Imageprocessing

1.0

Generated by Doxygen 1.7.4

Wed Apr 5 2017 17:00:50

Contents

Chapter 1

Test List

Member Image::Dimension() basic_funcions.cpp

Image I;

I.Dimension();

Member Image::Filtering(arma::mat, int, int) filtering_median.cpp filtering_mean.cpp

mat matrix;

int l = 5;

int flag = 0;

Filtering(matrix, l, flag);

Member Image::Four_windows_opencv(const char ∗, const char ∗, const char ∗, const char ∗, const char ∗, const ch
main.cpp

const char∗ imag1= "src/Resources/images_salt_and_pepper/lady_256_0_1.pgm";

const char∗ imag2= "src/Resources/images_salt_and_pepper/lady_256_1.pgm";

const char∗ imag3= "src/Resources/images_salt_and_pepper/lady_256_10.pgm";

const char∗ imag4= "src/Resources/images_salt_and_pepper/lady_256_100.pgm";

Four_windows_opencv(imag1, imag2, imag3, imag4, label1, label2, label3, label4);

Member Image::Gaussian_noise(double) gaussian_noise.cpp

int nivel=0.1;

Impulsive_uniform_noise(nivel1);

Member Image::Get_pixel(int, int) basic_funcions.cpp

Image I;

int row=5;

int col=5;

I.Get_pixel(row, col);

Member Image::Get_pixels() basic_funcions.cpp

Image I;

I.Get_pixels();

Member Image::Image() basic_funcions.cpp

Image I;

I.Dimension();

Member Image::Image_load(arma::mat, std::string) basic_funcions.cpp

mat matrix;

string ruta = "src/Resources/images/house.256.pgm";

Image_load(matrix, ruta);

Member Image::Impulsive_uniform_noise(int) uniform_impulsive_noise.cpp

int nivel=0.1;

Impulsive_uniform_noise(nivel);

Member Image::Lost_pixels_noise(int) lost_Pixels_noise.cpp

int nivel=0.1;

Lost_pixels_noise(nivel);

Member Image::MAE(arma::mat, arma::mat) filtering_mean.cpp

mat matrix_r;

mat matriz_f;

MAE(matrix_r, matriz_f);

Member Image::MSE(arma::mat, arma::mat) filtering_mean.cpp

mat matrix_r;

mat matriz_f;

MSE(matrix_r, matriz_f);

Member Image::Noise_remover(arma::mat, int) noise_remover.cpp

mat matrix;

int image_tam = 256;

Noise_remover(matrix, image_tam);

Member Image::Overlap(arma::mat) overlap.cpp

mat matrix;

Overlap(matrix);

Member Image::PSNR(arma::mat, arma::mat) filtering_mean.cpp

mat matrix_r;

mat matriz_f;

PSNR(matrix_r, matriz_f);

Member Image::Salt_and_pepper_noise(int) salt_pepper.cpp

int nivel=0.1;

Salt_and_pepper_noise(nivel);

Member Image::SaveImage(arma::mat, std::string) gaussian_noise.cpp

mat matrix;

string ruta= "src/Resources/images/house.256.pgm";

SaveImage(matrix, ruta);

Member Image::Two_windows_opencv(const char ∗, const char ∗, std::string, std::string) main.cpp

const char∗ imag1= "src/Resources/images_salt_and_pepper/lady_256_0_1.pgm";

const char∗ imag2= "src/Resources/images_salt_and_pepper/lady_256_1.pgm";

string label1 = "Imagen1";

string label2 = "Imagen2";

Two_windows_opencv(imag1, imag2, label1, label2);

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 Image::h Class Reference

The documentation for this class was generated from the following file:

- include/Image.hpp

4.2 Image Class Reference

```
#include <Image.hpp>
```

Classes

- class h

Public Member Functions

- arma::mat Image_load (arma::mat, std::string)

  Carga de Imagen pgm a mat.
- Image ()

  Constructor de la clase.
- Image (arma::mat)
- void Dimension ()

  Obtiene las dimensiones de la matriz.
- int Get_pixel (int, int)

  Obtiene el valor de un pixel.
- void Get_pixels ()

  Imprime todos los pixeles de la matriz imagen.

