PPM/PGM Image library

v1.0 by Romuald GRIGNON

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 PPM_IMG Struct Reference

Structure for a PPM image.

Public Attributes

- int W
- int H
- int range
- int nbColors
- int hdrSize
- int dataSize
- unsigned char * pHeader
- unsigned char * pData

3.1.1 Detailed Description

Structure for a PPM image.

3.1.2 Member Data Documentation

3.1.2.1 W

```
int PPM_IMG::W
```

width of the image in pixels

3.1.2.2 H

```
int PPM_IMG::H
```

height of the image in pixels

3.1.2.3 range

```
int PPM_IMG::range
```

range of dynamic (between black (0) and white (range): usually we will use 255)

3.1.2.4 nbColors

```
int PPM_IMG::nbColors
```

number of color components (either 1 or 3, for grayscale-PGM or color-PPM)

3.1.2.5 hdrSize

```
int PPM_IMG::hdrSize
```

number of bytes in the header part (useful size, not HEADER_SIZE)

3.1.2.6 dataSize

```
int PPM_IMG::dataSize
```

number of bytes in the data part (header+data means the total size of the file)

3.1.2.7 pHeader

```
unsigned char* PPM_IMG::pHeader
```

pointer to the first byte of the header

3.1.2.8 pData

```
unsigned char* PPM_IMG::pData
```

pointer to the first byte of the data part

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

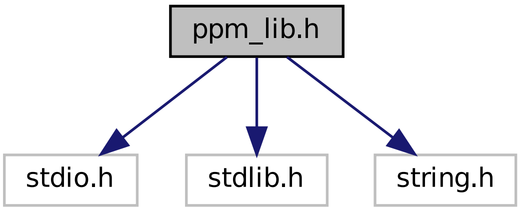
- ppm_lib.h

Chapter 4

File Documentation

4.1 ppm_lib.h File Reference

Include dependency graph for ppm_lib.h:



Classes

- struct PPM_IMG

  Structure for a PPM image.

Macros

- #define **HEADER_SIZE** 1024
- #define **SHIFT_R** 16
- #define **SHIFT_G** 8
- #define **SHIFT_B** 0
- #define red(v) ((v >> SHIFT_R) & 0xFF)
- #define green(v) ((v >> SHIFT_G) & 0xFF)
- #define blue(v) ((v) & 0xFF)
- #define pixel(r, g, b) (((r&0xFF)<<SHIFT_R) | ((g&0xFF)<<SHIFT_G) | ((b&0xFF)))

Functions

- PPM_IMG ∗ ppmNew (int w, int h, int rng, int nbColors)

  This function creates an empty (black image) `PPM_IMG` structure according to the parameters.
- PPM_IMG ∗ ppmOpen (char ∗path)

  This function opens an existing ppm/pgm image from a file on your disk.
- int ppmRead (PPM_IMG ∗img, int x, int y)

  This function reads a pixel from an existing `PPM_IMG` structure pointer.
- void ppmWrite (PPM_IMG ∗img, int x, int y, int value)

  This function writes a pixel into an existing `PPM_IMG` structure pointer.
- void ppmDisplay (PPM_IMG ∗img)

  This function displays the properties of a `PPM_IMG` structure pointer on the standard output.
- void ppmSave (PPM_IMG ∗img, char ∗path)

  This function saves a complete `PPM_IMG` structure into a valid PPM/PGM file.
- void ppmClose (PPM_IMG ∗img)

  This function destroys a `PPM_IMG` structure. This means all the previously allocated memory areas will be freed.
- int ppmGetWidth (PPM_IMG ∗img)

  This function returns the current image width from the `PPM_IMG` structure pointer.
- int ppmGetHeight (PPM_IMG ∗img)

  This function returns the current image height from the `PPM_IMG` structure pointer.
- int ppmGetColors (PPM_IMG ∗img)

  This function returns the number of components of the current image from the `PPM_IMG` structure pointer.
- int ppmGetRange (PPM_IMG ∗img)

  This function returns the current image color range from the `PPM_IMG` structure pointer.

4.1.1 Macro Definition Documentation

4.1.1.1 red

```
#define red(
                v ) ((v >> SHIFT_R) & 0xFF)
```

macro used to get the RED component byte of a pixel value

4.1.1.2 green

```
#define green(
                v ) ((v >> SHIFT_G) & 0xFF)
```

macro used to get the GREEN component byte of a pixel value

4.1.1.3 blue

```
#define blue(
                v ) ((v ) & 0xFF)
```

macro used to get the BLUE component byte of a pixel value

4.1.1.4 pixel

```
#define pixel(
              r,
              g,
              b ) (((r&0xFF)<<SHIFT_R) | ((g&0xFF)<<SHIFT_G) | ((b&0xFF)))
```

macro used to get the pixel value from RED, GREEN and BLUE components (3 x 1-byte)

4.1.2 Function Documentation

4.1.2.1 ppmNew()

```
PPM_IMG* ppmNew (
             int w,
             int h,
             int rng,
             int nbColors )
```

This function creates an empty (black image) PPM_IMG structure according to the parameters.

