

My Project

Generated by Doxygen 1.8.17

1 A build beállításai	1
1.0.1 Változások a specifikációhoz képest	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Filter Struct Reference	7
4.1.1 Detailed Description	7
4.1.2 Member Data Documentation	7
4.1.2.1 filt	7
4.1.2.2 mult	8
4.1.2.3 size_x	8
4.1.2.4 size_y	8
4.2 HSL Struct Reference	8
4.2.1 Detailed Description	9
4.2.2 Member Data Documentation	9
4.2.2.1 h	9
4.2.2.2 l	9
4.2.2.3 s	9
4.3 PPM_Image Struct Reference	10
4.3.1 Detailed Description	10
4.3.2 Member Data Documentation	10
4.3.2.1 image_data	10
4.3.2.2 magic	10
4.3.2.3 maxval	11
4.3.2.4 size_x	11
4.3.2.5 size_y	11
4.4 PsOptions Struct Reference	11
4.4.1 Detailed Description	12
4.4.2 Member Data Documentation	12
4.4.2.1 interval	12
4.4.2.2 interval_max	12
4.4.2.3 interval_min	13
4.4.2.4 merge	13
4.4.2.5 pstype	13
4.4.2.6 threshold	13
4.4.2.7 threshold_bottom_max	14
4.4.2.8 threshold_bottom_min	14
4.4.2.9 threshold_top_max	14

4.4.2.10 threshold_top_min	14
4.5 RGB Struct Reference	15
4.5.1 Detailed Description	15
4.5.2 Member Data Documentation	15
4.5.2.1 b	15
4.5.2.2 g	15
4.5.2.3 r	16
4.6 RGB_SHIFT Struct Reference	16
4.6.1 Detailed Description	16
4.6.2 Member Data Documentation	16
4.6.2.1 blue_x	16
4.6.2.2 blue_y	17
4.6.2.3 green_x	17
4.6.2.4 green_y	17
4.6.2.5 red_x	17
4.6.2.6 red_y	18
4.7 Sort Struct Reference	18
4.7.1 Detailed Description	18
4.7.2 Member Data Documentation	18
4.7.2.1 idx	18
4.7.2.2 value	18
5 File Documentation	19
5.1 mainpage.md File Reference	19
5.2 src/addmath.c File Reference	19
5.2.1 Detailed Description	20
5.2.2 Function Documentation	20
5.2.2.1 clamp()	20
5.2.2.2 max()	20
5.2.2.3 min()	21
5.2.2.4 partition()	21
5.2.2.5 quicksort()	21
5.3 src/addmath.h File Reference	22
5.3.1 Typedef Documentation	23
5.3.1.1 Sort	23
5.3.2 Function Documentation	23
5.3.2.1 clamp()	23
5.3.2.2 max()	24
5.3.2.3 min()	24
5.3.2.4 partition()	24
5.3.2.5 quicksort()	25
5.4 src/imagefunc.c File Reference	26

5.4.1 Detailed Description	28
5.4.2 Function Documentation	28
5.4.2.1 allocatefilter()	28
5.4.2.2 anaglyph3d()	28
5.4.2.3 change_light()	29
5.4.2.4 checkinterval()	30
5.4.2.5 checktreshhold_bottom()	31
5.4.2.6 checktreshhold_top()	31
5.4.2.7 contrast()	32
5.4.2.8 convolve()	32
5.4.2.9 corrupt()	33
5.4.2.10 detect_edges()	34
5.4.2.11 freefilter()	35
5.4.2.12 grayscale()	36
5.4.2.13 hsl2rgb()	36
5.4.2.14 hsl2rgbcolor()	37
5.4.2.15 hue_shift()	37
5.4.2.16 invert()	38
5.4.2.17 mirror_diagonal()	38
5.4.2.18 mirror_horizontal()	39
5.4.2.19 mirror_vertical()	39
5.4.2.20 pixelsort()	40
5.4.2.21 rgb2hsl()	41
5.4.2.22 rgb_shift()	42
5.4.2.23 rotate()	43
5.4.2.24 rotate_vertical()	43
5.4.2.25 set_white()	44
5.4.2.26 setfilter()	44
5.4.2.27 sharp_grayscale()	45
5.4.2.28 sinecolor_shift()	46
5.4.2.29 sortcopy()	47
5.5 src/imagefunc.h File Reference	47
5.5.1 Typedef Documentation	50
5.5.1.1 Filter	50
5.5.1.2 HSL	50
5.5.1.3 mirror_type	51
5.5.1.4 pixelsort_preset	51
5.5.1.5 ps_option_type	51
5.5.1.6 ps_type	51
5.5.1.7 PsOptions	51
5.5.1.8 RGB	51
5.5.1.9 RGB_SHIFT	52

5.5.2 Enumeration Type Documentation	52
5.5.2.1 mirror_type	52
5.5.2.2 pixelsort_preset	52
5.5.2.3 ps_option_type	53
5.5.2.4 ps_type	53
5.5.3 Function Documentation	53
5.5.3.1 allocatfilter()	53
5.5.3.2 anaglyph3d()	54
5.5.3.3 change_light()	55
5.5.3.4 contrast()	55
5.5.3.5 convolve()	56
5.5.3.6 corrupt()	57
5.5.3.7 detect_edges()	58
5.5.3.8 freefilter()	59
5.5.3.9 grayscale()	60
5.5.3.10 hsl2rgb()	60
5.5.3.11 hsl2rgbcolor()	61
5.5.3.12 hue_shift()	61
5.5.3.13 invert()	62
5.5.3.14 mirror_diagonal()	62
5.5.3.15 mirror_horizontal()	63
5.5.3.16 mirror_vertical()	63
5.5.3.17 pixelsort()	64
5.5.3.18 rgb2hsl()	65
5.5.3.19 rgb_shift()	66
5.5.3.20 set_black()	67
5.5.3.21 setfilter()	67
5.5.3.22 sharp_grayscale()	68
5.5.3.23 sinecolor_shift()	68
5.5.3.24 sortcopy()	69
5.6 src/main.c File Reference	70
5.6.1 Detailed Description	71
5.6.2 Function Documentation	71
5.6.2.1 main()	71
5.7 src/ppm.c File Reference	72
5.7.1 Detailed Description	73
5.7.2 Function Documentation	73
5.7.2.1 allocateimage()	74
5.7.2.2 checktype()	74
5.7.2.3 freeimage()	75
5.7.2.4 parsepixels()	75
5.7.2.5 parseword()	76

5.7.2.6 PPM_Parser()	77
5.7.2.7 PPM_Writer()	78
5.8 src/ppm.h File Reference	78
5.8.1 Typedef Documentation	80
5.8.1.1 PPM_Image	80
5.8.2 Function Documentation	80
5.8.2.1 allocateimage()	80
5.8.2.2 allocateimage1d()	81
5.8.2.3 checktype()	81
5.8.2.4 freeimage()	82
5.8.2.5 getpixelcolor()	82
5.8.2.6 parsepixels()	83
5.8.2.7 parseword()	83
5.8.2.8 PPM_Parser()	84
5.8.2.9 PPM_Writer()	85
5.8.2.10 setpixelcolor()	85
Index	87

Chapter 1

A build beállításai

Az builder Meson 0.53.2 és a kódja a következő:

```
nhf_c_sources = [  
    'main.c',  
    'ppm.c',  
    'addmath.c',  
    'imagefunc.c',  
]  
nhf_c_deps = [  
    dependency('glib-2.0'),  
    meson.get_compiler('c').find_library('m', required: false)  
]  
executable('imageproc', nhf_c_sources,  
    dependencies: nhf_c_deps,  
    install: true,  
)
```

compiler: gcc 9.3.0

linker: ld.bfd 2.34

A projektet a Gnome-Builder-ben készítettem, Meson kódja a következő:

```
project('imageproc', 'c',  
    version: '0.1.0',  
    meson_version: '>= 0.50.0',  
    default_options: [ 'warning_level=2',  
        'c_std=gnull',  
    ],  
)  
config_h = configuration_data()  
config_h.set_quoted('PACKAGE_VERSION', meson.project_version())  
configure_file(  
    output: 'nhf_c-config.h',  
    configuration: config_h,  
)  
add_project_arguments([  
    '-I' + meson.build_root(),  
], language: 'c')  
subdir('src')
```

1.0.1 Változások a specifikációhoz képest

Új funkciók:

- + sinecolor-shift
- + pixelsort/edges
- + 3d
- + edge-detect

Megváltozott a funkciók végrehajtásának prioritása:

- h --help
- i --input
- o --output
- lightness

```
--contrast  
--hue-shift  
--invert  
--sinecolor-shift  
--mirror  
--rgb-shift  
--pixelsort  
--blur  
--sharpen  
--corrupt  
--grayscale  
--3d  
--edge-detect
```

Chapter 2

Class Index

2.1 Class List

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

Filter	A konvolúcióhoz használt filter tulajdonságai	7
HSL	HSL színskála struktúrája	8
PPM_Image	PPM fájl tárolására használt struktúra	10
PsOptions	A pixelsort beállításai	11
RGB	RGB színskála struktúrája	15
RGB_SHIFT	Az RGB shífteléshez használt értékek	16
Sort	18

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

src/ addmath.c		
	Kiegészítő matematikai függvények	19
src/ addmath.h	22
src/ imagefunc.c		
	A kép módosításához használt függvények	26
src/ imagefunc.h	47
src/ main.c		
	Fő fájl, ami a parancssort kezeli	70
src/ ppm.c		
	PPM fájl beolvasása	72
src/ ppm.h	78

Chapter 4

Class Documentation

4.1 Filter Struct Reference

A konvolúcióhoz használt filter tulajdonságai.

```
#include <imagefunc.h>
```

Public Attributes

- int ** [filt](#)
- int [size_x](#)
- int [size_y](#)
- double [mult](#)

4.1.1 Detailed Description

A konvolúcióhoz használt filter tulajdonságai.

Definition at line 27 of file imagefunc.h.

4.1.2 Member Data Documentation

4.1.2.1 `filt`

```
int** Filter::filt
```

2 dimenziós tömb a filter értékeivel

Definition at line 28 of file imagefunc.h.

Referenced by `convolve()`, `freefilter()`, and `setfilter()`.

4.1.2.2 mult

```
double Filter::mult
```

[Filter](#) értékeinek szorzásához használt konstans

Definition at line 31 of file imagefunc.h.

Referenced by `convolve()`, and `setfilter()`.

4.1.2.3 size_x

```
int Filter::size_x
```

[Filter](#) oszlopainak száma

Definition at line 29 of file imagefunc.h.

Referenced by `convolve()`, and `setfilter()`.

4.1.2.4 size_y

```
int Filter::size_y
```

[Filter](#) sorainak száma

Definition at line 30 of file imagefunc.h.

Referenced by `convolve()`, `freefilter()`, and `setfilter()`.

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

- [src/imagefunc.h](#)

4.2 HSL Struct Reference

[HSL](#) színskála struktúrája.

```
#include <imagefunc.h>
```

Public Attributes

- double [h](#)
- double [s](#)
- double [l](#)

4.2.1 Detailed Description

[HSL](#) színskála struktúrája.

Definition at line 9 of file imagefunc.h.

4.2.2 Member Data Documentation

4.2.2.1 h

```
double HSL::h
```

[HSL](#) Hue

Definition at line 10 of file imagefunc.h.

Referenced by [hsl2rgb\(\)](#), [hue_shift\(\)](#), and [rgb2hsl\(\)](#).

4.2.2.2 l

```
double HSL::l
```

[HSL](#) Lightness

Definition at line 12 of file imagefunc.h.

Referenced by [change_light\(\)](#), [hsl2rgb\(\)](#), [pixelsort\(\)](#), and [rgb2hsl\(\)](#).

4.2.2.3 s

```
double HSL::s
```

[HSL](#) Saturation

Definition at line 11 of file imagefunc.h.

Referenced by [hsl2rgb\(\)](#), and [rgb2hsl\(\)](#).

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

- [src/imagefunc.h](#)

4.3 PPM_Image Struct Reference

a PPM fájl tárolására használt struktúra

```
#include <ppm.h>
```

Public Attributes

- unsigned char *** [image_data](#)
- int [size_x](#)
- int [size_y](#)
- char [magic](#) [2+1]
- int [maxval](#)

4.3.1 Detailed Description

a PPM fájl tárolására használt struktúra

Definition at line 9 of file ppm.h.

4.3.2 Member Data Documentation

4.3.2.1 image_data

```
unsigned char*** PPM_Image::image_data
```

3 dimenziós tömb amiben a kép pixeleinek az értékeit tároljuk

Definition at line 10 of file ppm.h.

Referenced by [anaglyph3d\(\)](#), [corrupt\(\)](#), [main\(\)](#), [parsepixels\(\)](#), [parseword\(\)](#), [PPM_Writer\(\)](#), [rgb_shift\(\)](#), and [rotate_↵vertical\(\)](#).

4.3.2.2 magic

```
char PPM_Image::magic[2+1]
```

a kép két karakterből álló magic-je

Definition at line 13 of file ppm.h.

Referenced by [parseword\(\)](#), [PPM_Parser\(\)](#), and [PPM_Writer\(\)](#).

4.3.2.3 maxval

```
int PPM_Image::maxval
```

a kép maxval-ja

Definition at line 14 of file ppm.h.

Referenced by `parsepixels()`, `parseword()`, `PPM_Parser()`, and `PPM_Writer()`.

4.3.2.4 size_x

```
int PPM_Image::size_x
```

a kép oszlopainak száma

Definition at line 11 of file ppm.h.

Referenced by `anaglyph3d()`, `corrupt()`, `main()`, `parsepixels()`, `parseword()`, `PPM_Parser()`, `PPM_Writer()`, and `rgb↔_shift()`.

4.3.2.5 size_y

```
int PPM_Image::size_y
```

a kép sorainak száma

Definition at line 12 of file ppm.h.

Referenced by `anaglyph3d()`, `corrupt()`, `main()`, `parsepixels()`, `parseword()`, `PPM_Parser()`, `PPM_Writer()`, `rgb↔_shift()`, and `rotate_vertical()`.

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

- [src/ppm.h](#)

4.4 PsOptions Struct Reference

A pixelsort beállításai.

```
#include <imagefunc.h>
```

Public Attributes

- `ps_type` `pstype`
- `ps_option_type` `threshold`
- `int` `threshold_bottom_min`
- `int` `threshold_bottom_max`
- `int` `threshold_top_min`
- `int` `threshold_top_max`
- `ps_option_type` `interval`
- `int` `interval_min`
- `int` `interval_max`
- `double` `merge`

4.4.1 Detailed Description

A pixelsort beállításai.

Definition at line 92 of file `imagefunc.h`.

4.4.2 Member Data Documentation

4.4.2.1 `interval`

`ps_option_type` `PsOptions::interval`

a rendezési környezet kiválasztásának típusa. Ha `ran` akkor alsó és felső korlát között választ egy véletlenszerű értéket. Ha `man` akkor csak az alsó korlát számít.