- arma::mat Gaussian_noise (double)

  Aplica Ruido Gaussiano a una matriz.
- arma::mat Lost_pixels_noise (int)

  Aplica Perdida de pixeles a una matriz.
- arma::mat Salt_and_pepper_noise (int)

  Aplica Ruido Sal y Pimienta a una matriz.
- arma::mat Impulsive_uniform_noise (int)

  Aplica Ruido Impulsivo Uniforme a una matriz.
- void SaveImage (arma::mat, std::string)

  Guarda la matriz en formato pgm.
- double PSNR (arma::mat, arma::mat)

  Calculo del PSNR.
- double MAE (arma::mat, arma::mat)

  Calculo del MAE.
- double MSE (arma::mat, arma::mat)

  Calculo del MSE.
- arma::mat Filtering (arma::mat, int, int)

  Aplica un filtrado usando el promedio o la mediana para remover ruido.
- void Four_windows_opencv (const char ∗, const char ∗, const char ∗, const char ∗, const char ∗, const char ∗, const char ∗, const char ∗)

  Carga cuatro imagenes en una ventana de opencv.
- void Two_windows_opencv (const char ∗, const char ∗, std::string, std::string)

  Carga dos imagenes en una ventana de opencv.
- arma::mat Noise_remover (arma::mat, int)

  Remocion de Ruido no solapado (ventana 8x8)
- arma::mat Overlap (arma::mat)

  Remocion de Ruido solapado (ventana 8x8)
- arma::mat Idtc_Robusta (arma::mat, int)
- double wmedianf (arma::vec &, arma::vec &)
- arma::mat Diccionary ()
- arma::vec Fast_awmr (arma::vec &y, arma::mat &A, int sparsity, int itmax, double beta, double tol, double epsilon, int numcoefperiter, double Kpar)
- ∼Image ()

Private Attributes

- int cols
- int rows
- arma::mat matrix

4.2.1 Constructor & Destructor Documentation

4.2.1.1 Image::Image ( )

Constructor de la clase.

Descripcion: Constructor de la clase Image para acceder a sus metodos internos

Test

> basic_funcions.cpp
> Image I;
> I.Dimension();

4.2.1.2 Image::Image (arma::mat)

4.2.1.3 Image::∼Image ( )

4.2.2 Member Function Documentation

4.2.2.1 mat Image::Diccionary ( )

4.2.2.2 void Image::Dimension ( )

Obtiene las dimensiones de la matriz.

Descripcion: Permite obtener la dimension nxn de la matriz.

Returns

> Retorna la dimension de la imagen nxn.

Test

> basic_funcions.cpp
> Image I;
> I.Dimension();

4.2.2.3 vec Image::Fast_awmr (arma::vec & *y*, arma::mat & *A*, int *sparsity*, int *itmax*, double *beta*, double *tol*, double *epsilon*, int *numcoefperiter*, double *Kpar*)

4.2.2.4 mat Image::Filtering (arma::mat , int , int)

Aplica un filtrado usando el promedio o la mediana para remover ruido.

Descripcion: Permite aplicar un filtrado a una matriz, utilizando el sparse. Se selecciona una ventana de lxl a la cual se le aplica un promedio o la mediana para restaurar la imagen. Se calcula utilizando el tamaño de la ventana l y el pixel central de la ventana, luego se aplica el promedio o mediana de los pixels adyacentes al central.

Parameters

matrix	es la matriz con ruido a la cual se le aplicara el filtrado.
l	es el tamaño de la ventana
flag	es la bandera que indica si se efectua el promedio o la mediana. Donde el promedio es igual 0 y la mediana igual a 1

Returns

filt_matrix: Retorna la matriz con el filtrado.

Test

filtering_median.cpp filtering_mean.cpp
mat matrix;
int l = 5;
int flag = 0;
Filtering(matrix, l, flag);

4.2.2.5 void Image::Four_windows_opencv (const char ∗ *imag1,* const char ∗ *imag2,* const char ∗ *imag3,* const char ∗ *imag4,* const char ∗ *label1,* const char ∗ *label2,* const char ∗ *label3,* const char ∗ *label4*)

Carga cuatro imagenes en una ventana de opencv.

