Reserva().

Referenciado por crearIcono(), ecualizar(), mejorarContraste(), morphing(), y umbralizar().

3.1.3.5. int Imagen::num_filas () const

Calcular el número de filas de la imagen.

Devuelve:

Número de filas de la imagen

Definición en la línea 138 del archivo Imagen.cpp.

Referenciado por calcularUmbral(), crearIcono(), ecualizar(), mejorarContraste(), morphing(), y umbralizar().

3.1.3.6. int Imagen::num_columnas () const

Calcular el número de columnas de la imagen.

Devuelve:

Número de columnas de la imagen

Definición en la línea 145 del archivo Imagen.cpp.

Referenciado por calcularUmbral(), crearIcono(), ecualizar(), mejorarContraste(), morphing(), y umbralizar().

3.1.3.7. void Imagen::asigna_pixel (int *fil*, int *col*, byte *valor*)

Asignar el valor valor al píxel (fil, col) de la imagen.

Parámetros:

fil fila a modificar

col columna a modificar

valor valor a almacenar en (fil, col)

Postcondición:

$I(fil,col) == valor$

Definición en la línea 155 del archivo Imagen.cpp.

Referenciado por calcularUmbral(), cargarPGM(), crearIcono(), ecualizar(), mejorarContraste(), morphing(), RGB2Gris(), y umbralizar().

3.1.3.8. byte Imagen::valor_pixel (int *fil*, int *col*) const

Consultar el valor de la casilla (*fil*, *col*) de la imagen.

Parámetros:

fil fila a consultar

col columna a consultar

Devuelve:

Valor en (*fil*, *col*)

Definición en la línea 167 del archivo Imagen.cpp.

Referenciado por calcularUmbral(), crearIcono(), ecualizar(), Imagen(), mejorarContraste(), morphing(), operator=(), y umbralizar().

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

- [include/Imagen.h](#)
- [src/Imagen.cpp](#)

Capítulo 4

Documentación de archivos

4.1. Referencia del Archivo include/funciones.h

Fichero cabecera de funciones extra para la modificación de imagenes.

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

Funciones

- void [RGB2Gris](#) (const char *fich_E, const char *fich_S)
Función que convierte una imagen RGB (PPM) a escala de grises (PGM).
- void [mejorarContraste](#) (const char *fich_E, const char *fich_S, float gamma)
Función que cambia el contraste de la imagen en función del parámetro gamma.
- void [ecualizar](#) (const char *fich_E, const char *fich_S)
Función que realiza un ecualizado automático de la imagen.
- void [umbralizar](#) (const char *fich_E, const char *fich_S, int &T)
Función que realiza un umbralizado automático de la imagen.
- void [crearIcono](#) (const char *fich_E, const char *fich_S, int factor)
Función que reduce el tamaño de una imagen en un factor "factor".
- void [morphing](#) (const char *fich_E, const char *fich_S, const char *prefijo)
Función que genera una lista de imagenes interpoladas entre dos de referencia.

4.1.1. Descripción detallada

Fichero cabecera de funciones extra para la modificación de imagenes.

Autor:

Miguel Cantón Cortés

Estas funciones sirven de ejemplo de la clase [Imagen](#)

Definición en el archivo [funciones.h](#).

4.1.2. Documentación de las funciones

4.1.2.1. void crearIcono (const char *fich_E, const char *fich_S, int factor)

Función que reduce el tamaño de una imagen en un factor "factor".

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

factor factor de reducción de la imagen

Agrupar la imagen en bloques de factor*factor píxeles y calcula su media. Esta media se corresponde con cada píxel de la imagen reducida.

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

factor factor de reducción de la imagen

Definición en la línea 187 del archivo funciones.cpp.

Hace referencia a `Imagen::asigna_pixel()`, `Imagen::cargarPGM()`, `Imagen::guardarPGM()`, `Imagen::num_columns()`, `Imagen::num_filas()`, `Imagen::Reserva()`, y `Imagen::valor_pixel()`.

Referenciado por `main()`.

4.1.2.2. void ecualizar (const char *fich_E, const char *fich_S)

Función que realiza un ecualizado automático de la imagen.