Definition at line 99 of file `imagefunc.h`.

Referenced by `checkinterval()`, and `main()`.

4.4.2.2 `interval_max`

`int` `PsOptions::interval_max`

a rendezési környezet felső határa

Definition at line 101 of file `imagefunc.h`.

Referenced by `checkinterval()`, and `main()`.

4.4.2.3 interval_min

```
int PsOptions::interval_min
```

a rendezési környezet alsó határa. Az értékét úgy érdemes megválasztani, hogy igazodjon a képen található objektumok méretéhez. Egy tájképnél például lehet nagy értéket választani, mivel ott nem fontosak a részletek, míg egy részletes képnél minnél kisebbre kell választani, hogy minden felismerhető legyen.

Definition at line 100 of file imagefunc.h.

Referenced by `checkinterval()`, and `main()`.

4.4.2.4 merge

```
double PsOptions::merge
```

minél kisebb a merge mérete annál kisebbet ugrik a ciklus, ennek megfelelően annál nagyobb lesz az átfedés a környezetek között. Ha ez 0, az azt jelenti hogy minden pixelt megvizsgál, így kellően nagy treshold tartományban majdnem minden pixel bekerül és a kép el fog csúszni a rendezés irányának megfelelően, mivel a legvilágosabb pixelek a kép szélére sodródnak. Ez azt is jelenti, hogy sokkal lassabb lesz a program (1080x1080-as képen akár 500 ezer - 1 millió rendezést is el kell végezni.).

Definition at line 102 of file imagefunc.h.

Referenced by `main()`, and `pixelsort()`.

4.4.2.5 pstype

```
ps_type PsOptions::pstype
```

A pixelsort típusa

Definition at line 93 of file imagefunc.h.

Referenced by `main()`, and `pixelsort()`.

4.4.2.6 threshold

```
ps_option_type PsOptions::threshold
```

A treshold kiválasztásának típusa. Ha ran akkor alsó és felső korlát között választ egy véletlenszerű értéket. Ha man akkor csak az alsó korlát számít.

Definition at line 94 of file imagefunc.h.

Referenced by `checkthreshold_bottom()`, `checkthreshold_top()`, and `main()`.

4.4.2.7 `treshold_bottom_max`

```
int PsOptions::treshold_bottom_max
```

Az alsó treshold felső határa.

Definition at line 96 of file `imagefunc.h`.

Referenced by `checktreshold_bottom()`, and `main()`.

4.4.2.8 `treshold_bottom_min`

```
int PsOptions::treshold_bottom_min
```

Az alsó treshold alsó határa.

Definition at line 95 of file `imagefunc.h`.

Referenced by `checktreshold_bottom()`, and `main()`.

4.4.2.9 `treshold_top_max`

```
int PsOptions::treshold_top_max
```

Az felső treshold felső határa.

Definition at line 98 of file `imagefunc.h`.

Referenced by `checktreshold_top()`, and `main()`.

4.4.2.10 `treshold_top_min`

```
int PsOptions::treshold_top_min
```

Az felső treshold alsó határa.

Definition at line 97 of file `imagefunc.h`.

Referenced by `checktreshold_top()`, and `main()`.

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

- [src/imagefunc.h](#)

4.5 RGB Struct Reference

RGB színskála struktúrája.

```
#include <imagefunc.h>
```

Public Attributes

- unsigned char [r](#)
- unsigned char [g](#)
- unsigned char [b](#)

4.5.1 Detailed Description

RGB színskála struktúrája.

Definition at line 18 of file imagefunc.h.

4.5.2 Member Data Documentation

4.5.2.1 [b](#)

```
unsigned char RGB::b
```

[RGB](#) Blue

Definition at line 21 of file imagefunc.h.

Referenced by [change_light\(\)](#), [hsl2rgb\(\)](#), and [hue_shift\(\)](#).

4.5.2.2 [g](#)

```
unsigned char RGB::g
```

[RGB](#) Green

Definition at line 20 of file imagefunc.h.

Referenced by [change_light\(\)](#), [hsl2rgb\(\)](#), and [hue_shift\(\)](#).

4.5.2.3 r

```
unsigned char RGB::r
```

RGB Red

Definition at line 19 of file imagefunc.h.

Referenced by `change_light()`, `hsl2rgb()`, and `hue_shift()`.

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

- `src/imagefunc.h`

4.6 RGB_SHIFT Struct Reference

Az RGB shifteléshez használt értékek.

```
#include <imagefunc.h>
```

Public Attributes

- int `red_x`
- int `red_y`
- int `green_x`
- int `green_y`
- int `blue_x`
- int `blue_y`

4.6.1 Detailed Description

Az RGB shifteléshez használt értékek.

Definition at line 37 of file imagefunc.h.

4.6.2 Member Data Documentation

4.6.2.1 blue_x

```
int RGB_SHIFT::blue_x
```

Kék szín eltolásának x értéke

Definition at line 42 of file imagefunc.h.

Referenced by `rgb_shift()`.

4.6.2.2 blue_y

```
int RGB_SHIFT::blue_y
```

Kék szín eltolásának y értéke

Definition at line 43 of file imagefunc.h.

Referenced by rgb_shift().

4.6.2.3 green_x

```
int RGB_SHIFT::green_x
```

Zöld szín eltolásának x értéke

Definition at line 40 of file imagefunc.h.

Referenced by rgb_shift().

4.6.2.4 green_y

```
int RGB_SHIFT::green_y
```

Zöld szín eltolásának y értéke

Definition at line 41 of file imagefunc.h.

Referenced by rgb_shift().

4.6.2.5 red_x

```
int RGB_SHIFT::red_x
```

Vörös szín eltolásának x értéke

Definition at line 38 of file imagefunc.h.

Referenced by rgb_shift().

4.6.2.6 red_y

```
int RGB_SHIFT::red_y
```

Vörös szín eltolásának y értéke

Definition at line 39 of file imagefunc.h.

Referenced by `rgb_shift()`.

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

- [src/imagefunc.h](#)

4.7 Sort Struct Reference

```
#include <addmath.h>
```

Public Attributes

- double [value](#)
- int [idx](#)

4.7.1 Detailed Description

Definition at line 4 of file addmath.h.

4.7.2 Member Data Documentation

4.7.2.1 idx

```
int Sort::idx
```

az értékhez tartozó index az eredeti tömbben

Definition at line 6 of file addmath.h.

Referenced by `pixelsort()`, and `sortcopy()`.

4.7.2.2 value

```
double Sort::value
```

a rendezés alapjául szolgáló érték

Definition at line 5 of file addmath.h.

Referenced by `partition()`, and `pixelsort()`.

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

- [src/addmath.h](#)

Chapter 5

File Documentation

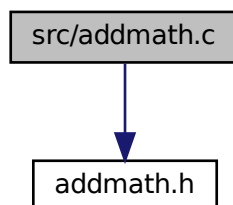
5.1 mainpage.md File Reference

5.2 src/addmath.c File Reference

Kiegészítő matematikai függvények.

```
#include "addmath.h"
```

Include dependency graph for addmath.c:



Functions

- double **min** (double list[], int size)
: egy double tömb legkisebb elemét veszi
- double **max** (double list[], int size)
: egy double tömb legnagyobb elemét veszi
- double **clamp** (double value, double minval, double maxval)
: minimum és maximum értékek közé szorítja a megadott számot
- int **partition** (Sort *list, int lo, int hi)
A quicksort segédfüggvénye.
- void **quicksort** (Sort *list, int lo, int hi)
Quicksort alapú rendezés.

5.2.1 Detailed Description

Kiegészítő matematikai függvények.

5.2.2 Function Documentation

5.2.2.1 clamp()

```
double clamp (  
    double value,  
    double minval,  
    double maxval )
```

: minimum és maximum értékek közé szorítja a megadott számot

Parameters

in	<i>value</i>	a szám amit korlátozni akarunk
in	<i>minval</i>	minimum
in	<i>maxval</i>	maximum

Definition at line 43 of file addmath.c.

Referenced by `change_light()`, `contrast()`, `convolve()`, `corrupt()`, and `sinecolor_shift()`.

5.2.2.2 max()

```
double max (  
    double list[],  
    int size )
```

: egy double tömb legnagyobb elemét veszi

Parameters

in	<i>list</i>	a tömb amiben a legnagyobbat keressük
in	<i>size</i>	a tömb mérete
out	<i>maximum</i>	a legnagyobb érték

Definition at line 28 of file addmath.c.

Referenced by `pixelsort()`, and `rgb2hsl()`.

5.2.2.3 min()

```
double min (
    double list[],
    int size )
```

: egy double tömb legkisebb elemét veszi

Parameters

in	<i>list</i>	a tömb amiben a legkisebbet keressük
in	<i>size</i>	a tömb mérete
out	<i>minimum</i>	a legkisebb érték

Definition at line 13 of file addmath.c.

Referenced by rgb2hsl().

5.2.2.4 partition()

```
int partition (
    Sort * list,
    int lo,
    int hi )
```

A quicksort segédfüggvénye.

Parameters

in	<i>*list</i>	a rendezni kívánt tömb
in	<i>lo</i>	a legkisebb index
in	<i>hi</i>	a legnagyobb index

Definition at line 57 of file addmath.c.

References Sort::value.

Referenced by quicksort().

5.2.2.5 quicksort()

```
void quicksort (
    Sort * list,
    int lo,
    int hi )
```

Quicksort alapú rendezés.

Parameters

in	<i>*list</i>	a rendezni kívánt tömb
in	<i>lo</i>	a legkisebb index
in	<i>hi</i>	a legnagyobb index

See also

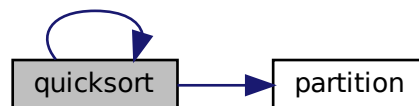
pszeudokód itt: https://en.wikipedia.org/wiki/Quicksort#Lomuto_partition_scheme
[partition](#)

Definition at line 82 of file addmath.c.

References `partition()`, and `quicksort()`.

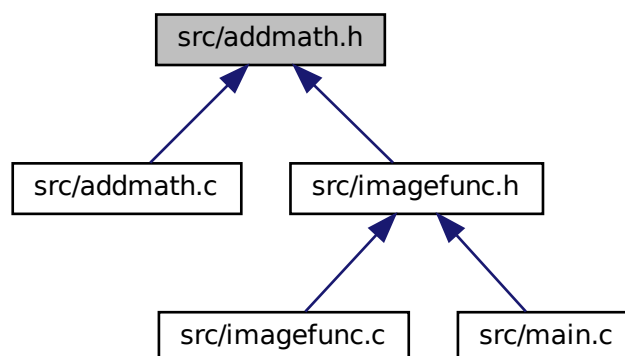
Referenced by `quicksort()`, and `sortcopy()`.

Here is the call graph for this function:



5.3 src/addmath.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct [Sort](#)

Typedefs

- typedef struct [Sort](#) [Sort](#)

Functions

- double [min](#) (double list[], int size)
: egy double tömb legkisebb elemét veszi
- double [max](#) (double list[], int size)
: egy double tömb legnagyobb elemét veszi
- double [clamp](#) (double value, double minval, double maxval)
: minimum és maximum értékek közé szorítja a megadott számot
- int [partition](#) ([Sort](#) *list, int lo, int hi)
A quicksort segédfüggvénye.
- void [quicksort](#) ([Sort](#) *list, int lo, int hi)
Quicksort alapú rendezés.

5.3.1 Typedef Documentation

5.3.1.1 Sort

```
typedef struct Sort Sort
```

5.3.2 Function Documentation

5.3.2.1 clamp()

```
double clamp (
    double value,
    double minval,
    double maxval )
```

: minimum és maximum értékek közé szorítja a megadott számot

Parameters

in	<i>value</i>	a szám amit korlátozni akarunk
in	<i>minval</i>	minimum
in	<i>maxval</i>	maximum

Definition at line 43 of file addmath.c.

Referenced by `change_light()`, `contrast()`, `convolve()`, `corrupt()`, and `sinecolor_shift()`.

5.3.2.2 `max()`

```
double max (
    double list[],
    int size )
```

: egy double tömb legnagyobb elemét veszi

Parameters

in	<i>list</i>	a tömb amiben a legnagyobbat keressük
in	<i>size</i>	a tömb mérete
out	<i>maximum</i>	a legnagyobb érték

Definition at line 28 of file addmath.c.

Referenced by `pixelsort()`, and `rgb2hsl()`.

5.3.2.3 `min()`

```
double min (
    double list[],
    int size )
```

: egy double tömb legkisebb elemét veszi

Parameters

in	<i>list</i>	a tömb amiben a legkisebbet keressük
in	<i>size</i>	a tömb mérete
out	<i>minimum</i>	a legkisebb érték

Definition at line 13 of file addmath.c.

Referenced by `rgb2hsl()`.

5.3.2.4 `partition()`

```
int partition (
    Sort * list,
```



```
int lo,  
int hi )
```

A quicksort segédfüggvénye.

Parameters

in	<i>*list</i>	a rendezni kívánt tömb
in	<i>lo</i>	a legkisebb index
in	<i>hi</i>	a legnagyobb index

Definition at line 57 of file addmath.c.

References Sort::value.

Referenced by quicksort().

5.3.2.5 quicksort()

```
void quicksort (  
    Sort * list,  
    int lo,  
    int hi )
```

Quicksort alapú rendezés.

Parameters

in	<i>*list</i>	a rendezni kívánt tömb
in	<i>lo</i>	a legkisebb index
in	<i>hi</i>	a legnagyobb index

See also

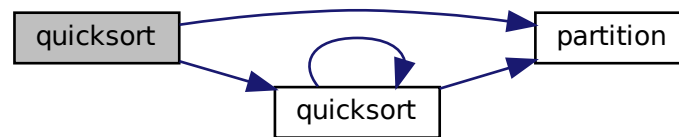
pszeudokód itt: [https://en.wikipedia.org/wiki/Quicksort#Lomuto_partition_↵
scheme](https://en.wikipedia.org/wiki/Quicksort#Lomuto_partition_scheme)
[partition](#)

Definition at line 82 of file addmath.c.

References partition(), and quicksort().

Referenced by quicksort(), and sortcopy().