Descripcion: Permite cargar cuatro imagenes en una ventana de opencv

Parameters

imag1	es la ruta o ubicación de la imagen1
imag2	es la ruta o ubicación de la imagen2 ....Imag3 ... Imag4
label1	es el nombre de la imagen1
label2	es el nombre de la imagen2 ....Imag3 ... Imag4

Returns

Abre la ventana opencv con cuatro imagenes

Test

main.cpp
const char∗ imag1= "src/Resources/images_salt_and_pepper/lady_256_0_1.pgm";
const char∗ imag2= "src/Resources/images_salt_and_pepper/lady_256_1.pgm";
const char∗ imag3= "src/Resources/images_salt_and_pepper/lady_256_10.pgm";
const char∗ imag4= "src/Resources/images_salt_and_pepper/lady_256_100.pgm";
Four_windows_opencv(imag1, imag2, imag3, imag4, label1, label2, label3, label4);

4.2.2.6 mat Image::Gaussian_noise (double *nivel*)

Aplica Ruido Gaussiano a una matriz.

Descripcion: Permite aplicarle Ruido Gaussiano a una matriz. El Ruido Gaussiano es una matriz de media 0 y con una desviacion estandar variable.

Parameters

nivel	es el nivel de varianza variable Parametros Internos: A es una matriz gaussiana de dimensiones iguales a la matriz y de media 0 y varianza 1 B es una matriz donde se almacena la matriz gaussiana de media 0 y el calculo de una varianza determinada std es la desviacion estandar

Returns

value: Retorna la matriz con el ruido aplicado

Test

gaussian_noise.cpp
int nivel=0.1;
Impulsive_uniform_noise(nivel1);

4.2.2.7 int Image::Get_pixel (int *row,* int *col*)

Obtiene el valor de un pixel.

Descripcion: Permite obtener el valor de un pixel de la matriz de la imagen original

Parameters

row	es el numero de filas de la matriz
col	es el numero de columnas de la matriz

Returns

Retorna el valor (value) de un pixel

Test

basic_funcions.cpp
Image I;
int row=5;
int col=5;
I.Get_pixel(row, col);

4.2.2.8 void Image::Get_pixels ( )

Imprime todos los pixeles de la matriz imagen.

Descripcion: Permite obtener e imprimir todos los pixeles de la imagen.

Returns

value: Retorna todos los pixeles.

Test

basic_funcions.cpp
Image I;
I.Get_pixels();

4.2.2.9 mat Image::Idtc_Robusta (arma::mat , int)

4.2.2.10 mat Image::Image_load (arma::mat , std::string)

Carga de Imagen pgm a mat.

Descripcion: Permite cargar la imagen en formato pgm a mat

Parameters

matrix	es la matriz que contine los pixeles de la imagen.
ruta	es la ubicacion o localizacion de la imagen.

Returns

Retorna si la matriz fue cargada o si hubo problemas en la carga.

Test

basic_funcions.cpp
mat matrix;
string ruta = "src/Resources/images/house.256.pgm";
Image_load(matrix, ruta);

4.2.2.11 mat Image::Impulsive_uniform_noise (int *nivel*)

Aplica Ruido Impulsivo Uniforme a una matriz.

Descripcion: Permite aplicarle Ruido Impulsivo Uniforme a una matriz

Parameters

nivel	es el porcentaje de ruido.

Returns

value: Retorna la matriz con el ruido aplicado.

Test

uniform_impulsive_noise.cpp
int nivel=0.1;

Impulsive_uniform_noise(nivel);

4.2.2.12 mat Image::Lost_pixels_noise (int *nivel*)

Aplica Perdida de pixeles a una matriz.

Descripcion: Permite aplicarle Perdida de Pixeles a una matriz

Parameters

nivel	es el porcentaje de ruido

Returns

value: Retorna la matriz con el ruido aplicado

Test

lost_Pixels_noise.cpp
int nivel=0.1;
Lost_pixels_noise(nivel);

4.2.2.13 double Image::MAE (arma::mat , arma::mat)

Calculo del MAE.

Descripcion: Permite calcular el MAE (Error promedio absoluto) de una imagen con ruido y una imagen filtrada

Parameters

matrix_r	es la matriz con ruido.
matriz_f	es la matriz filtrada o recuperada

Returns

Retorna el valor del MAE

Test

filtering_mean.cpp
mat matrix_r;
mat matriz_f;
MAE(matrix_r, matriz_f);

4.2.2.14 double Image::MSE (arma::mat , arma::mat)

Calculo del MSE.