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

Calculamos el numero de apariciones de cada nivel de gris y lo almacenamos en el vector histograma. Calculamos las probabilidades de cada nivel de gris, dividiendo su número de apariciones entre el total de píxeles, y lo almacenamos en el vector probabilidades. Finalmente usando el vector de probabilidades, y aplicando una fórmula, obtenemos un nuevo vector con los nuevos valores de grises, transformacion

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

Definición en la línea 72 del archivo funciones.cpp.

Hace referencia a `Imagen::asigna_pixel()`, `Imagen::cargarPGM()`, `Imagen::guardarPGM()`, `Imagen::num_columns()`, `Imagen::num_filas()`, y `Imagen::valor_pixel()`.

Referenciado por `main()`.

4.1.2.3. void mejorarContraste (const char **fich_E*, const char **fich_S*, float *gamma*)

Función que cambia el contraste de la imagen en función del parámetro gamma.

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

gamma Obtenemos el nuevo valor de cada píxel mediante la transformación $f(x) = 255 * (x/255)^{\gamma}$

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

gamma

Definición en la línea 50 del archivo funciones.cpp.

Hace referencia a Imagen::asigna_pixel(), Imagen::cargarPGM(), Imagen::guardarPGM(), Imagen::num_ - columnas(), Imagen::num_filas(), y Imagen::valor_pixel().

Referenciado por main().

4.1.2.4. void morphing (const char **fich_E*, const char **fich_S*, const char **prefijo*)

Función que genera una lista de imagenes interpoladas entre dos de referencia.

Parámetros:

fich_E dirección de la imagen A

fich_S dirección de la imagen B

prefijo dirección de las imagenes destino

Va estabilizando una imagen en relación a otra sumándole o restándole en cada ciclo 1 a cada píxel hasta llegar a los mismos niveles de gris de la segunda imagen. En cada ciclo guarda cada imagen intermedia

Parámetros:

fich_E dirección de la imagen A

fich_S dirección de la imagen B

prefijo dirección de las imagenes destino

Definición en la línea 223 del archivo funciones.cpp.

Hace referencia a Imagen::asigna_pixel(), Imagen::cargarPGM(), Imagen::guardarPGM(), Imagen::num_ - columnas(), Imagen::num_filas(), y Imagen::valor_pixel().

Referenciado por main().

4.1.2.5. void RGB2Gris (const char **fich_E*, const char **fich_S*)

Función que convierte una imagen RGB (PPM) a escala de grises (PGM).

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

Cada 3 bytes de la imagen PPM se corresponden a uno en la PGM, para pasar de uno a otro usamos una sencilla función: $f(r,g,b) = 0.2989*r + 0.587*g + 0.114*b$

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

Definición en la línea 25 del archivo funciones.cpp.

Hace referencia a Imagen::asigna_pixel(), Imagen::guardarPGM(), IMG_PPM, LeerImagenPPM(), LeerTipoImagen(), y Imagen::Reserva().

Referenciado por main().

4.1.2.6. void umbralizar (const char *fich_E, const char *fich_S, int &T)

Función que realiza un umbralizado automático de la imagen.

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

Una vez calculado el umbral T, recorre la imagen, convirtiendo en 255 aquellos píxeles con un gris \geq que T, y a 0 aquellos con un gris $<$ T

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

Definición en la línea 163 del archivo funciones.cpp.

Hace referencia a Imagen::asigna_pixel(), calcularUmbral(), Imagen::cargarPGM(), Imagen::guardarPGM(), Imagen::num_columnas(), Imagen::num_filas(), y Imagen::valor_pixel().

Referenciado por main().

4.2. Referencia del Archivo include/Imagen.h

Fichero de cabecera asociado a la biblioteca libImagen.a.

```
#include "imagenES.h"
```

Dependencia gráfica adjunta para Imagen.h:

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

Clases

- class [Imagen](#)

Clase que almacena la información de una imagen y se encarga de su gestión.

Tipos definidos

- typedef unsigned char [byte](#)

Tipo base de cada píxel.