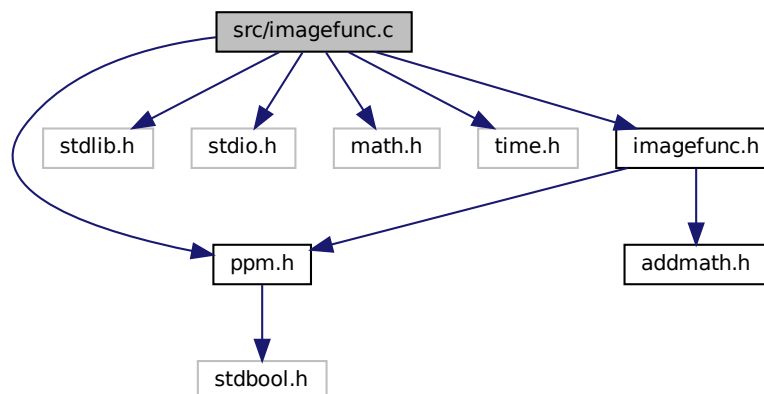
Here is the call graph for this function:



5.4 src/imagefunc.c File Reference

A kép módosításához használt függvények.

```
#include "ppm.h"
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>
#include "imagefunc.h"
Include dependency graph for imagefunc.c:
```



Functions

- void [setfilter](#) ([Filter](#) *filter, int *filt, double mult, int size_x, int size_y)
beállítja egy filter értékét amit a convolve használ
- void [freefilter](#) ([Filter](#) filter)
felszabadítja a filtert amit a convolve használ
- int ** [allocatefilter](#) (int size_x, int size_y)

- lefolgalja a [Filter](#) 2 dimenziós tömbjét
- [HSL rgb2hsl](#) (unsigned char pixel[])
 - [RGB](#) színskála átalakítása [HSL](#) skálára.
- double [hsl2rgbcolor](#) (double temp1, double temp2, double temp3)
- [RGB hsl2rgb](#) (HSL pixel)
 - [HSL](#) színskála átalakítása [RGB](#) skálára.
- void [invert](#) (unsigned char pixel[])
 - A pixel értékének invertálása.
- void [contrast](#) (unsigned char pixel[], double level)
 - Megváltoztatja a pixel kontrasztját.
- void [grayscale](#) (unsigned char pixel[])
 - fekete-fehérré alakít egy képet
- void [sharp_grayscale](#) (unsigned char pixel[])
 - a pixel intenzitása (a három szín átlaga) alapján a pixel értékéhez legközelebbi szélsőértékhez igazítja a pixel értéket (128 alatt 0, 128 felett 255)
- void [set_white](#) (unsigned char pixel[], int threshold)
 - a threshold feletti értéket fehérre változtatja, az alatta lévőket feketévé
- unsigned char *** [detect_edges](#) (unsigned char ***image, int size_x, int size_y)
 - megkeresi a képen található objektumok függőleges széleit
- void [change_light](#) (unsigned char pixel[], int percent)
 - egy pixel fényességének megváltoztatása
- void [mirror_diagonal](#) (unsigned char ***matrix, int size_x, int size_y)
 - az átló mentén tükrözi a képet
- void [mirror_vertical](#) (unsigned char ***matrix, int size_x, int size_y)
 - a függőleges tengelyre tükrözi a képet
- void [mirror_horizontal](#) (unsigned char ***matrix, int size_x, int size_y)
 - a vízszintes tengelyre tükrözi a képet
- void [convolve](#) (unsigned char ***original, int size_x, int size_y, [Filter](#) filter, int times)
 - végrehajtja a konvolúciót, ami a blur és sharpen lépésekhez kell
- void [sortcopy](#) (Sort partline[], unsigned char ***image, int line, int start, int end, int dir)
- static void [checkthreshold_top](#) (int *threshold, [PsOptions](#) options)
 - A pixelsort segédfüggvénye. A cím szerint megadott *threshold paraméterbe állítja be a felső thresholdot az alapján hogy a pixelsort milyen paramétereket kapott.
- static void [checkthreshold_bottom](#) (int *threshold, [PsOptions](#) options)
 - A pixelsort segédfüggvénye. A cím szerint megadott *threshold paraméterbe állítja be az alsó thresholdot az alapján hogy a pixelsort milyen paramétereket kapott.
- static void [checkinterval](#) (int *interval, [PsOptions](#) options, int elem)
 - A pixelsort segédfüggvénye. A cím szerint megadott *interval paraméterbe állítja be, hogy mekkora legyen az a környezet amin el kell végezni a rendezést, az alapján hogy a pixelsort milyen paramétereket kapott.
- void [pixelsort](#) (unsigned char ***image, int size_x, int size_y, [PsOptions](#) options)
 - Végrehajtja a pixelsort-ot.
- void [hue_shift](#) (unsigned char pixel[], double value)
 - A [HSL](#) színskála Hue értékét tolja el.
- void [sinecolor_shift](#) (unsigned char pixel[], double amplifier, double freq, double phase, double bias)
 - A szinusz függvény alapján eltolja a színeket, [RGB](#) színcsatornánként.
- static void [rotate](#) (unsigned char **line, int size, int color, int rot)
 - Megforgatja a paraméterként kapott sort.
- static void [rotate_vertical](#) ([PPM_Image](#) *image, int col, int color, int rot)
 - Oszlopokat forgat meg.
- void [rgb_shift](#) ([PPM_Image](#) *image, [RGB_SHIFT](#) options)
 - Elmozdítja az [RGB](#) színskála értékeit a megadott irányba.
- void [anaglyph3d](#) ([PPM_Image](#) *image)

Vörös-Cián 3D képpé alakítja a bemenetet.

- void `corrupt` (`PPM_Image` *image)

Véletlenszerűen tönkreteszi a képet.

5.4.1 Detailed Description

A kép módosításához használt függvények.

5.4.2 Function Documentation

5.4.2.1 `allocatefilter()`

```
int** allocatefilter (
    int size_x,
    int size_y )
```

lefoglalja a `Filter` 2 dimenziós tömbjét

Parameters

in	<code>size_x</code>	a filter oszlopainak száma
in	<code>size_y</code>	a filter sorainak száma

Definition at line 51 of file `imagefunc.c`.

Referenced by `setfilter()`.

5.4.2.2 `anaglyph3d()`

```
void anaglyph3d (
    PPM_Image * image )
```

Vörös-Cián 3D képpé alakítja a bemenetet.

Parameters

in	*image	átalakítandó kép (<code>PPM_Image</code>)
----	--------	---

A lényege, hogy a vörös csatornát el kell tolni balra valamennyivel. Ez az érték a szélesség nagyjából 0,925%-a (1080 pixel szélességnél kb. 10 pixel). A kép úgy fog kinézni, mintha be lenne süllyesztve.

See also

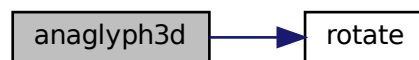
[PPM_Image](#)
[rotate](#)

Definition at line 794 of file imagefunc.c.

References [PPM_Image::image_data](#), [rotate\(\)](#), [PPM_Image::size_x](#), and [PPM_Image::size_y](#).

Referenced by [main\(\)](#).

Here is the call graph for this function:



5.4.2.3 change_light()

```
void change_light (
    unsigned char pixel[],
    int percent )
```

egy pixel fényességének megváltoztatása

Parameters

in	<i>pixel</i> []	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
in	<i>percent</i>	mennyi százalékkal

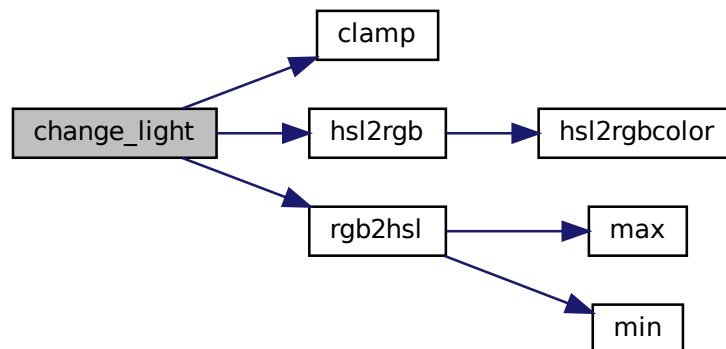
A pixel értéket először [HSL](#) színskálára alakítjuk át, majd ehhez hozzáadjuk a változtatás mértékét, végül korlátozzuk 0, 1 intervallumra és visszaalakítjuk [RGB](#) színskálára.

Definition at line 298 of file imagefunc.c.

References [RGB::b](#), [clamp\(\)](#), [RGB::g](#), [hsl2rgb\(\)](#), [HSL::l](#), [RGB::r](#), and [rgb2hsl\(\)](#).

Referenced by [corrupt\(\)](#), and [main\(\)](#).

Here is the call graph for this function:



5.4.2.4 checkinterval()

```
static void checkinterval (
    int * interval,
    PsOptions options,
    int elem ) [static]
```

A pixelsort segédfüggvénye. A cím szerint megadott *interval* paraméterbe állítja be, hogy mekkora legyen az a környezet amin el kell végezni a rendezést, az alapján hogy a pixelsort milyen paraméterket kapott.

Parameters

in	<i>*interval</i>	a paraméter amibe vissza kell írni az értéket
in	<i>options</i>	a pixelsort által kapott tulajdonságok
in	<i>elem</i>	az éppen aktuálisan vizsgált pixel oszlopszáma

See also

[PsOptions](#)
[ps_option_type](#)

Ha a `ps_option_type` ran akkor egy véletlenszerű értéket választunk a felső és alsó korlát között. Ha man akkor csak az alsó korlátot vesszük figyelembe.

Definition at line 518 of file `imagefunc.c`.

References `PsOptions::interval`, `PsOptions::interval_max`, `PsOptions::interval_min`, and `ran`.

Referenced by `pixelsort()`.

5.4.2.5 checktreshold_bottom()

```
static void checktreshold_bottom (
    int * treshold,
    PsOptions options ) [static]
```

A pixelsort segédfüggvénye. A cím szerint megadott **treshold* paraméterbe állítja be az alsó tresholdot az alapján hogy a pixelsort milyen paraméterket kapott.

Parameters

in	<i>*treshold</i>	a paraméter amibe vissza kell írni az értéket
in	<i>options</i>	a pixelsort által kapott tulajdonságok

See also

[PsOptions](#)

[ps_option_type](#)

Ha a *ps_option_type* ran akkor egy véletlenszerű értéket választunk a felső és alsó korlát között. Ha man akkor csak az alsó korlátot vesszük figyelembe.

Definition at line 501 of file imagefunc.c.

References [ran](#), [PsOptions::treshold](#), [PsOptions::treshold_bottom_max](#), and [PsOptions::treshold_bottom_min](#).

Referenced by [pixelsort\(\)](#).

5.4.2.6 checktreshold_top()

```
static void checktreshold_top (
    int * treshold,
    PsOptions options ) [static]
```

A pixelsort segédfüggvénye. A cím szerint megadott **treshold* paraméterbe állítja be a felső tresholdot az alapján hogy a pixelsort milyen paraméterket kapott.

Parameters

in	<i>*treshold</i>	a paraméter amibe vissza kell írni az értéket
in	<i>options</i>	a pixelsort által kapott tulajdonságok

See also

[PsOptions](#)

[ps_option_type](#)

Ha a *ps_option_type* ran akkor egy véletlenszerű értéket választunk a felső és alsó korlát között. Ha man akkor csak az alsó korlátot vesszük figyelembe.

Definition at line 485 of file imagefunc.c.

References `ran`, `PsOptions::treshold`, `PsOptions::treshold_top_max`, and `PsOptions::treshold_top_min`.

Referenced by `pixelsort()`.

5.4.2.7 contrast()

```
void contrast (
    unsigned char pixel[],
    double level )
```

Megváltoztatja a pixel kontrasztját.

Parameters

in	<i>pixel</i> []	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
in	<i>level</i>	a változtatás mértéke (a 259 nem megengedett)

See also

pszeudokód innen: <https://www.dfstudios.co.uk/articles/programming/image-programming-al>

Definition at line 191 of file imagefunc.c.

References `clamp()`.

Referenced by `corrupt()`, and `main()`.

Here is the call graph for this function:



5.4.2.8 convolve()

```
void convolve (
    unsigned char *** original,
    int size_x,
    int size_y,
    Filter filter,
    int times )
```

végrehajtja a konvolúciót, ami a blur és sharpen lépésekhez kell

See also

pszeudokód és működési elv itt: [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))

[Filter](#)

Parameters

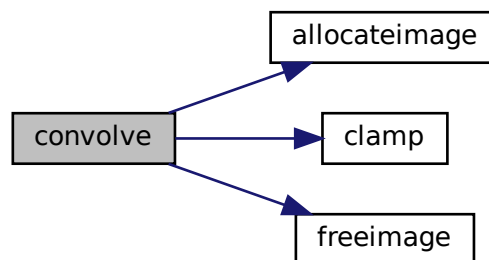
in	*** <i>original</i>	módosítandó kép
in	<i>size_x</i>	a kép oszlopainak száma
in	<i>size_y</i>	a kép sorainak száma
in	<i>filter</i>	a használandó filter

Definition at line 375 of file imagefunc.c.

References `allocateimage()`, `clamp()`, `Filter::filt`, `freeimage()`, `Filter::mult`, `Filter::size_x`, and `Filter::size_y`.

Referenced by `detect_edges()`, and `main()`.

Here is the call graph for this function:



5.4.2.9 corrupt()

```
void corrupt (
    PPM_Image * image )
```

Véletlenszerűen tönkreteszi a képet.

Parameters

in	*** <i>image</i>	a módosítandó kép
----	------------------	-------------------

Először kiválasztunk 3 értéket amivel sorrendben a: fényességet (`change_light`), a kontrasztot (`contrast`) és a [HSL](#)

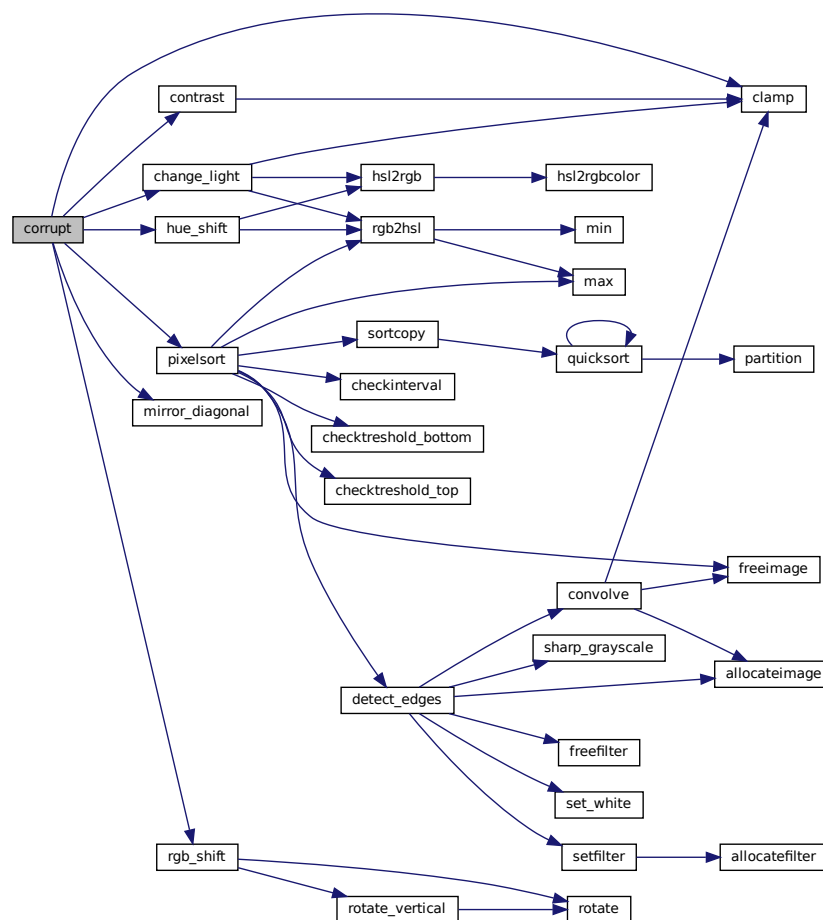
Hue -t (hue_shift) módosítjuk. Ezek után a kép bizonyos részeit tükrözzük, ha páratlan számú alkalommal, akkor a kép egy része fordítva lesz, ha páros számú alkalommal, akkor az eredeti irányban. Ez után az RGB színeit kell shiftelni véletlenszerű irányba és értékkel. Ezután a lehető legbővebb véletlenszerű beállítással végrehajtjuk a pixelsort -ot

Definition at line 810 of file imagefunc.c.