Descripcion: Permite calcular el MSE (Error cuadratico medio) de una imagen con ruido y una imagen filtrada

Parameters

matrix_r	es la matriz con ruido.
matriz_f	es la matriz filtrada o recuperada

Returns

Retorna el valor del MSE

Test

filtering_mean.cpp
mat matrix_r;
mat matriz_f;
MSE(matrix_r, matriz_f);

4.2.2.15 mat Image::Noise_remover (arma::mat , int)

Remocion de Ruido no solapado (ventana 8x8)

Descripcion: Permite remover ruido mediante el uso de una ventana 8x8 no solapado

Parameters

matrix	es la matriz que se le removerá el ruido.
image_tam	es el tamaño de la imagen.

Returns

removermatrix: Retorna la matriz sin ruido.

Test

noise_remover.cpp
mat matrix;
int image_tam = 256;
Noise_remover(matrix, image_tam);

4.2.2.16 mat Image::Overlap (arma::mat)

Remocion de Ruido solapado (ventana 8x8)

Descripcion: Permite remover ruido mediante el uso de una ventana 8x8 solapado

Parameters

matrix	es la matriz que se le removerá el ruido.

Returns

removermatrix: Retorna la matriz sin ruido.

Test

overlap.cpp
mat matrix;
Overlap(matrix);

4.2.2.17 double Image::PSNR (arma::mat , arma::mat)

Calculo del PSNR.

Descripcion: Permite calcular el PSNR de una imagen con ruido y una imagen filtrada

Parameters

matrix_r	es la matriz con ruido.
matriz_f	es la matriz filtrada o recuperada

Returns

Retorna el valor del PSNR

Test

filtering_mean.cpp
mat matrix_r;
mat matriz_f;
PSNR(matrix_r, matriz_f);

4.2.2.18 mat Image::Salt_and_pepper_noise (int *nivel*)

Aplica Ruido Sal y Pimienta a una matriz.

Descripcion: Permite aplicarle Ruido Sal y Pimienta a una matriz

Parameters

nivel	es el porcentaje de ruido

Returns

Retorna la matriz con el ruido (RG)

Test

salt_pepper.cpp
int nivel=0.1;
Salt_and_pepper_noise(nivel);

4.2.2.19 void Image::SaveImage (arma::mat , std::string)

Guarda la matriz en formato pgm.

Descripcion: Permite guardar una matriz mat en formato pgm

Parameters

matrix	matriz que contine los pixeles de la imagen.
ruta	ubicacion o localizacion de la imagen.

Returns

Retorna si la matriz fue guardada o si hubo problemas al guardar.

Test

gaussian_noise.cpp
mat matrix;
string ruta= "src/Resources/images/house.256.pgm";
SaveImage(matrix, ruta);

4.2.2.20 void Image::Two_windows_opencv (const char ∗ , const char ∗ , std::string , std::string)

Carga dos imagenes en una ventana de opencv.

Descripcion: Permite cargar dos imagenes en una ventana de opencv con sus respectivas etiquetas (label).

Parameters

imag1	es la ruta o ubicación de la imagen1.
imag2	es la ruta o ubicación de la imagen2.
label1	es el nombre de la imagen1.
label2	es el nombre de la imagen2.

Returns

Abre la ventana opencv con dos imagenes.

Test

main.cpp
const char∗ imag1= "src/Resources/images_salt_and_pepper/lady_256_0_1.pgm";
const char∗ imag2= "src/Resources/images_salt_and_pepper/lady_256_1.pgm";
string label1 = "Imagen1";
string label2 = "Imagen2";
Two_windows_opencv(imag1, imag2, label1, label2);

4.2.2.21 double Image::wmedianf (arma::vec & , arma::vec &)

4.2.3 Member Data Documentation

4.2.3.1 int **Image::cols** `[private]`

4.2.3.2 arma::mat **Image::matrix** `[private]`

4.2.3.3 int **Image::rows** `[private]`

The documentation for this class was generated from the following files:

- include/Image.hpp
- src/Four_windows_opencv.cpp
- src/Image_load.cpp
- src/Two_windows_opencv.cpp