4.2.1. Descripción detallada

Fichero de cabecera asociado a la biblioteca libImagen.a.

Autor:

Miguel Cantón Cortés

Implementación del TDA imagen (imagen digital de niveles de gris)

Definición en el archivo [Imagen.h](#).

4.2.2. Documentación de los tipos definidos

4.2.2.1. typedef unsigned char byte

Tipo base de cada píxel.

Definición en la línea 14 del archivo Imagen.h.

4.3. Referencia del Archivo `include/imagenES.h`

Fichero cabecera para la E/S de imágenes.

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

Enumeraciones

- enum `TipoImagen` { `IMG_DESCONOCIDO`, `IMG_PGM`, `IMG_PPM` }
Tipo de imagen.

Funciones

- `TipoImagen LeerTipoImagen` (const char *nombre)
Devuelve el tipo de imagen del archivo.
- unsigned char * `LeerImagenPPM` (const char *nombre, int &fils, int &cols)
Lee una imagen de tipo PPM.
- bool `EscribirImagenPPM` (const char *nombre, const unsigned char *datos, const int fils, const int cols)
Escribe una imagen de tipo PPM.
- unsigned char * `LeerImagenPGM` (const char *nombre, int &fils, int &cols)
Lee una imagen de tipo PGM.
- bool `EscribirImagenPGM` (const char *nombre, const unsigned char *datos, const int fils, const int cols)
Escribe una imagen de tipo PGM.

4.3.1. Descripción detallada

Fichero cabecera para la E/S de imágenes.

Permite la E/S de archivos de tipo PGM,PPM

Definición en el archivo `imagenES.h`.

4.3.2. Documentación de las enumeraciones

4.3.2.1. enum `TipoImagen`

Tipo de imagen.

Declara una serie de constantes para representar los distintos tipos de imágenes que se pueden manejar.

Ver también:

[LeerTipoImagen](#)

Valores de la enumeración:

IMG_DESCONOCIDO

IMG_PGM

IMG_PPM

Definición en la línea 20 del archivo imagenES.h.

4.3.3. Documentación de las funciones

4.3.3.1. **bool EscribirImagenPGM (const char * *nombre*, const unsigned char * *datos*, const int *fil*s, const int *col*s)**

Escribe una imagen de tipo PGM.

Parámetros:

nombre archivo a escribir

datos punteros a los $f \times c$ bytes que corresponden a los valores de los píxeles de la imagen de grises.

f filas de la imagen

c columnas de la imagen

Devuelve:

si ha tenido éxito en la escritura.

Definición en la línea 133 del archivo imagenES.cpp.

Referenciado por Imagen::guardarPGM(), y main().

4.3.3.2. **bool EscribirImagenPPM (const char * *nombre*, const unsigned char * *datos*, const int *fil*s, const int *col*s)**

Escribe una imagen de tipo PPM.

Parámetros:

nombre archivo a escribir

datos punteros a los $f \times c \times 3$ bytes que corresponden a los valores de los píxeles de la imagen en formato RGB.

f filas de la imagen

c columnas de la imagen

Devuelve:

si ha tenido éxito en la escritura.

Definición en la línea 116 del archivo imagenES.cpp.

4.3.3.3. unsigned char* LeerImagenPGM (const char * *nombre*, int & *filas*, int & *cols*)

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 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.

Definición en la línea 95 del archivo imagenES.cpp.

Hace referencia a IMG_PGM, LeerCabecera(), y LeerTipo().

Referenciado por Imagen::cargarPGM(), y main().

4.3.3.4. unsigned char* LeerImagenPPM (const char * *nombre*, int & *filas*, int & *cols*)

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 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.

Definición en la línea 74 del archivo imagenES.cpp.

Hace referencia a IMG_PPM, LeerCabecera(), y LeerTipo().

Referenciado por RGB2Gris().

4.3.3.5. 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](#)

Definición en la línea 36 del archivo imagenES.cpp.

Hace referencia a LeerTipo().

Referenciado por Imagen::cargarPGM(), y RGB2Gris().