References change_light(), clamp(), contrast(), hsl_l, hue_shift(), PPM_Image::image_data, mirror_diagonal(), pixelsort(), ran, rgb_shift(), PPM_Image::size_x, and PPM_Image::size_y.

Referenced by main().

Here is the call graph for this function:



5.4.2.10 detect_edges()

```

unsigned char*** detect_edges (
    unsigned char *** image,
    int size_x,
    int size_y )

```

megkeresi a képen található objektumok függőleges széleit

Parameters

in	***image	a kép amin keresni kell
in	size_x	a kép oszlopainak száma
in	size_y	a kép sorainak száma

Mivel egy teljesen új képet hozunk létre, először lemásoljuk a képet. Ezek után sorrendben a következő műveleteket hajtjuk végre:

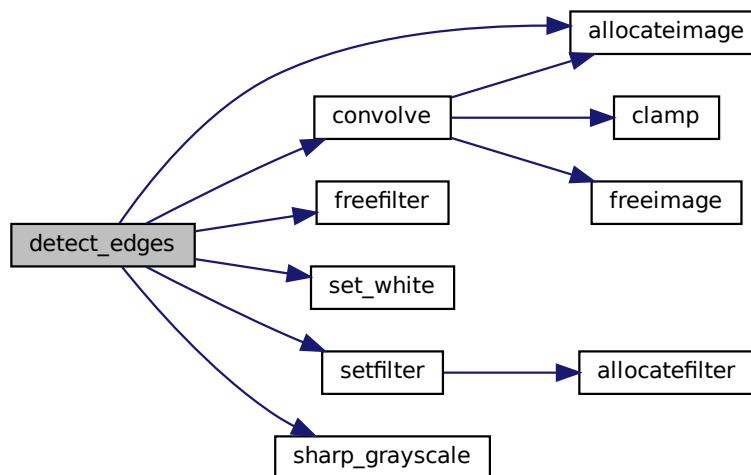
- 3x3 Gauss-blur - ez előkészíti a következő művelethez a képet, mivel így sokkal kevesebb él lesz a képen.
- 3x3 vertical-line filter - a függőleges éleket keresi meg, az élek fehérek, minden más fekete
- set_white a kicsit sötét színek kivételével (9 felett) teljesen fehérre (255) változtatjuk a pixeleket, a többi fekete (0) lesz, így élesebbek lesznek az élek
- 3x3 Gauss-blur tompítjuk az éleket, a következő művelethez
- sharp_grayscale ami a pixel értékéhez legközelebbi szélsőértékhez igazítja a pixel értékét (128 alatt 0, 128 felett 255), így nagyon vékony élek keletkeznek, mivel az előző művelet összemossa a fehér és fekete színeket, így csak a legbelső élek maradnak meg.

Definition at line 254 of file imagefunc.c.

References `allocateimage()`, `convolve()`, `freefilter()`, `set_white()`, `setfilter()`, and `sharp_grayscale()`.

Referenced by `main()`, and `pixelsort()`.

Here is the call graph for this function:



5.4.2.11 freefilter()

```
void freefilter (
    Filter filter )
```

felszabadítja a filtert amit a convolve használ

Parameters

in	<i>filter</i>	a felszabadítandó filter
----	---------------	--------------------------

Definition at line 39 of file imagefunc.c.

References Filter::filt, and Filter::size_y.

Referenced by detect_edges(), and main().

5.4.2.12 grayscale()

```
void grayscale (
    unsigned char pixel[] )
```

fekete-fehérré alakít egy képet

Parameters

in	<i>pixel</i> []	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
----	-----------------	--

See also

használt értékek innen: <https://www.dfstudios.co.uk/articles/programming/image-programming>

Definition at line 205 of file imagefunc.c.

Referenced by main().

5.4.2.13 hsl2rgb()

```
RGB hsl2rgb (
    HSL pixel )
```

HSL színskála átalakítása RGB skálára.

Parameters

in	<i>pixel</i>	egy pixel adatai HSL struktúrában
out	<i>rgbpixel</i>	a pixel RGB struktúráként

See also

pszeudókód innen: <http://marcocorvi.altervista.org/games/imgpr/rgb-hsl.htm>
[hsl2rgbcOLOR](#)

Definition at line 120 of file imagefunc.c.

References RGB::b, RGB::g, HSL::h, hsl2rgbcolor(), HSL::l, RGB::r, and HSL::s.

Referenced by change_light(), and hue_shift().

Here is the call graph for this function:



5.4.2.14 hsl2rgbcolor()

```
double hsl2rgbcolor (
    double temp1,
    double temp2,
    double temp3 )
```

Definition at line 102 of file imagefunc.c.

Referenced by hsl2rgb().

5.4.2.15 hue_shift()

```
void hue_shift (
    unsigned char pixel[],
    double value )
```

A [HSL](#) színskála Hue értékét tolja el.

Parameters

in	<i>pixel[]</i>	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
in	<i>value</i>	mekkora értékkel (-100)-100

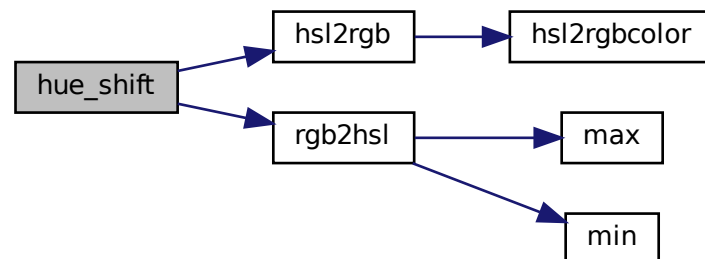
A pixel értékét átalakítja [HSL](#) színskálára, majd a Hue értékhez hozzáadja a választott értéket. Nincs határ megsza-
bva, de mivel a Hue 0 és 1 közötti értéket vesz fel, így a -100-nál kisebb és a 100 nagyobb értékek mod 100-zal
lesznek számolva

Definition at line 645 of file imagefunc.c.

References RGB::b, RGB::g, HSL::h, hsl2rgb(), RGB::r, and rgb2hsl().

Referenced by corrupt(), and main().

Here is the call graph for this function:



5.4.2.16 invert()

```
void invert (
    unsigned char pixel[] )
```

A pixel értékének invertálása.

Parameters

in	<i>pixel</i> []	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
----	-----------------	--

A pixelből kivonunk 255-öt majd az abszolút értékét vesszük, és így kapjuk meg az invertált értéket.

Definition at line 179 of file imagefunc.c.

Referenced by main().

5.4.2.17 mirror_diagonal()

```
void mirror_diagonal (
    unsigned char *** matrix,
    int size_x,
    int size_y )
```

az átló mentén tükrözi a képet

Parameters

in	*** <i>matrix</i>	a tükrözendő matrix (többnyire a kép)
in	<i>size_x</i>	a mátrix oszlopainak száma
in	<i>size_y</i>	a mátrix sorainak száma

A bal felső sarokból kezdve, a pixeleket a jobb alsó sarokba helyezi át. Azért csak a *size_y*/2-ig mert ha ennél tovább menne akkor visszacserélné az egész mátrixot.

Definition at line 319 of file imagefunc.c.

Referenced by corrupt(), and main().

5.4.2.18 mirror_horizontal()

```
void mirror_horizontal (
    unsigned char *** matrix,
    int size_x,
    int size_y )
```

a vízszintes tengelyre tükrözi a képet

Parameters

in	*** <i>matrix</i>	a tükrözendő matrix (többnyire a kép)
in	<i>size_x</i>	a mátrix oszlopainak száma
in	<i>size_y</i>	a mátrix sorainak száma

Az oszlop első elemét a végére helyezi. Azért csak *size_y*/2-ig megy, mert ha ennél tovább menne akkor visszacserélné az egész mátrixot.

Definition at line 355 of file imagefunc.c.

Referenced by main().

5.4.2.19 mirror_vertical()

```
void mirror_vertical (
    unsigned char *** matrix,
    int size_x,
    int size_y )
```

a függőleges tengelyre tükrözi a képet

Parameters

in	*** <i>matrix</i>	a tükrözendő matrix (többnyire a kép)
in	<i>size_x</i>	a mátrix oszlopainak száma
in	<i>size_y</i>	a mátrix sorainak száma

A sor első elemét a végére helyezi. Azért csak `size_x/2`-ig megy, mert ha ennél tovább menne akkor visszacserélné az egész mátrixot.

Definition at line 337 of file `imagefunc.c`.

Referenced by `main()`.

5.4.2.20 pixelsort()

```
void pixelsort (
    unsigned char *** image,
    int size_x,
    int size_y,
    PsOptions options )
```

Végrehajtja a `pixelsort`-ot.

Parameters

in	*** <i>image</i>	a módosítandó kép
in	<i>size_x</i>	a kép oszlopainak száma
in	<i>size_y</i>	a kép sorainak száma
in	<i>options</i>	a <code>pixelsort</code> tulajdonságai a <code>PsOptions</code> struktúráként

See also

[PsOptions](#)

[sortcopy](#)

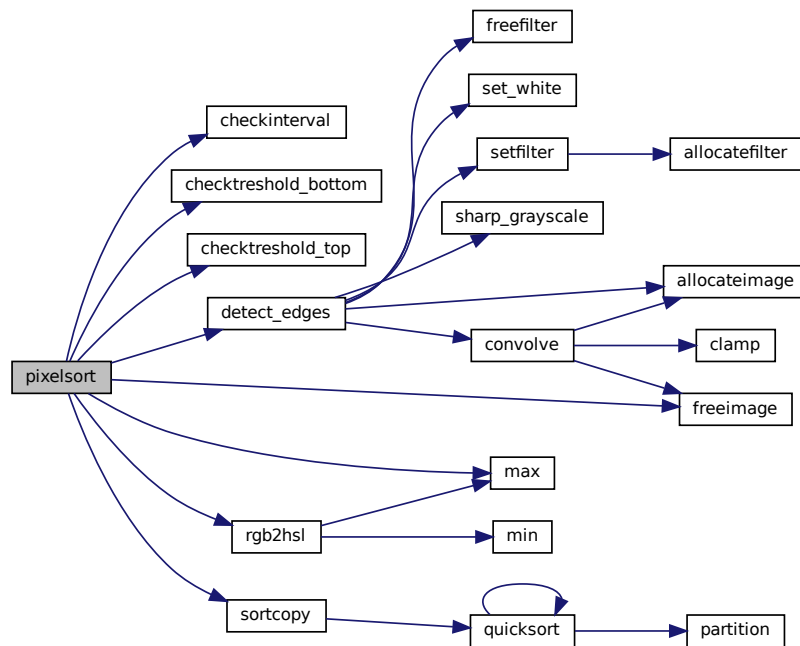
Az algoritmus lényege, hogy a megadott típus alapján ([HSL](#) lightness, [RGB](#) intensity) minden sorban keres egy olyan pixelt ami belefér a megadott threshold-ba (bottom, top). Az `edges` típusnál a kép objektumainak függőleges széleit keresi meg és ezek a határok között rendez. A megtalált pixelnek egy valamilyen a `PsOptions`-ban meghatározott környezetét vesszük és ezen a környezeten sorba rendezzük a pixeleket a megadott típus alapján, a szintén megadott irányba.

Definition at line 540 of file `imagefunc.c`.

References `checkinterval()`, `checkthreshold_bottom()`, `checkthreshold_top()`, `detect_edges()`, `edges`, `freeimage()`, `hsl_l`, `Sort::idx`, `HSL::l`, `max()`, `PsOptions::merge`, `PsOptions::pstype`, `rgb2hsl()`, `rgb_sum`, `sortcopy()`, and `Sort↔::value`.

Referenced by `corrupt()`, and `main()`.

Here is the call graph for this function:



5.4.2.21 rgb2hsl()

HSL `rgb2hsl (`
 unsigned char *pixel*[])

RGB színskála átalakítása **HSL** skálára.

Parameters

in	<i>pixel</i> []	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
out	<i>hslpixel</i>	a pixel adatai HSL struktúráként

See also

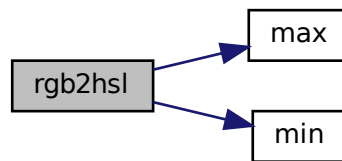
pszeudókód innen: <http://marcocorvi.altervista.org/games/imgpr/rgb-hsl.htm>

Definition at line 66 of file imagefunc.c.

References **HSL::h**, **HSL::l**, **max()**, **min()**, and **HSL::s**.

Referenced by **change_light()**, **hue_shift()**, and **pixelsort()**.

Here is the call graph for this function:



5.4.2.22 rgb_shift()

```
void rgb_shift (
    PPM_Image * image,
    RGB_SHIFT options )
```

Elmozdítja az [RGB](#) színskála értékeit a megadott irányba.

Parameters

in	<i>*image</i>	a kép amin alkalmazni kell
in	<i>options</i>	melyik színt milyen irányba, mennyivel (RGB_SHIFT)

Először a soronként mozgat a rotate segítségével, majd oszloponként a rotate_vertical segítségével.

See also

[RGB_SHIFT](#)

[rotate](#)

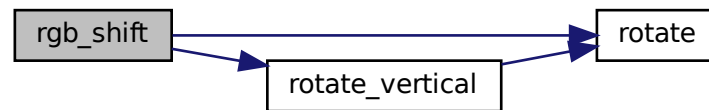
[rotate_vertical](#)

Definition at line 749 of file imagefunc.c.

References [RGB_SHIFT::blue_x](#), [RGB_SHIFT::blue_y](#), [RGB_SHIFT::green_x](#), [RGB_SHIFT::green_y](#), [PPM_Image::image_data](#), [RGB_SHIFT::red_x](#), [RGB_SHIFT::red_y](#), [rotate\(\)](#), [rotate_vertical\(\)](#), [PPM_Image::size_x](#), and [PPM_Image::size_y](#).