Chapter 5

File Documentation

5.1 examples/basic_funcions.cpp File Reference

```
#include "../include/Image.hpp"
#include <iostream>
#include <armadillo>
```

Functions

- int main ()

5.1.1 Function Documentation

5.1.1.1 int main ( )

5.2 examples/filtering_mean.cpp File Reference

```
#include <iostream>
#include <armadillo>
#include "Image.hpp"
```

Functions

- int main ()

5.2.1 Function Documentation

5.2.1.1 int main ( )

5.3 examples/filtering_median.cpp File Reference

```
#include <iostream>
#include <armadillo>
#include "Image.hpp"
```

Functions

- int main ()

5.3.1 Function Documentation

5.3.1.1 int main ( )

5.4 examples/gaussian_noise.cpp File Reference

```
#include <iostream>
#include <armadillo>
#include "Image.hpp"
```

Functions

- int main ()

5.4.1 Function Documentation

5.4.1.1 int main ( )

5.5 examples/idtc_Robusta_mean.cpp File Reference

```
#include <iostream>
#include <armadillo>
#include "Image.hpp"
```

Functions

- int main ()

5.5.1 Function Documentation

5.5.1.1 int main ( )

5.6 examples/idtc_Robusta_median.cpp File Reference

```
#include <iostream>
#include <armadillo>
#include "Image.hpp"
```

Functions

- int main ()

5.6.1 Function Documentation

5.6.1.1 int main ( )

5.7 examples/lost_Pixels_noise.cpp File Reference

```
#include <iostream>
#include <armadillo>
#include "Image.hpp"
```

Functions

- int main ()

5.7.1 Function Documentation

5.7.1.1 int main ( )

5.8 examples/noise_remover.cpp File Reference

```
#include <iostream>
#include <armadillo>
#include "Image.hpp"
```

Functions

- int main ()

5.8.1 Function Documentation

5.8.1.1 int main ( )

5.9 examples/overlap.cpp File Reference

```
#include <iostream>
#include <armadillo>
#include "Image.hpp"
```

Functions

- int main ()

5.9.1 Function Documentation

5.9.1.1 int main ( )

5.10 examples/salt_pepper.cpp File Reference

```
#include <iostream>
#include <armadillo>
#include "Image.hpp"
```

Functions

- int main ()

5.10.1 Function Documentation

5.10.1.1 int main ( )

5.11 examples/uniform_impulsive_noise.cpp File Reference

```
#include <iostream>
#include <armadillo>
```

```
#include "Image.hpp"
```

Functions

- int main ()

5.11.1 Function Documentation

5.11.1.1 int main ( )

5.12 include/Image.hpp File Reference

```
#include <iostream>
#include <armadillo>
```

Classes

- class Image

5.13 src/Four_windows_opencv.cpp File Reference

```
#include "../include/Image.hpp"
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/core/utility.hpp>
#include <iostream>
```

5.14 src/Image_load.cpp File Reference

```
#include "../include/Image.hpp"
#include <iostream>
#include <armadillo>
```

Functions

- double Average_Median (vec vector, int flag)

  Calcula el promedio y la mediana.

5.14.1 Function Documentation

5.14.1.1 double Image::Average_Median (vec *vector,* int *flag*)

Calcula el promedio y la mediana.

Descripcion: Permite calcular el promedio y la mediana.

Parameters

vector	es el vector al cual se le calculará el promedio o la mediana
flag	cuando flag = 0 se calcula el promedio y cuando flag = 1 se calcula la mediana

Returns

value: Retorna la matriz con el ruido
vec B;
Average_Median(B, 0); //Promedio-Media
Average_Median(B, 1); //Mediana

5.15 src/main.cpp File Reference

```
#include <iostream>
#include <armadillo>
#include "../include/Image.hpp"
```

Functions

- int main ()

  Titulo: Funcion de Inicio.

5.15.1 Function Documentation

5.15.1.1 int main ( )

Titulo: Funcion de Inicio.

Descripcion: Comienzo de codigo.

FA = I.Filtering(SC, 5);

5.16 src/Two_windows_opencv.cpp File Reference

```
#include "../include/Image.hpp"
```

```
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/core/utility.hpp>
#include <iostream>
```

Index