4.4. Referencia del Archivo src/contraste.cpp

Fichero de aplicación de la función de mejora de contraste.

```
#include "Imagen.h"
#include "funciones.h"
#include <stdlib.h>
#include <iostream>
#include <cstdlib>
```

Dependencia gráfica adjunta para contraste.cpp:

Funciones

- `int main (int argc, char *argv[])`

4.4.1. Descripción detallada

Fichero de aplicación de la función de mejora de contraste.

Autor:

Miguel Cantón Cortés

Ver también:

[mejorarContraste](#)

Definición en el archivo [contraste.cpp](#).

4.4.2. Documentación de las funciones

4.4.2.1. `int main (int argc, char * argv[])`

Definición en la línea 17 del archivo `contraste.cpp`.

Hace referencia a `mejorarContraste()`.

4.5. Referencia del Archivo src/crearicono.cpp

Fichero de aplicación de la función de reducción de imagen.

```
#include "Imagen.h"
#include "funciones.h"
#include <stdlib.h>
#include <iostream>
#include <cstdlib>
```

Dependencia gráfica adjunta para crearicono.cpp:

Funciones

- `int main (int argc, char *argv[])`

4.5.1. Descripción detallada

Fichero de aplicación de la función de reducción de imagen.

Autor:

Miguel Cantón Cortés

Ver también:

[crearIcono](#)

Definición en el archivo [crearicono.cpp](#).

4.5.2. Documentación de las funciones

4.5.2.1. `int main (int argc, char * argv[])`

Definición en la línea 16 del archivo crearicono.cpp.

Hace referencia a crearIcono().

4.6. Referencia del Archivo `src/ecualizar.cpp`

Fichero de aplicación de la función de ecualización automática.

```
#include "Imagen.h"
#include "funciones.h"
#include <iostream>
#include <cstdlib>
```

Dependencia gráfica adjunta para `ecualizar.cpp`:

Funciones

- `int main (int argc, char *argv[])`

4.6.1. Descripción detallada

Fichero de aplicación de la función de ecualización automática.

Autor:

Miguel Cantón Cortés

Ver también:

[ecualizar](#)

Definición en el archivo [ecualizar.cpp](#).

4.6.2. Documentación de las funciones

4.6.2.1. `int main (int argc, char * argv[])`

Definición en la línea 15 del archivo `ecualizar.cpp`.

Hace referencia a `ecualizar()`.

4.7. Referencia del Archivo src/funciones.cpp

Fichero con definiciones de funciones extra para la modificación de imagenes.

```
#include "Imagen.h"
#include "imagenES.h"
#include <cstdio>
#include <cmath>
#include <cassert>
```

Dependencia gráfica adjunta para funciones.cpp:

Funciones

- void [RGB2Gris](#) (const char *fich_E, const char *fich_S)
Función que convierte una imagen RGB (PPM) a escala de grises (PGM).
- void [mejorarContraste](#) (const char *fich_E, const char *fich_S, float gamma)
Función que cambia el contraste de la imagen en función del parámetro gamma.
- void [ecualizar](#) (const char *fich_E, const char *fich_S)
Función que realiza un ecualizado automático de la imagen.
- int [calcularUmbral](#) (const [Imagen](#) &imagen, const int &T)
Función que calcula el umbral T de una imagen.
- void [umbralizar](#) (const char *fich_E, const char *fich_S, int &T)
Función que realiza un umbralizado automático de la imagen.
- void [crearIcono](#) (const char *fich_E, const char *fich_S, int factor)
Función que reduce el tamaño de una imagen en un factor "factor".
- void [morphing](#) (const char *fich_E, const char *fich_S, const char *prefijo)
Función que genera una lista de imagenes interpoladas entre dos de referencia.

4.7.1. Descripción detallada

Fichero con definiciones de funciones extra para la modificación de imagenes.

Autor:

Miguel Cantón Cortés

Estas funciones sirven de ejemplo de la clase [Imagen](#)

Definición en el archivo [funciones.cpp](#).

4.7.2. Documentación de las funciones

4.7.2.1. `int calcularUmbral (const Imagen & imagen, const int & T)`

Función que calcula el umbral *T* de una imagen.