This function shall be used when you want to create an empty image structure. Once the function has returned the PPM_IMG structure pointer, you are free to use all the other functions to set pixels values, and save it to a file on your disk.

Parameters

w	total width of the new image in pixels.
h	total height of the new image in pixels
rng	range of each pixel component value. Thix value is the maximum allowed value. Usually for 8-bit components, this value will be set to 255.
nbColors	the number of components of your image. Allowed values are either 1 or 3, respectively for gray level images and RGB color images. For gray images, the pixel value will be the less significant byte. For color images, the red, green and blue component values will be stored in the 3 less significant bytes : e.g. `0x00RRGGBB`.

Returns

the pointer to a dynamically allocated PPM structure. This pointer must be used in any call to the ppm_lib functions. The memory free is performed during the call of `ppmClose()`. The returned structure pointer can be used like the one returned by `ppmOpen()` function.

4.1.2.2 ppmOpen()

```
PPM_IMG* ppmOpen (
             char * path )
```

This function opens an existing ppm/pgm image from a file on your disk.

This function shall be used when you want to create an PPM_IMG structure fomr an existing file. The file can be either a valid PPM image (RGB color image) or PGM image (gray level image). The returned structure pointer can be used like the one returned by ppmNew() function.

Parameters

path	Relative path to the image file to open.

Returns

the pointer to a dynamically allocated PPM structure. This pointer must be used in any call to the ppm_lib functions. The memory free is performed during the call of ppmClose().

4.1.2.3 ppmRead()

```
int ppmRead (
             PPM_IMG * img,
             int x,
             int y )
```

This function reads a pixel from an existing PPM_IMG structure pointer.

You just have to give the X and Y coordinates. The origin is in the top left corner of the image. X increasing to the right, Y increasing to the bottom.

Parameters

img	a valid PPM_IMG structure pointer.
x	positive or zero integer value for x coordinate.
y	positive or zero integer value for y coordinate.

Returns

the pixel value stored on a 32-bit value. Check red, green, or blue macros to extract a specific component value.

4.1.2.4 ppmWrite()

```
void ppmWrite (
             PPM_IMG * img,
             int x,
             int y,
             int value )
```

This function writes a pixel into an existing `PPM_IMG` structure pointer.

You just have to give the X and Y coordinates, and the pixel value (32 bits). The origin is in the top left corner of the image. X increasing to the right, Y increasing to the bottom.

Parameters

img	a valid PPM_IMG structure pointer.
x	positive or zero integer value for x coordinate.
y	positive or zero integer value for y coordinate.
value	32-bit pixel value. Look at pixel macro to create a pixel value from the 3 RGB components.

Returns

nothing.

4.1.2.5 ppmDisplay()

```
void ppmDisplay (
            PPM_IMG * img )
```

This function displays the properties of a PPM_IMG structure pointer on the standard output.

Parameters

img	a valid PPM_IMG structure pointer.

Returns

nothing.

4.1.2.6 ppmSave()

```
void ppmSave (
            PPM_IMG * img,
            char * path )
```

This function saves a complete PPM_IMG structure into a valid PPM/PGM file.

Parameters

img	a valid PPM_IMG structure pointer.
path	output PPM/PGM file path

Returns

nothing.

4.1.2.7 ppmClose()

```
void ppmClose (
            PPM_IMG * img )
```

This function destroys a PPM_IMG structure. This means all the previously allocated memory areas will be freed.

Parameters

img	a valid PPM_IMG structure pointer.

Returns

nothing.

4.1.2.8 ppmGetWidth()

```
int ppmGetWidth (
            PPM_IMG * img )
```

This function returns the current image width from the PPM_IMG structure pointer.

Parameters

img	a valid PPM_IMG structure pointer.

Returns

integer value for the image width in pixels.

4.1.2.9 ppmGetHeight()

```
int ppmGetHeight (
            PPM_IMG * img )
```

This function returns the current image height from the PPM_IMG structure pointer.

Parameters

img	a valid PPM_IMG structure pointer.

Returns

integer value for the image height in pixels.

4.1.2.10 ppmGetColors()

```
int ppmGetColors (
            PPM_IMG * img )
```

This function returns the number of components of the current image from the PPM_IMG structure pointer.

Parameters

img	a valid PPM_IMG structure pointer.

Returns

integer value for the number of color components.

4.1.2.11 ppmGetRange()

```
int ppmGetRange (
            PPM_IMG * img )
```

This function returns the current image color range from the PPM_IMG structure pointer.

Parameters

img	a valid PPM_IMG structure pointer.

Returns

integer value for the color component range (maximum value).

Index