Referenced by [corrupt\(\)](#), and [main\(\)](#).

Here is the call graph for this function:



5.4.2.23 rotate()

```
static void rotate (
    unsigned char ** line,
    int size,
    int color,
    int rot ) [static]
```

Megforgatja a paraméterként kapott sort.

Parameters

in	<i>line</i>	a forgatandó sor
in	<i>size</i>	a sor mérete
in	<i>color</i>	melyik színt kell áthelyezni
in	<i>rot</i>	mennyivel kell forgatni

az első lefutásnál a 2. helyre így az első kerül, majd így tovább

ha balra forgatunk akkor a méretből kell elvenni hogy melyik kell, ha jobbra akkor csak eggyel növelünk

azért kell mod size-ot venni, mert így a végén lévő utolsó elem a legelejére kerül

Definition at line 682 of file imagefunc.c.

Referenced by `anaglyph3d()`, `rgb_shift()`, and `rotate_vertical()`.

5.4.2.24 rotate_vertical()

```
static void rotate_vertical (
    PPM_Image * image,
    int col,
    int color,
    int rot ) [static]
```

Oszlopokat forgat meg.

Parameters

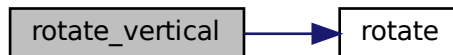
in	<i>*image</i>	a teljes kép ahonnan ki lehet venni az oszlopokat
in	<i>col</i>	melyik oszlopot kell
in	<i>color</i>	melyik színt kell forgatni
in	<i>rot</i>	mennyivel kell forgatni

Definition at line 704 of file imagefunc.c.

References PPM_Image::image_data, rotate(), and PPM_Image::size_y.

Referenced by rgb_shift().

Here is the call graph for this function:

**5.4.2.25 set_white()**

```
void set_white (
    unsigned char pixel[],
    int threshold )
```

a threshold feletti értéket fehérre változtatja, az alatta lévőket feketévé

Parameters

in	<i>pixel[]</i>	a módosítandó pixel
in	<i>threshold</i>	ez az érték felett kell fehérre ez alatt feketévé

Definition at line 229 of file imagefunc.c.

Referenced by detect_edges().

5.4.2.26 setfilter()

```
void setfilter (
    Filter * filter,
```

```

int * filt,
double mult,
int size_x,
int size_y )

```

beállítja egy filter értékét amit a convolve használ

Parameters

in	<i>*filter</i>	a Filter egy példánya pointerként
in	<i>*filt</i>	a betöltendő adatok
in	<i>mult</i>	a filter szorzásához szükséges konstans
in	<i>size_x</i>	filter oszlopainak száma
in	<i>size_y</i>	filter sorainak száma

Definition at line 22 of file imagefunc.c.

References `allocatefilter()`, `Filter::filt`, `Filter::mult`, `Filter::size_x`, and `Filter::size_y`.

Referenced by `detect_edges()`, and `main()`.

Here is the call graph for this function:



5.4.2.27 sharp_grayscale()

```

void sharp_grayscale (
    unsigned char pixel[] )

```

a pixel intenzitása (a három szín átlaga) alapján a pixel értékéhez legközelebbi szélsőértékhez igazítja a pixel értékét (128 alatt 0, 128 felett 255)

Parameters

in	<i>pixel[]</i>	a módosítandó pixel
----	----------------	---------------------

Definition at line 216 of file imagefunc.c.

Referenced by `detect_edges()`.

5.4.2.28 sinecolor_shift()

```
void sinecolor_shift (
    unsigned char pixel[],
    double amplifier,
    double freq,
    double phase,
    double bias )
```

A szinusz függvény alapján eltolja a színeket, RGB színcsatornánként.

Parameters

in	<i>pixel</i> []	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
in	<i>amplifier</i>	a szinusz függvény amplitúdóját adja meg
in	<i>freq</i>	a szinusz függvény frekvenciáját adja meg. Ajánlott értékei:]0.004, 0.05[vagy]10, 50[. Nagyon nagy frekvencia zajos képet eredményez, mivel egymáshoz nagyon közeli színek is teljesen különböző értéket kapnak.
in	<i>phase</i>	a szinusz fáziseltolása
in	<i>bias</i>	megmondja hogy minimum mennyivel kell szorozni a pixel értékét. Ajánlott]0.7, 1[között, különben elkezd sötétedni a kép

A pixelt csatornánként felbontjuk, majd az értékét megszorozzuk a szinusz(pixelcsatorna) értékével. Végül korlátozzuk a 0, 255 intervallumra, mivel az amplitúdó és a bias miatt kicsúszhat a határból.

See also

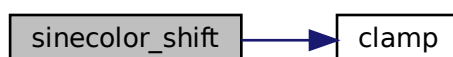
<https://github.com/ImageMagick/ImageMagick/blob/f842055ddb5936c1cacbbcf8c8e06a7be6MagickCore/accelerate-kernels-private.h> CLQuantum ApplyFunction (line 1707)

Definition at line 669 of file imagefunc.c.

References clamp().

Referenced by main().

Here is the call graph for this function:



5.4.2.29 sortcopy()

```
void sortcopy (
    Sort partline[],
    unsigned char *** image,
    int line,
    int start,
    int end,
    int dir )
```

A pixlsort segédfüggvénye ami rendezi és helyére rakja a pixeleket

Parameters

in	<i>partline[]</i>	a rendezésre kiválasztott pixelek
in	<i>***image</i>	a kép ahova vissza kell írni a pixeleket
in	<i>line</i>	a kép aktuálisan módosítandó sora
in	<i>start</i>	a kezdés pozíciója, ahonnan el kell kezdeni a másolást
in	<i>end</i>	ameddig másolni kell
in	<i>dir</i>	a rendezés iránya: 0 balról jobbra 1: jobbról balra

Először rendezzük a pixeleket. Ezután egy ideiglenes tömbbe másoljuk a rendezett elemeknek megfelelő adatokat és utána az iránynak megfelelően visszaírjuk őket az eredeti képbe.

Definition at line 451 of file imagefunc.c.

References Sort::idx, and quicksort().

Referenced by pixelsort().

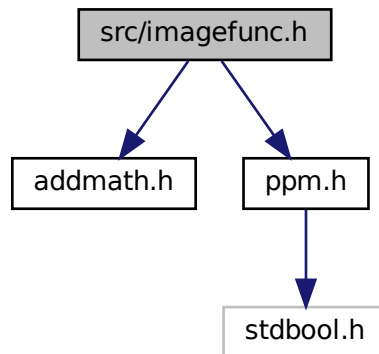
Here is the call graph for this function:



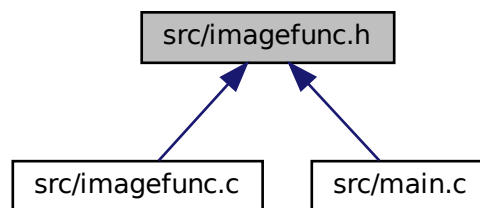
5.5 src/imagefunc.h File Reference

```
#include "addmath.h"
#include "ppm.h"
```

Include dependency graph for imagefunc.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [HSL](#)
[HSL](#) színskála struktúrája.
- struct [RGB](#)
[RGB](#) színskála struktúrája.
- struct [Filter](#)
A konvolúcióhoz használt filter tulajdonságai.
- struct [RGB_SHIFT](#)
Az [RGB](#) shifteléshez használt értékek.
- struct [PsOptions](#)
A pixelsort beállításai.

Typedefs

- typedef struct [HSL](#) [HSL](#)
HSL színskála struktúrája.
- typedef struct [RGB](#) [RGB](#)
RGB színskála struktúrája.
- typedef struct [Filter](#) [Filter](#)
A konvolúcióhoz használt filter tulajdonságai.
- typedef struct [RGB_SHIFT](#) [RGB_SHIFT](#)
Az RGB shifeléshez használt értékek.
- typedef enum [pixelsort_preset](#) [pixelsort_preset](#)
Pixel-sort preset típusa.
- typedef enum [mirror_type](#) [mirror_type](#)
A tükrözés típusa.
- typedef enum [ps_type](#) [ps_type](#)
A pixel-sort típusa.
- typedef enum [ps_option_type](#) [ps_option_type](#)
A pixel-sort threshold és interval értékének kiválasztása.
- typedef struct [PsOptions](#) [PsOptions](#)
A pixel-sort beállításai.

Enumerations

- enum [pixelsort_preset](#) {
[psnone](#), [landscape](#), [macro](#), [fewcolors](#),
[allrandom](#), [dark](#), [edge](#) }
Pixel-sort preset típusa.
- enum [mirror_type](#) { [none](#), [diagonal](#), [vertical](#), [horizontal](#) }
A tükrözés típusa.
- enum [ps_type](#) { [rgb_sum](#), [hsl_l](#), [edges](#) }
A pixel-sort típusa.
- enum [ps_option_type](#) { [ran](#), [man](#) }
A pixel-sort threshold és interval értékének kiválasztása.

Functions

- void [setfilter](#) ([Filter](#) *filter, int *filt, double mult, int size_x, int size_y)
beállítja egy filter értékét amit a convolve használ
- void [freefilter](#) ([Filter](#) filter)
felszabadítja a filtert amit a convolve használ
- int ** [allocatefilter](#) (int size_x, int size_y)
lefoglalja a Filter 2 dimenziós tömbjét
- [HSL rgb2hsl](#) (unsigned char pixel[])
RGB színskála átalakítása HSL skálára.
- double [hsl2rgbcolor](#) (double temp1, double temp2, double temp3)
- [RGB hsl2rgb](#) ([HSL](#) pixel)
HSL színskála átalakítása RGB skálára.
- void [invert](#) (unsigned char pixel[])
A pixel értékének invertálása.
- void [contrast](#) (unsigned char pixel[], double level)

- *Megváltoztatja a pixel kontrasztját.*
- void [grayscale](#) (unsigned char pixel[])
fekete-fehérré alakít egy képet
- void [sharp_grayscale](#) (unsigned char pixel[])
a pixel intenzitása (a három szín átlaga) alapján a pixel értékéhez legközelebbi szélsőértékhez igazítja a pixel értékét (128 alatt 0, 128 felett 255)
- void [set_black](#) (unsigned char pixel[], int threshold)
- unsigned char *** [detect_edges](#) (unsigned char ***image, int size_x, int size_y)
megkeresi a képen található objektumok függőleges széleit
- void [change_light](#) (unsigned char pixel[], int percent)
egy pixel fényességének megváltoztatása
- void [mirror_diagonal](#) (unsigned char ***matrix, int size_x, int size_y)
az átló mentén tükrözi a képet
- void [mirror_vertical](#) (unsigned char ***matrix, int size_x, int size_y)
a függőleges tengelyre tükrözi a képet
- void [mirror_horizontal](#) (unsigned char ***matrix, int size_x, int size_y)
a vízszintes tengelyre tükrözi a képet
- void [convolve](#) (unsigned char ***original, int size_x, int size_y, [Filter](#) filter, int times)
végrehajtja a konvolúciót, ami a blur és sharpen lépésekhez kell
- void [sortcopy](#) ([Sort](#) partline[], unsigned char ***image, int line, int start, int elem, int dir)
- void [pixelsort](#) (unsigned char ***image, int size_x, int size_y, [PsOptions](#) options)
Végrehajtja a pixelsort-ot.
- void [hue_shift](#) (unsigned char pixel[], double value)
A HSL színskála Hue értékét tolja el.
- void [sinecolor_shift](#) (unsigned char pixel[], double amplifier, double freq, double phase, double bias)
A szinusz függvény alapján eltolja a színeket, RGB színcsatornánként.
- void [rgb_shift](#) ([PPM_Image](#) *image, [RGB_SHIFT](#) options)
Elmozdítja az RGB színskála értékeit a megadott irányba.
- void [anaglyph3d](#) ([PPM_Image](#) *image)
Vörös-Cián 3D képpé alakítja a bemenetet.
- void [corrupt](#) ([PPM_Image](#) *image)
Véletlenszerűen tönkreteszi a képet.

5.5.1 Typedef Documentation

5.5.1.1 Filter

```
typedef struct Filter Filter
```

A konvolúcióhoz használt filter tulajdonságai.

5.5.1.2 HSL

```
typedef struct HSL HSL
```

[HSL](#) színskála struktúrája.

5.5.1.3 mirror_type

```
typedef enum mirror_type mirror_type
```

A tükrözés típusa.

5.5.1.4 pixelsort_preset

```
typedef enum pixelsort_preset pixelsort_preset
```

Pixelsort preset típusa.

5.5.1.5 ps_option_type

```
typedef enum ps_option_type ps_option_type
```

A pixelsort treshold és interval értékének kiválasztása.

5.5.1.6 ps_type

```
typedef enum ps_type ps_type
```

A pixelsort típusa.

See also

[pixelsort](#)

5.5.1.7 PsOptions

```
typedef struct PsOptions PsOptions
```

A pixelsort beállításai.

5.5.1.8 RGB

```
typedef struct RGB RGB
```

[RGB](#) színskála struktúrája.

5.5.1.9 RGB_SHIFT

```
typedef struct RGB_SHIFT RGB_SHIFT
```

Az RGB shifteléshez használt értékek.

5.5.2 Enumeration Type Documentation

5.5.2.1 mirror_type

```
enum mirror_type
```

A tükrözés típusa.

Enumerator

none	
diagonal	
vertical	
horizontal	

Definition at line 63 of file imagefunc.h.

5.5.2.2 pixelsort_preset

```
enum pixelsort_preset
```

Pixelsort preset típusa.

Enumerator

psnone	Semmilyen rendezés
landscape	Nagy objektumokhoz használt rendezés
macro	Kis objektumokhoz használt rendezés
fewcolors	Kevés színhez használt rendezés
allrandom	Teljesen véletlenszerű rendezés
dark	Sötét képekhez használt rendezés
edge	A képen található objektumok széle alapján rendezés

Definition at line 50 of file imagefunc.h.

5.5.2.3 ps_option_type

enum [ps_option_type](#)

A pixelsort treshold és interval értékének kiválasztása.

Enumerator

ran	véletlenszerű az alsó és felső határok között
man	csak az alsó határ

Definition at line 83 of file imagefunc.h.

5.5.2.4 ps_type

enum [ps_type](#)

A pixelsort típusa.

See also

[pixelsort](#)

Enumerator

rgb_sum	RGB intensity
hsl_l	HSL Lightness
edges	Megkeresi az objektumok szélét és onnan kezdi

Definition at line 74 of file imagefunc.h.

5.5.3 Function Documentation

5.5.3.1 allocatefilter()

```
int** allocatefilter (
    int size_x,
    int size_y )
```

lefoglalja a [Filter](#) 2 dimenziós tömbjét

Parameters

in	<i>size</i> _↔ _x	a filter oszlopainak száma
in	<i>size</i> _↔ _y	a filter sorainak száma

Definition at line 51 of file imagefunc.c.

Referenced by setfilter().