Se trata de una función recursiva que calcula la media de los niveles de gris de la imagen y crea otras dos nuevas imagenes una con los valores por encima a la media y otra con los valores menores. Repite este proceso con cada subimagen hasta que las medias se estabilizan

Parámetros:

imagen imagen original

T umbral

Definición en la línea 112 del archivo funciones.cpp.

Hace referencia a `Imagen::asigna_pixel()`, `Imagen::num_columnas()`, `Imagen::num_filas()`, `Imagen::Reserva()`, y `Imagen::valor_pixel()`.

Referenciado por `umbralizar()`.

4.7.2.2. `void crearIcono (const char *fich_E, const char *fich_S, int factor)`

Función que reduce el tamaño de una imagen en un factor "factor".

Agrupar la imagen en bloques de `factor*factor` píxeles y calcula su media. Esta media se corresponde con cada píxel de la imagen reducida.

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

factor factor de reducción de la imagen

Definición en la línea 187 del archivo funciones.cpp.

Hace referencia a `Imagen::asigna_pixel()`, `Imagen::cargarPGM()`, `Imagen::guardarPGM()`, `Imagen::num_columnas()`, `Imagen::num_filas()`, `Imagen::Reserva()`, y `Imagen::valor_pixel()`.

Referenciado por `main()`.

4.7.2.3. `void ecualizar (const char *fich_E, const char *fich_S)`

Función que realiza un ecualizado automático de la imagen.

Calculamos el numero de apariciones de cada nivel de gris y lo almacenamo en el vector histograma. Calculamos las probabilidades de cada nivel de gris, dividiendo su número de apariciones entre el total de píxeles, y lo almacenamos en el vector probabilidades. Finalmente usando el vector de probabilidades, y aplicando una fórmula, obtenemos un nuevo vector con los nuevos valores de grises, transformacion

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

Definición en la línea 72 del archivo funciones.cpp.

Hace referencia a Imagen::asigna_pixel(), Imagen::cargarPGM(), Imagen::guardarPGM(), Imagen::num_columns(), Imagen::num_filas(), y Imagen::valor_pixel().

Referenciado por main().

4.7.2.4. void mejorarContraste (const char *fich_E, const char *fich_S, float gamma)

Función que cambia el contraste de la imagen en función del parámetro gamma.

Obtenemos el nuevo valor de cada píxel mediante la transformación $f(x) = 255 * (x/255)^\gamma$

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

gamma

Definición en la línea 50 del archivo funciones.cpp.

Hace referencia a Imagen::asigna_pixel(), Imagen::cargarPGM(), Imagen::guardarPGM(), Imagen::num_columns(), Imagen::num_filas(), y Imagen::valor_pixel().

Referenciado por main().

4.7.2.5. void morphing (const char *fich_E, const char *fich_S, const char *prefijo)

Función que genera una lista de imágenes interpoladas entre dos de referencia.

Va estabilizando una imagen en relación a otra sumándole o restándole en cada ciclo 1 a cada píxel hasta llegar a los mismos niveles de gris de la segunda imagen. En cada ciclo guarda cada imagen intermedia

Parámetros:

fich_E dirección de la imagen A

fich_S dirección de la imagen B

prefijo dirección de las imágenes destino

Definición en la línea 223 del archivo funciones.cpp.

Hace referencia a Imagen::asigna_pixel(), Imagen::cargarPGM(), Imagen::guardarPGM(), Imagen::num_columns(), Imagen::num_filas(), y Imagen::valor_pixel().

Referenciado por main().

4.7.2.6. void RGB2Gris (const char *fich_E, const char *fich_S)

Función que convierte una imagen RGB (PPM) a escala de grises (PGM).

Cada 3 bytes de la imagen PPM se corresponden a uno en la PGM, para pasar de uno a otro usamos una sencilla función: $f(r,g,b) = 0.2989*r + 0.587*g + 0.114*b$

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

Definición en la línea 25 del archivo funciones.cpp.

Hace referencia a Imagen::asigna_pixel(), Imagen::guardarPGM(), IMG_PPM, LeerImagenPPM(), LeerTipoImagen(), y Imagen::Reserva().

Referenciado por main().

4.7.2.7. void umbralizar (const char **fich_E*, const char **fich_S*, int & *T*)

Función que realiza un umbralizado automático de la imagen.

Una vez calculado el umbral T, recorre la imagen, convirtiendo en 255 aquellos píxeles con un gris \geq que T, y a 0 aquellos con un gris $<$ T

Parámetros:

fich_E dirección de la imagen origen

fich_S dirección de la imagen destino

Definición en la línea 163 del archivo funciones.cpp.

Hace referencia a Imagen::asigna_pixel(), calcularUmbral(), Imagen::cargarPGM(), Imagen::guardarPGM(), Imagen::num_columnas(), Imagen::num_filas(), y Imagen::valor_pixel().

Referenciado por main().

4.8. Referencia del Archivo src/Imagen.cpp

Fichero con las definiciones de la clase [Imagen](#) asociada a la biblioteca libImagen.a.

```
#include <cassert>
#include "Imagen.h"
```

Dependencia gráfica adjunta para Imagen.cpp:

4.8.1. Descripción detallada

Fichero con las definiciones de la clase [Imagen](#) asociada a la biblioteca libImagen.a.

Autor:

Miguel Cantón Cortés

Implementación del TDA imagen (imagen digital de niveles de gris)

Definición en el archivo [Imagen.cpp](#).

4.9. Referencia del Archivo src/imagenES.cpp

Fichero con definiciones para la E/S de imágenes.

```
#include <fstream>
#include <string>
#include "imagenES.h"
```

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.

4.9.1. Descripción detallada

Fichero con definiciones para la E/S de imágenes.

Permite la E/S de archivos de tipo PGM,PPM

Definición en el archivo [imagenES.cpp](#).

4.9.2. Documentación de las funciones

4.9.2.1. bool EscribirImagenPGM (const char * *nombre*, const unsigned char * *datos*, const int *fils*, const int *cols*)

Escribe una imagen de tipo PGM.

Parámetros:

nombre archivo a escribir

datos punteros a los $f \times c$ bytes que corresponden a los valores de los píxeles de la imagen de grises.
f filas de la imagen
c columnas de la imagen

Devuelve:

si ha tenido éxito en la escritura.

Definición en la línea 133 del archivo imagenES.cpp.

Referenciado por Imagen::guardarPGM(), y main().

4.9.2.2. bool EscribirImagenPPM (const char * *nombre*, const unsigned char * *datos*, const int *filas*, const int *cols*)

Escribe una imagen de tipo PPM.

Parámetros:

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

Devuelve:

si ha tenido éxito en la escritura.

Definición en la línea 116 del archivo imagenES.cpp.

4.9.2.3. bool LeerCabecera (ifstream & *f*, int & *filas*, int & *cols*)

Definición en la línea 55 del archivo imagenES.cpp.

Hace referencia a SaltarSeparadores().

Referenciado por LeerImagenPGM(), y LeerImagenPPM().

4.9.2.4. unsigned char* LeerImagenPGM (const char * *nombre*, int & *filas*, int & *cols*)

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 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.

Definición en la línea 95 del archivo imagenES.cpp.

Hace referencia a IMG_PGM, LeerCabecera(), y LeerTipo().

Referenciado por Imagen::cargarPGM(), y main().

4.9.2.5. unsigned char* LeerImagenPPM (const char * *nombre*, int & *filas*, int & *cols*)

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 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.

Definición en la línea 74 del archivo imagenES.cpp.

Hace referencia a IMG_PPM, LeerCabecera(), y LeerTipo().

Referenciado por RGB2Gris().

4.9.2.6. TipoImagen LeerTipo (ifstream & *f*)

Definición en la línea 17 del archivo imagenES.cpp.

Hace referencia a IMG_DESCONOCIDO, IMG_PGM, y IMG_PPM.

Referenciado por LeerImagenPGM(), LeerImagenPPM(), y LeerTipoImagen().

4.9.2.7. 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](#)

Definición en la línea 36 del archivo imagenES.cpp.

Hace referencia a LeerTipo().