5.5.3.2 anaglyph3d()

```
void anaglyph3d (
    PPM_Image * image )
```

Vörös-Cián 3D képpé alakítja a bemenetet.

Parameters

in	* <i>image</i>	átalakítandó kép (PPM_Image)
----	----------------	--

A lényege, hogy a vörös csatornát el kell tolni balra valamennyivel. Ez az érték a szélesség nagyjából 0,925%-a (1080 pixel szélességnél kb. 10 pixel). A kép úgy fog kinézni, mintha be lenne süllyesztve.

See also

[PPM_Image](#)

[rotate](#)

Definition at line 794 of file imagefunc.c.

References [PPM_Image::image_data](#), [rotate\(\)](#), [PPM_Image::size_x](#), and [PPM_Image::size_y](#).

Referenced by [main\(\)](#).

Here is the call graph for this function:



5.5.3.3 change_light()

```
void change_light (
    unsigned char pixel[],
    int percent )
```

egy pixel fényességének megváltoztatása

Parameters

in	<i>pixel</i> []	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
in	<i>percent</i>	mennyi százalékkal

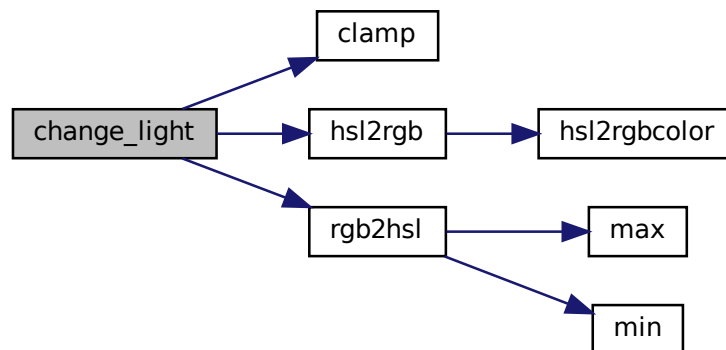
A pixel értéket először **HSL** színskálára alakítjuk át, majd ehhez hozzáadjuk a változtatás mértékét, végül korlátozzuk 0, 1 intervallumra és visszaalakítjuk **RGB** színskálára.

Definition at line 298 of file imagefunc.c.

References `RGB::b`, `clamp()`, `RGB::g`, `hsl2rgb()`, `HSL::l`, `RGB::r`, and `rgb2hsl()`.

Referenced by `corrupt()`, and `main()`.

Here is the call graph for this function:



5.5.3.4 contrast()

```
void contrast (
    unsigned char pixel[],
    double level )
```

Megváltoztatja a pixel kontrasztját.

Parameters

in	<i>pixel[]</i>	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
in	<i>level</i>	a változtatás mértéke (a 259 nem megengedett)

See also

pszeudokód innen: <https://www.dfstudios.co.uk/articles/programming/image-programming-al>

Definition at line 191 of file imagefunc.c.

References clamp().

Referenced by corrupt(), and main().

Here is the call graph for this function:



5.5.3.5 convolve()

```

void convolve (
    unsigned char *** original,
    int size_x,
    int size_y,
    Filter filter,
    int times )
  
```

végrehajtja a konvolúciót, ami a blur és sharpen lépésekhez kell

See also

pszeudokód és működési elv itt: [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))

[Filter](#)

Parameters

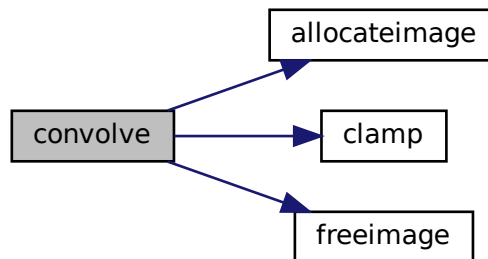
in	<i>***original</i>	módosítandó kép
in	<i>size_x</i>	a kép oszlopainak száma
in	<i>size_y</i>	a kép sorainak száma
in	<i>filter</i>	a használandó filter

Definition at line 375 of file imagefunc.c.

References `allocateimage()`, `clamp()`, `Filter::filt`, `freeimage()`, `Filter::mult`, `Filter::size_x`, and `Filter::size_y`.

Referenced by `detect_edges()`, and `main()`.

Here is the call graph for this function:



5.5.3.6 corrupt()

```
void corrupt (
    PPM_Image * image )
```

Véletlenszerűen tönkreteszi a képet.

Parameters

in	*** <i>image</i>	a módosítandó kép
----	------------------	-------------------

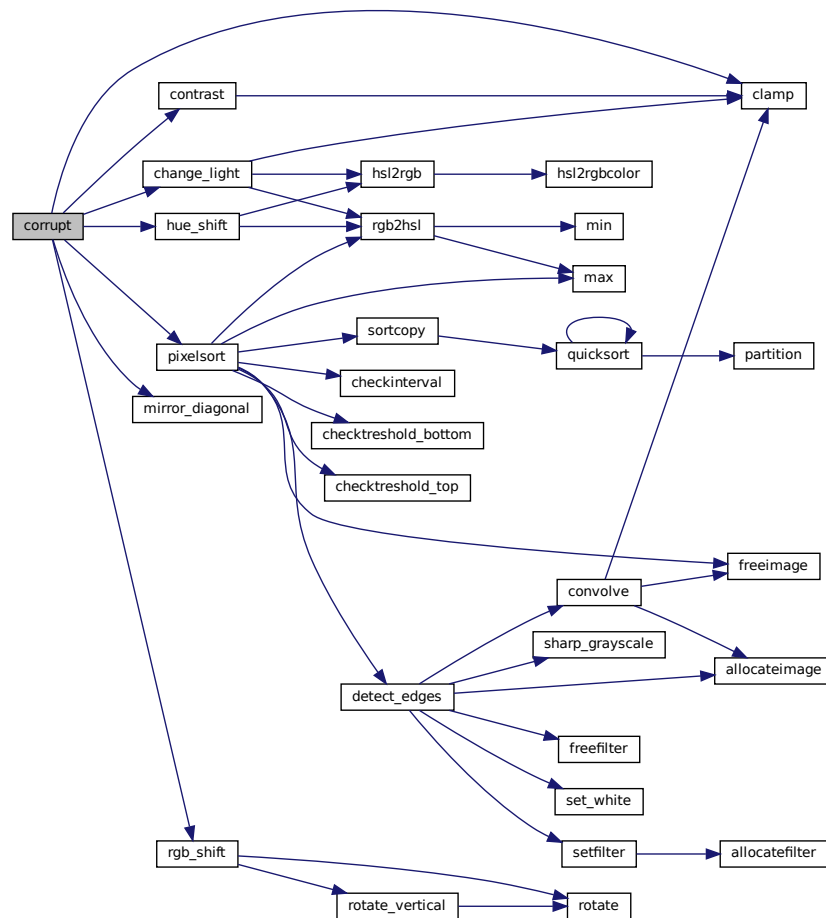
Először kiválasztunk 3 értéket amivel sorrendben a: fényességet (`change_light`), a kontrasztot (`contrast`) és a [HSL](#) Hue -t (`hue_shift`) módosítjuk. Ezek után a kép bizonyos részeit tükrözzük, ha páratlan számú alkalommal, akkor a kép egy része fordítva lesz, ha páros számú alkalommal, akkor az eredeti irányban. Ez után az [RGB](#) színeit kell shiftelni véletlenszerű irányba és értékkel. Ezután a lehető legbővebb véletlenszerű beállítással végrehajtjuk a pixelsort -ot

Definition at line 810 of file imagefunc.c.

References `change_light()`, `clamp()`, `contrast()`, `hsl_l`, `hue_shift()`, `PPM_Image::image_data`, `mirror_diagonal()`, `pixelsort()`, `ran`, `rgb_shift()`, `PPM_Image::size_x`, and `PPM_Image::size_y`.

Referenced by `main()`.

Here is the call graph for this function:



5.5.3.7 detect_edges()

```

unsigned char*** detect_edges (
    unsigned char *** image,
    int size_x,
    int size_y )
  
```

megkeresi a képen található objektumok függőleges széleit

Parameters

in	*** <i>image</i>	a kép amin keresni kell
in	<i>size_x</i>	a kép oszlopainak száma
in	<i>size_y</i>	a kép sorainak száma

Mivel egy teljesen új képet hozunk létre, először lemásoljuk a képet. Ezek után sorrendben a következő műveleteket

hajtjuk végre:

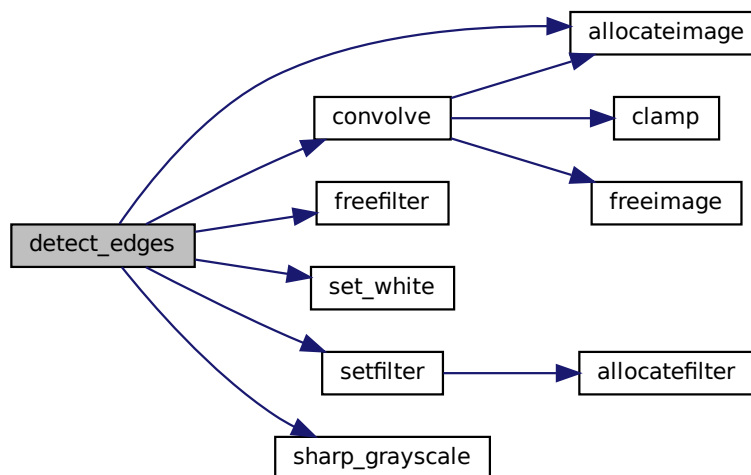
- 3x3 Gauss-blur - ez előkészíti a következő művelethez a képet, mivel így sokkal kevesebb él lesz a képen.
- 3x3 vertical-line filter - a függőleges éleket keresi meg, az élek fehérek, minden más fekete
- set_white a kicsit sötét színek kivételével (9 felett) teljesen fehérre (255) változtatjuk a pixeleket, a többi fekete (0) lesz, így élesebbek lesznek az élek
- 3x3 Gauss-blur tompítjuk az éleket, a következő művelethez
- sharp_grayscale ami a pixel értékéhez legközelebbi szélsőértékhez igazítja a pixel értékét (128 alatt 0, 128 felett 255), így nagyon vékony élek keletkeznek, mivel az előző művelet összemossa a fehér és fekete színeket, így csak a legbelső élek maradnak meg.

Definition at line 254 of file imagefunc.c.

References `allocateimage()`, `convolve()`, `freefilter()`, `set_white()`, `setfilter()`, and `sharp_grayscale()`.

Referenced by `main()`, and `pixelsort()`.

Here is the call graph for this function:



5.5.3.8 freefilter()

```
void freefilter (
    Filter filter )
```

felszabadítja a filtert amit a convolve használ

Parameters

in	<i>filter</i>	a felszabadítandó filter
----	---------------	--------------------------

Definition at line 39 of file imagefunc.c.

References Filter::filt, and Filter::size_y.

Referenced by detect_edges(), and main().

5.5.3.9 grayscale()

```
void grayscale (
    unsigned char pixel[] )
```

fekete-fehérré alakít egy képet

Parameters

in	<i>pixel</i> []	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
----	-----------------	--

See also

használt értékek innen: <https://www.dfstudios.co.uk/articles/programming/image-programming>

Definition at line 205 of file imagefunc.c.

Referenced by main().

5.5.3.10 hsl2rgb()

```
RGB hsl2rgb (
    HSL pixel )
```

HSL színskála átalakítása RGB skálára.

Parameters

in	<i>pixel</i>	egy pixel adatai HSL struktúrában
out	<i>rgbpixel</i>	a pixel RGB struktúráként

See also

pszeudókód innen: <http://marcocorvi.altervista.org/games/imgpr/rgb-hsl.htm>
[hsl2rgbcOLOR](#)

Definition at line 120 of file imagefunc.c.

References RGB::b, RGB::g, HSL::h, hsl2rgbcolor(), HSL::l, RGB::r, and HSL::s.

Referenced by change_light(), and hue_shift().

Here is the call graph for this function:



5.5.3.11 hsl2rgbcolor()

```
double hsl2rgbcolor (
    double temp1,
    double temp2,
    double temp3 )
```

Definition at line 102 of file imagefunc.c.

Referenced by hsl2rgb().

5.5.3.12 hue_shift()

```
void hue_shift (
    unsigned char pixel[],
    double value )
```

A [HSL](#) színskála Hue értékét tolja el.

Parameters

in	<i>pixel[]</i>	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
in	<i>value</i>	mekkora értékkel (-100)-100

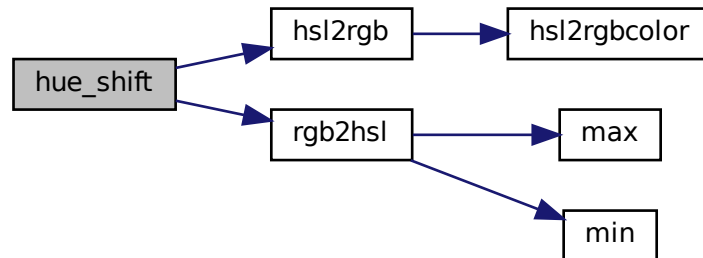
A pixel értékét átalakítja [HSL](#) színskálára, majd a Hue értékhez hozzáadja a választott értéket. Nincs határ megsza-
bva, de mivel a Hue 0 és 1 közötti értéket vesz fel, így a -100-nál kisebb és a 100 nagyobb értékek mod 100-zal
lesznek számolva

Definition at line 645 of file imagefunc.c.

References RGB::b, RGB::g, HSL::h, hsl2rgb(), RGB::r, and rgb2hsl().

Referenced by corrupt(), and main().

Here is the call graph for this function:



5.5.3.13 invert()

```
void invert (
    unsigned char pixel[] )
```

A pixel értékének invertálása.

Parameters

in	<i>pixel</i> []	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
----	-----------------	--

A pixelből kivonunk 255-öt majd az abszolút értékét vesszük, és így kapjuk meg az invertált értéket.

Definition at line 179 of file imagefunc.c.

Referenced by main().

5.5.3.14 mirror_diagonal()

```
void mirror_diagonal (
    unsigned char *** matrix,
    int size_x,
    int size_y )
```

az átló mentén tükrözi a képet

Parameters

in	*** <i>matrix</i>	a tükrözendő matrix (többnyire a kép)
in	<i>size_x</i>	a mátrix oszlopainak száma
in	<i>size_y</i>	a mátrix sorainak száma

A bal felső sarokból kezdve, a pixeleket a jobb alsó sarokba helyezi át. Azért csak a *size_y*/2-ig mert ha ennél tovább menne akkor visszacserélné az egész mátrixot.

Definition at line 319 of file imagefunc.c.

Referenced by corrupt(), and main().

5.5.3.15 mirror_horizontal()

```
void mirror_horizontal (
    unsigned char *** matrix,
    int size_x,
    int size_y )
```

a vízszintes tengelyre tükrözi a képet

Parameters

in	*** <i>matrix</i>	a tükrözendő matrix (többnyire a kép)
in	<i>size_x</i>	a mátrix oszlopainak száma
in	<i>size_y</i>	a mátrix sorainak száma

Az oszlop első elemét a végére helyezi. Azért csak *size_y*/2-ig megy, mert ha ennél tovább menne akkor visszacserélné az egész mátrixot.