Referenciado por Imagen::cargarPGM(), y RGB2Gris().

4.9.2.8. char SaltarSeparadores (ifstream & f)

Definición en la línea 44 del archivo imagenES.cpp.

Referenciado por LeerCabecera().

4.10. Referencia del Archivo src/morphing.cpp

Fichero de aplicación de la función `morphing`.

```
#include "Imagen.h"
#include "funciones.h"
#include <iostream>
#include <cstdlib>
```

Dependencia gráfica adjunta para `morphing.cpp`:

Funciones

- `int main (int argc, char *argv[])`

4.10.1. Descripción detallada

Fichero de aplicación de la función `morphing`.

Autor:

Miguel Cantón Cortés

Ver también:

[morphing](#)

Definición en el archivo [morphing.cpp](#).

4.10.2. Documentación de las funciones

4.10.2.1. `int main (int argc, char * argv[])`

Definición en la línea 15 del archivo `morphing.cpp`.

Hace referencia a `morphing()`.

4.11. Referencia del Archivo src/negativo.cpp

Calcula el negativo de una imagen PGM.

```
#include <iostream>
#include <cstring>
#include <cstdlib>
#include "imagenES.h"
```

Dependencia gráfica adjunta para negativo.cpp:

Funciones

- `int main (int argc, char *argv[])`

4.11.1. Descripción detallada

Calcula el negativo de una imagen PGM.

Formato: `negativo <FichImagenOriginal> <FichImagenDestino>` donde: `<FichImagenOriginal>` es la imagen PGM a la que se va a calcular su negativo. No se modifica. `<FichImagenDestino>` es la imagen PGM resultado de calcular el negativo de `<FichImagenOriginal>`.

Definición en el archivo [negativo.cpp](#).

4.11.2. Documentación de las funciones

4.11.2.1. `int main (int argc, char * argv[])`

Definición en la línea 22 del archivo `negativo.cpp`.

Hace referencia a `EscribirImagenPGM()`, y `LeerImagenPGM()`.

4.12. Referencia del Archivo `src/rgb2gris.cpp`

Fichero de aplicación de la función conversión a escala de grises.

```
#include "Imagen.h"
#include "funciones.h"
#include <iostream>
#include <cstdlib>
```

Dependencia gráfica adjunta para `rgb2gris.cpp`:

Funciones

- `int main (int argc, char *argv[])`

4.12.1. Descripción detallada

Fichero de aplicación de la función conversión a escala de grises.

Autor:

Miguel Cantón Cortés

Ver también:

[RGB2Gris](#)

Definición en el archivo [rgb2gris.cpp](#).

4.12.2. Documentación de las funciones

4.12.2.1. `int main (int argc, char * argv[])`

Definición en la línea 15 del archivo `rgb2gris.cpp`.

Hace referencia a `RGB2Gris()`.

4.13. Referencia del Archivo src/umbralizar.cpp

Fichero de aplicación de la función de umbralización automática.

```
#include "Imagen.h"
#include "funciones.h"
#include <iostream>
#include <cstdlib>
```

Dependencia gráfica adjunta para umbralizar.cpp:

Funciones

- `int main (int argc, char *argv[])`

4.13.1. Descripción detallada

Fichero de aplicación de la función de umbralización automática.

Autor:

Miguel Cantón Cortés

Ver también:

[umbralizar](#)

Definición en el archivo [umbralizar.cpp](#).

4.13.2. Documentación de las funciones

4.13.2.1. `int main (int argc, char * argv[])`

Definición en la línea 15 del archivo umbralizar.cpp.

Hace referencia a umbralizar().

Índice alfabético

- ~Imagen
 - Imagen, [6](#)
- asigna_pixel
 - Imagen, [8](#)
- byte
 - Imagen.h, [15](#)
- calcularUmbral
 - funciones.cpp, [24](#)
- cargarPGM
 - Imagen, [7](#)
- contraste.cpp
 - main, [20](#)
- crearIcono
 - funciones.cpp, [24](#)
 - funciones.h, [12](#)
- crearicono.cpp
 - main, [21](#)
- ecualizar
 - funciones.cpp, [24](#)
 - funciones.h, [12](#)
- ecualizar.cpp
 - main, [22](#)
- EscribirImagenPGM
 - imagenES.cpp, [28](#)
 - imagenES.h, [17](#)
- EscribirImagenPPM
 - imagenES.cpp, [29](#)
 - imagenES.h, [17](#)
- funciones.cpp
 - calcularUmbral, [24](#)
 - crearIcono, [24](#)
 - ecualizar, [24](#)
 - mejorarContraste, [25](#)
 - morphing, [25](#)
 - RGB2Gris, [25](#)
 - umbralizar, [26](#)
- funciones.h
 - crearIcono, [12](#)
 - ecualizar, [12](#)
 - mejorarContraste, [12](#)
 - morphing, [13](#)
 - RGB2Gris, [13](#)
 - umbralizar, [14](#)
- guardarPGM
 - Imagen, [7](#)
- Imagen, [5](#)
 - ~Imagen, [6](#)
 - asigna_pixel, [8](#)
 - cargarPGM, [7](#)
 - guardarPGM, [7](#)
 - Imagen, [6](#)
 - num_columnas, [8](#)
 - num_filas, [8](#)
 - operator=, [7](#)
 - Reserva, [7](#)
 - valor_pixel, [9](#)
- Imagen.h
 - byte, [15](#)
- imagenES.cpp
 - EscribirImagenPGM, [28](#)
 - EscribirImagenPPM, [29](#)
 - LeerCabecera, [29](#)
 - LeerImagenPGM, [29](#)
 - LeerImagenPPM, [30](#)
 - LeerTipo, [30](#)
 - LeerTipoImagen, [30](#)
 - SaltarSeparadores, [31](#)
- imagenES.h
 - EscribirImagenPGM, [17](#)
 - EscribirImagenPPM, [17](#)
 - IMG_DESCONOCIDO, [17](#)
 - IMG_PGM, [17](#)
 - IMG_PPM, [17](#)
 - LeerImagenPGM, [17](#)
 - LeerImagenPPM, [18](#)
 - LeerTipoImagen, [18](#)
 - TipoImagen, [16](#)
- IMG_DESCONOCIDO
 - imagenES.h, [17](#)
- IMG_PGM
 - imagenES.h, [17](#)
- IMG_PPM
 - imagenES.h, [17](#)

include/funciones.h, 11
include/Imagen.h, 15
include/imagenES.h, 16

LeerCabecera
 imagenES.cpp, 29

LeerImagenPGM
 imagenES.cpp, 29
 imagenES.h, 17

LeerImagenPPM
 imagenES.cpp, 30
 imagenES.h, 18

LeerTipo
 imagenES.cpp, 30

LeerTipoImagen
 imagenES.cpp, 30
 imagenES.h, 18

main
 contraste.cpp, 20
 crearicono.cpp, 21
 ecualizar.cpp, 22
 morphing.cpp, 32
 negativo.cpp, 33
 rgb2gris.cpp, 34
 umbralizar.cpp, 35

mejorarContraste
 funciones.cpp, 25
 funciones.h, 12

morphing
 funciones.cpp, 25
 funciones.h, 13

morphing.cpp
 main, 32

negativo.cpp
 main, 33

num_columnas
 Imagen, 8

num_filas
 Imagen, 8

operator=
 Imagen, 7

Reserva
 Imagen, 7

RGB2Gris
 funciones.cpp, 25
 funciones.h, 13

rgb2gris.cpp
 main, 34

SaltarSeparadores
 imagenES.cpp, 31

src/contraste.cpp, 20
src/crearicono.cpp, 21
src/ecualizar.cpp, 22
src/funciones.cpp, 23
src/Imagen.cpp, 27
src/imagenES.cpp, 28
src/morphing.cpp, 32
src/negativo.cpp, 33
src/rgb2gris.cpp, 34
src/umbralizar.cpp, 35

TipoImagen
 imagenES.h, 16

umbralizar
 funciones.cpp, 26
 funciones.h, 14

umbralizar.cpp
 main, 35

valor_pixel
 Imagen, 9