Definition at line 355 of file imagefunc.c.

Referenced by main().

5.5.3.16 mirror_vertical()

```
void mirror_vertical (
    unsigned char *** matrix,
    int size_x,
    int size_y )
```

a függőleges tengelyre tükrözi a képet

Parameters

in	*** <i>matrix</i>	a tükrözendő matrix (többnyire a kép)
in	<i>size_x</i>	a mátrix oszlopainak száma
in	<i>size_y</i>	a mátrix sorainak száma

A sor első elemét a végére helyezi. Azért csak `size_x/2`-ig megy, mert ha ennél tovább menne akkor visszacserélné az egész mátrixot.

Definition at line 337 of file `imagefunc.c`.

Referenced by `main()`.

5.5.3.17 pixelsort()

```
void pixelsort (
    unsigned char *** image,
    int size_x,
    int size_y,
    PsOptions options )
```

Végrehajtja a `pixelsort`-ot.

Parameters

in	*** <i>image</i>	a módosítandó kép
in	<i>size_x</i>	a kép oszlopainak száma
in	<i>size_y</i>	a kép sorainak száma
in	<i>options</i>	a <code>pixelsort</code> tulajdonságai a <code>PsOptions</code> struktúraként

See also

[PsOptions](#)

[sortcopy](#)

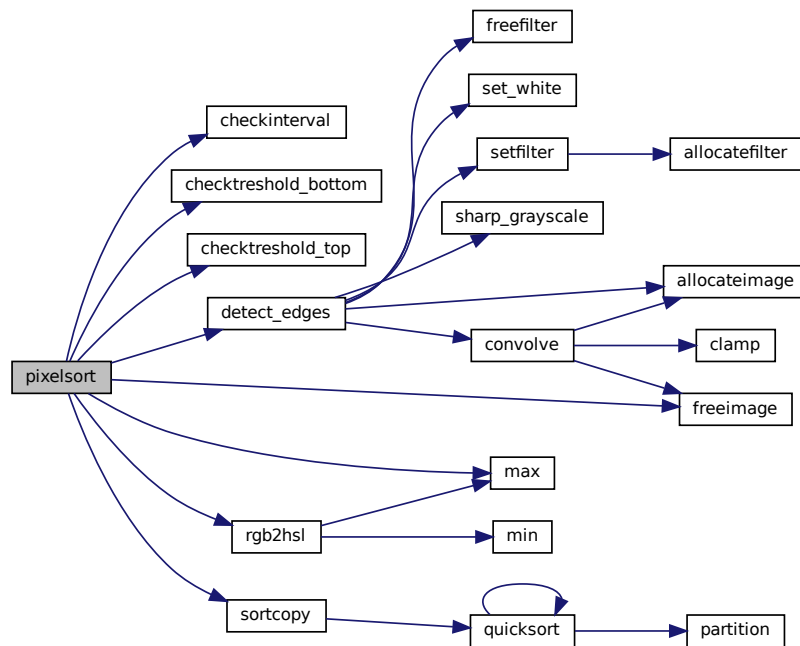
Az algoritmus lényege, hogy a megadott típus alapján ([HSL](#) lightness, [RGB](#) intensity) minden sorban keres egy olyan pixelt ami belefér a megadott threshold-ba (bottom, top). Az `edges` típusnál a kép objektumainak függőleges széleit keresi meg és ezek a határok között rendez. A megtalált pixelnek egy valamilyen a `PsOptions`-ban meghatározott környezetét vesszük és ezen a környezeten sorba rendezzük a pixeleket a megadott típus alapján, a szintén megadott irányba.

Definition at line 540 of file `imagefunc.c`.

References `checkinterval()`, `checkthreshold_bottom()`, `checkthreshold_top()`, `detect_edges()`, `edges`, `freeimage()`, `hsl_l`, `Sort::idx`, `HSL::l`, `max()`, `PsOptions::merge`, `PsOptions::pstype`, `rgb2hsl()`, `rgb_sum`, `sortcopy()`, and `Sort↔::value`.

Referenced by `corrupt()`, and `main()`.

Here is the call graph for this function:



5.5.3.18 rgb2hsl()

HSL `rgb2hsl (`
 unsigned char *pixel*[])

RGB színskála átalakítása **HSL** skálára.

Parameters

in	<i>pixel</i> []	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
out	<i>hslpixel</i>	a pixel adatai HSL struktúráként

See also

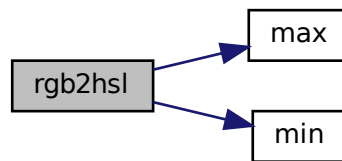
pszeudókód innen: <http://marcocorvi.altervista.org/games/imgpr/rgb-hsl.htm>

Definition at line 66 of file imagefunc.c.

References **HSL::h**, **HSL::l**, **max()**, **min()**, and **HSL::s**.

Referenced by **change_light()**, **hue_shift()**, and **pixelsort()**.

Here is the call graph for this function:



5.5.3.19 rgb_shift()

```
void rgb_shift (
    PPM_Image * image,
    RGB_SHIFT options )
```

Elmozdítja az [RGB](#) színskála értékeit a megadott irányba.

Parameters

in	<i>*image</i>	a kép amin alkalmazni kell
in	<i>options</i>	melyik színt milyen irányba, mennyivel (RGB_SHIFT)

Először a soronként mozgat a rotate segítségével, majd oszloponként a rotate_vertical segítségével.

See also

[RGB_SHIFT](#)

[rotate](#)

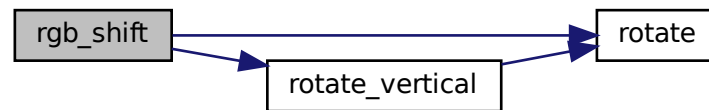
[rotate_vertical](#)

Definition at line 749 of file imagefunc.c.

References [RGB_SHIFT::blue_x](#), [RGB_SHIFT::blue_y](#), [RGB_SHIFT::green_x](#), [RGB_SHIFT::green_y](#), [PPM_Image::image_data](#), [RGB_SHIFT::red_x](#), [RGB_SHIFT::red_y](#), [rotate\(\)](#), [rotate_vertical\(\)](#), [PPM_Image::size_x](#), and [PPM_Image::size_y](#).

Referenced by [corrupt\(\)](#), and [main\(\)](#).

Here is the call graph for this function:



5.5.3.20 set_black()

```
void set_black (
    unsigned char pixel[],
    int treshold )
```

5.5.3.21 setfilter()

```
void setfilter (
    Filter * filter,
    int * filt,
    double mult,
    int size_x,
    int size_y )
```

beállítja egy filter értékét amit a convolve használ

Parameters

in	<i>*filter</i>	a Filter egy példánya pointerként
in	<i>*filt</i>	a betöltendő adatok
in	<i>mult</i>	a filter szorzásához szükséges konstans
in	<i>size_x</i>	filter oszlopainak száma
in	<i>size_y</i>	filter sorainak száma

Definition at line 22 of file imagefunc.c.

References [allocatefilter\(\)](#), [Filter::filt](#), [Filter::mult](#), [Filter::size_x](#), and [Filter::size_y](#).

Referenced by [detect_edges\(\)](#), and [main\(\)](#).

Here is the call graph for this function:



5.5.3.22 sharp_grayscale()

```
void sharp_grayscale (
    unsigned char pixel[] )
```

a pixel intenzitása (a három szín átlaga) alapján a pixel értékéhez legközelebbi szélsőértékhez igazítja a pixel értékét (128 alatt 0, 128 felett 255)

Parameters

in	<i>pixel</i> []	a módosítandó pixel
----	-----------------	---------------------

Definition at line 216 of file imagefunc.c.

Referenced by detect_edges().

5.5.3.23 sinecolor_shift()

```
void sinecolor_shift (
    unsigned char pixel[],
    double amplifier,
    double freq,
    double phase,
    double bias )
```

A szinusz függvény alapján eltolja a színeket, [RGB](#) színcsatornánként.

Parameters

in	<i>pixel</i> []	egy pixel RGB adatai nyers állapotban (Red, Green, Blue)
in	<i>amplifier</i>	a szinusz függvény amplitúdóját adja meg
in	<i>freq</i>	a szinusz függvény frekvenciáját adja meg. Ajánlott értékei:]0.004, 0.05[vagy]10, 50[. Nagyon nagy frekvencia zajos képet eredményez, mivel egymáshoz nagyon közeli színek is teljesen különböző értéket kapnak.
in	<i>phase</i>	a szinusz fáziseltolása
in	<i>bias</i>	megmondja hogy minimum mennyivel kell szorozni a pixel értékét. Ajánlott]0.7, 1[között,
		különben elkezd sötétedni a kép

A pixelt csatornánként felbontjuk, majd az értékét megszorozzuk a szinusz(pixelcsatorna) értékével. Végül korlátozzuk a 0, 255 intervallumra, mivel az amplitúdó és a bias miatt kicsúszhat a határból.

See also

<https://github.com/ImageMagick/ImageMagick/blob/f842055ddb5936c1cacbbcf8c8e06a7be6MagickCore/accelerate-kernels-private.h> CLQuantum ApplyFunction (line 1707)

Definition at line 669 of file imagefunc.c.

References clamp().

Referenced by main().

Here is the call graph for this function:



5.5.3.24 sortcopy()

```

void sortcopy (
    Sort partline[],
    unsigned char *** image,
    int line,
    int start,
    int end,
    int dir )
  
```

A pixlsort segédfüggvénye ami rendezi és helyére rakja a pixeleket

Parameters

in	<i>partline[]</i>	a rendezésre kiválasztott pixelek
in	<i>***image</i>	a kép ahova vissza kell írni a pixeleket
in	<i>line</i>	a kép aktuálisan módosítandó sora
in	<i>start</i>	a kezdés pozíciója, ahonnan el kell kezdeni a másolást
in	<i>end</i>	ameddig másolni kell
in	<i>dir</i>	a rendezés iránya: 0 balról jobbra 1: jobbról balra

Először rendezzük a pixeleket. Ezután egy ideiglenes tömbbe másoljuk a rendezett elemeknek megfelelő adatokat és utána az iránynak megfelelően visszaírjuk őket az eredeti képbe.

Definition at line 451 of file imagefunc.c.

References Sort::idx, and quicksort().

Referenced by pixelsort().

Here is the call graph for this function:



5.6 src/main.c File Reference

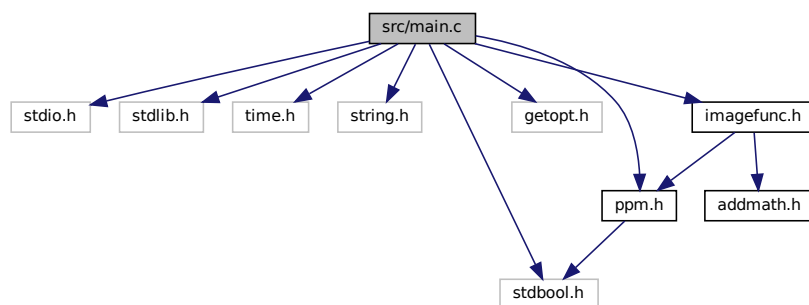
a fő fájl, ami a parancssort kezeli

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include <stdbool.h>
#include <getopt.h>
#include "ppm.h"
#include "imagefunc.h"

```

Include dependency graph for main.c:



Functions

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

5.6.1 Detailed Description

a fő fájl, ami a parancssort kezeli

debugmalloc helyett a Valgrindot használtam a tesztelésre és nem jelzett hibát

5.6.2 Function Documentation

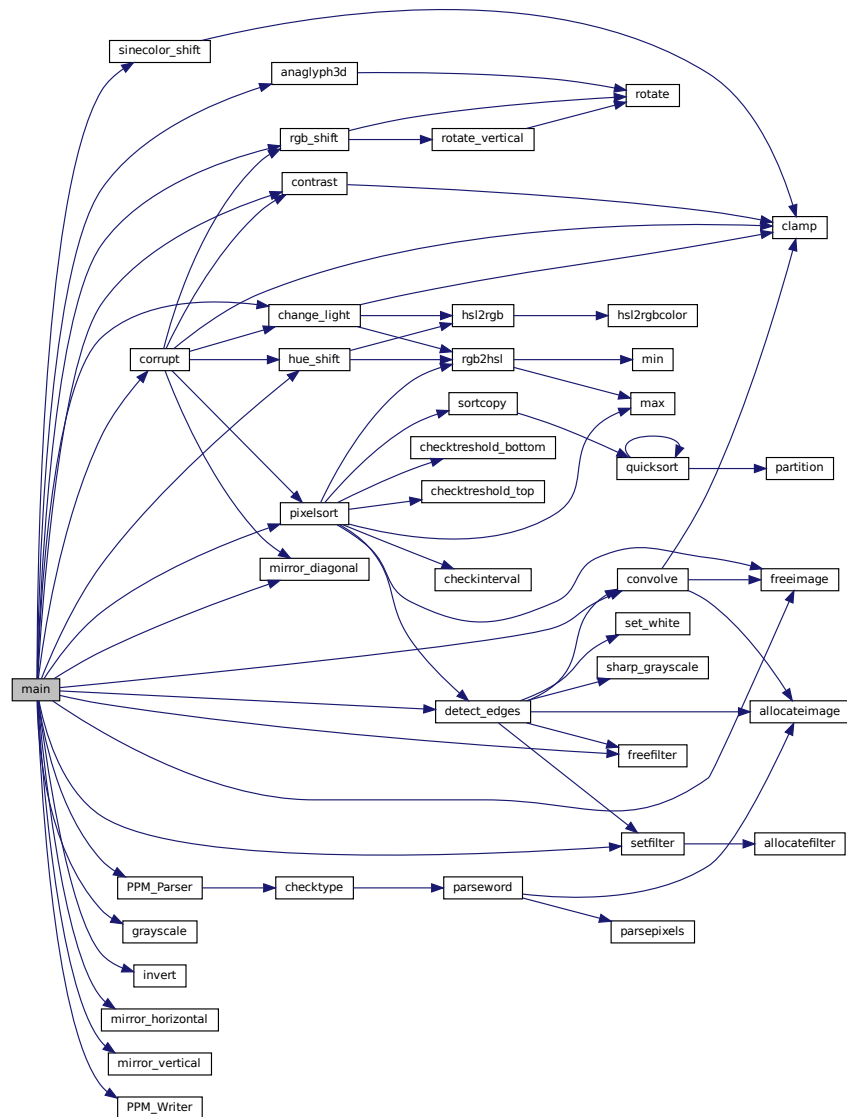
5.6.2.1 main()

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

Definition at line 17 of file main.c.

References allrandom, anaglyph3d(), change_light(), contrast(), convolve(), corrupt(), dark, detect_edges(), diagonal, edge, edges, fewcolors, freefilter(), freeimage(), grayscale(), horizontal, hsl_l, hue_shift(), PPM_Image::image_data, PsOptions::interval, PsOptions::interval_max, PsOptions::interval_min, invert(), landscape, macro, man, PsOptions::merge, mirror_diagonal(), mirror_horizontal(), mirror_vertical(), none, pixelsort(), PPM_Parser(), PPM_Writer(), psnone, PsOptions::pstype, ran, rgb_shift(), rgb_sum, setfilter(), sinecolor_shift(), PPM_Image::size_x, PPM_Image::size_y, PsOptions::threshold, PsOptions::threshold_bottom_max, PsOptions::threshold_bottom_min, PsOptions::threshold_top_max, PsOptions::threshold_top_min, and vertical.

Here is the call graph for this function:



5.7 src/ppm.c File Reference

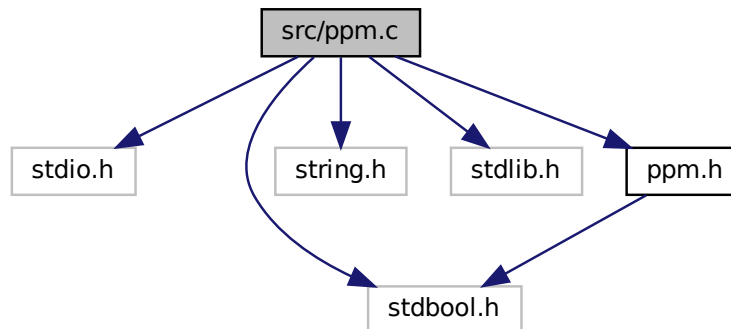
PPM fájl beolvasása.

```

#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include <stdlib.h>
#include "ppm.h"

```


Include dependency graph for ppm.c:



Functions

- void [freeimage](#) (unsigned char ***image, int size_x, int size_y)
- unsigned char *** [allocateimage](#) (int size_x, int size_y)
A kép tárolására használt 3 dimenziós tömb lefoglalása.
- bool [parsepixels](#) (char word[], [PPM_Image](#) *image, int *iter_h, int *iter_w, int *rgb)
egy pixel értékeinek beállítása a kapott szó alapján
- void [parseword](#) (char word[], [PPM_Image](#) *image, int *iter_h, int *iter_w, int *rgb)
Eldönti, hogy mit kell kezdeni a kapott szóval. Először azt nézi meg, hogy van-e magic. Ha nincs ezt kell beállítani. Majd sorban a sorszámot, oszlopszámot és a maxvált. Ha megvan az oszlopszám, le lehet foglalni a kép tömbjét. Ezek után a maradék biztosan pixel érték lesz.
- void [checktype](#) (char line[], [PPM_Image](#) *image, int *iter_h, int *iter_w, int *rgb)
Eldönti, hogy a megadott sor mit tartalmaz és mit kell vele csinálni Ha comment akkor kimarad, egyébként szóközönként fel kell bontani. Ezek lesznek a szavak, amiket utána tovább kell kategorizálni.
- [PPM_Image](#) [PPM_Parser](#) (char filename[])
beolvas egy képet Megnyitja a fájlt, majd létrehozza a [PPM_Image](#) struktúra egy példányát és beállítja a kezdő értékeit. Ezek után maximum 4096 karakter hosszú soronként beolvassa a fájlt és átadja a [checktype](#) -nak. Végül bezárja a fájlt.
- void [PPM_Writer](#) (char filename[], [PPM_Image](#) *image)
Fájlba írja a [PPM_Image](#) tartalmát A megnyitott fájlba először kiírjuk sorrendben a magic-et az oszlopok számát, a sorok számát, a maxvált. Ezek után a pixelek adatait írjuk ki, egy sorba egy pixelt, tehát három számot. Végül bezárjuk a fájlt.

5.7.1 Detailed Description

PPM fájl beolvasása.

5.7.2 Function Documentation

5.7.2.1 allocateimage()

```
unsigned char*** allocateimage (
    int size_x,
    int size_y )
```

A kép tárolására használt 3 dimenziós tömb lefoglalása.

Parameters

in	<i>size_x</i>	a kép oszlopainak száma
in	<i>size_y</i>	a kép sorainak száma
out	<i>image</i>	a létrehozott 3 dimenziós tömb

Definition at line 35 of file ppm.c.

Referenced by `convolve()`, `detect_edges()`, and `parseword()`.

5.7.2.2 checktype()

```
void checktype (
    char line[],
    PPM_Image * image,
    int * iter_h,
    int * iter_w,
    int * rgb )
```

Eldönti, hogy a megadott sor mit tartalmaz és mit kell vele csinálni Ha comment akkor kimarad, egyébként szóközőnként fel kell bontani. Ezek lesznek a szavak, amiket utána tovább kell kategorizálni.

Parameters

in	<i>line[]</i>	a sor amit vizsgálni kell
in	<i>*image</i>	PPM_Image kép aminek az image_data részébe kell beírni a pixelt
in	<i>*iter_h</i>	a sor, amibe írni kell a pixelt
in	<i>*iter_w</i>	az oszlop, amibe írni kell a pixelt
in	<i>*rgb</i>	megadja, hogy ez melyik szín

See also

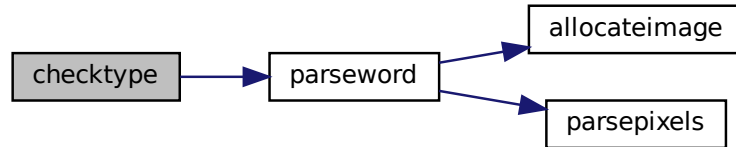
[parseword](#)

Definition at line 150 of file ppm.c.

References `parseword()`.

Referenced by PPM_Parser().

Here is the call graph for this function:



5.7.2.3 freeimage()

```
void freeimage (
    unsigned char *** image,
    int size_x,
    int size_y )
```

A kép tárolására használt 3 dimenziós tömb felszabadítása

Parameters

in	***image	felszabadítandó kép
in	size_x	a kép oszlopainak száma
in	size_y	a kép sorainak száma

Definition at line 19 of file ppm.c.

Referenced by `convolve()`, `main()`, and `pixelsort()`.

5.7.2.4 parsepixels()

```
bool parsepixels (
    char word[],
    PPM_Image * image,
    int * iter_h,
    int * iter_w,
    int * rgb )
```

egy pixel értékeinek beállítása a kapott szó alapján

A megkapott szót unsigned char-rá alakítja és beírja a képbe. Növeli az aktuális oszlop számát, és ha eléri a sorszélességet az aktuális sor számát is növeli. Az rgb értéket mindig növeli, de ha eléri a 3-at, átírja 0-ra. Ha az aktuális sor értéke megegyezik a kép sorainak számával az azt jelenti, hogy betelt a kép és megszakad a beolvasás.

Parameters

in	<i>word[]</i>	a szó amit pixellé kell átalakítani
in	<i>*image</i>	PPM_Image kép aminek az image_data részébe kell beírni a pixelt
in	<i>*iter_h</i>	a sor, amibe írni kell a pixelt
in	<i>*iter_w</i>	az oszlop, amibe írni kell a pixelt
in	<i>*rgb</i>	megadja, hogy ez melyik szín
out	<i>ongoing</i>	true ha még mindig fér a képbe, false ha betelt

See also

[PPM_Image](#)

Definition at line 78 of file ppm.c.

References [PPM_Image::image_data](#), [PPM_Image::maxval](#), [PPM_Image::size_x](#), and [PPM_Image::size_y](#).

Referenced by [parseword\(\)](#).

5.7.2.5 parseword()

```
void parseword (
    char word[],
    PPM_Image * image,
    int * iter_h,
    int * iter_w,
    int * rgb )
```

Eldönti, hogy mit kell kezdeni a kapott szóval. Először azt nézi meg, hogy van-e magic. Ha nincs ezt kell beállítani. Majd sorban a sorszámot, oszlopszámot és a maxvalt. Ha megvan az oszlopszám, le lehet foglalni a kép tömbjét. Ezek után a maradék biztosan pixel érték lesz.

Parameters

in	<i>word[]</i>	a szó amit le kell ellenőrizni
in	<i>*image</i>	PPM_Image kép aminek az image_data részébe kell beírni a pixelt
in	<i>*iter↵ _h</i>	a sor, amibe írni kell a pixelt
in	<i>*iter↵ _w</i>	az oszlop, amibe írni kell a pixelt
in	<i>*rgb</i>	megadja, hogy ez melyik szín

See also

[allocateimage](#)

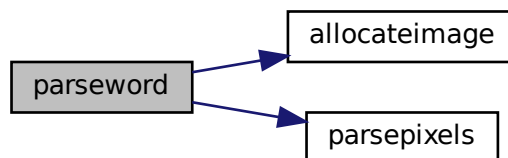
[parsepixels](#)

Definition at line 111 of file ppm.c.

References `allocateimage()`, `PPM_Image::image_data`, `PPM_Image::magic`, `PPM_Image::maxval`, `parsepixels()`, `PPM_Image::size_x`, and `PPM_Image::size_y`.

Referenced by `checktype()`.

Here is the call graph for this function:



5.7.2.6 PPM_Parser()

```
PPM_Image PPM_Parser (
    char filename[] )
```

beolvas egy képet Megnyitja a fájlt, majd létrehozza a [PPM_Image](#) struktúra egy példányát és beállítja a kezdő értékeit. Ezek után maximum 4096 karakter hosszú soronként beolvassa a fájlt és átadja a `checktype` -nak. Végül bezárja a fájlt.

Parameters

in	<code>filename[]</code>	ezt a fájlt fogja megnyitni
----	-------------------------	-----------------------------

See also

[checktype](#)

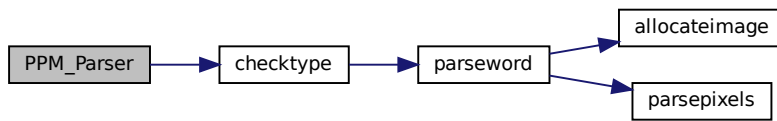
[PPM_Image](#)

Definition at line 176 of file `ppm.c`.

References `checktype()`, `PPM_Image::magic`, `PPM_Image::maxval`, `PPM_Image::size_x`, and `PPM_Image::size_y`.

Referenced by `main()`.

Here is the call graph for this function:



5.7.2.7 PPM_Writer()

```
void PPM_Writer (
    char filename[],
    PPM_Image * image )
```

Fájlba írja a `PPM_Image` tartalmát. A megnyitott fájlba először kiírjuk sorrendben a magic-et, az oszlopok számát, a sorok számát, a maxvalt. Ezek után a pixelek adatait írjuk ki, egy sorba egy pixelt, tehát három számot. Végül bezárjuk a fájlt.

Parameters

in	<code>filename[]</code>	a kimentő fájl neve
in	<code>*image</code>	a kiírandó kép

Definition at line 218 of file ppm.c.

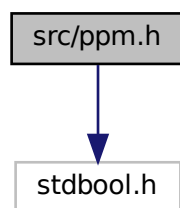
References `PPM_Image::image_data`, `PPM_Image::magic`, `PPM_Image::maxval`, `PPM_Image::size_x`, and `PPM_Image::size_y`.

Referenced by `main()`.

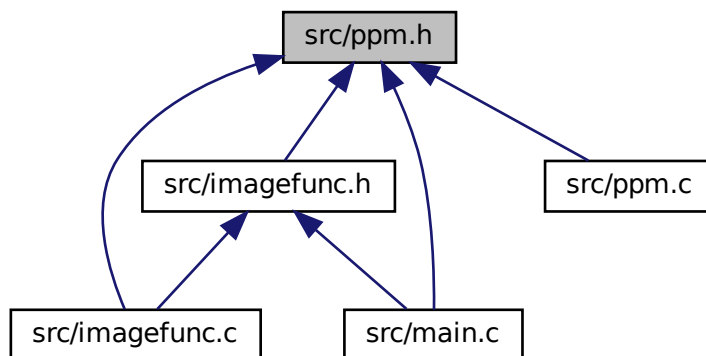
5.8 src/ppm.h File Reference

```
#include <stdbool.h>
```

Include dependency graph for ppm.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [PPM_Image](#)

a PPM fájl tárolására használt struktúra

Typedefs

- typedef struct [PPM_Image](#) [PPM_Image](#)

a PPM fájl tárolására használt struktúra

Functions

- unsigned char [getpixelcolor](#) (unsigned char *image, int x, int y, int z, int size_x)
- void [setpixelcolor](#) (unsigned char *image, int x, int y, int z, int size_x, unsigned char value)
- unsigned char * [allocateimage1d](#) (int size_x, int size_y)
- void [freeimage](#) (unsigned char ***image, int size_x, int size_y)
- unsigned char *** [allocateimage](#) (int size_x, int size_y)
A kép tárolására használt 3 dimenziós tömb lefoglalása.
- bool [parsepixels](#) (char word[], [PPM_Image](#) *image, int *iter_h, int *iter_w, int *rgb)
egy pixel értékeinek beállítása a kapott szó alapján
- void [parseword](#) (char word[], [PPM_Image](#) *image, int *iter_h, int *iter_w, int *rgb)
Eldönti, hogy mit kell kezdeni a kapott szóval. Először azt nézi meg, hogy van-e magic. Ha nincs ezt kell beállítani. Majd sorban a sorszámot, oszlopszámot és a maxvált. Ha megvan az oszlopszám, le lehet foglalni a kép tömbjét. Ezek után a maradék biztosan pixel érték lesz.
- void [checktype](#) (char line[], [PPM_Image](#) *image, int *iter_h, int *iter_w, int *rgb)
Eldönti, hogy a megadott sor mit tartalmaz és mit kell vele csinálni Ha comment akkor kimarad, egyébként szóközönként fel kell bontani. Ezek lesznek a szavak, amiket utána tovább kell kategorizálni.
- [PPM_Image](#) [PPM_Parser](#) (char filename[])
beolvas egy képet Megnyitja a fájlt, majd létrehozza a [PPM_Image](#) struktúra egy példányát és beállítja a kezdő értékeit. Ezek után maximum 4096 karakter hosszú soronként beolvassa a fájlt és átadja a [checktype](#) -nak. Végül bezárja a fájlt.
- void [PPM_Writer](#) (char filename[], [PPM_Image](#) *image)
Fájlbba írja a [PPM_Image](#) tartalmát A megnyitott fájlba először kiírjuk sorrendben a magic-et az oszlopok számát, a sorok számát, a maxvált. Ezek után a pixelek adatait írjuk ki, egy sorba egy pixelt, tehát három számot. Végül bezárjuk a fájlt.

5.8.1 Typedef Documentation

5.8.1.1 PPM_Image

```
typedef struct PPM_Image PPM_Image
```

a PPM fájl tárolására használt struktúra

5.8.2 Function Documentation

5.8.2.1 allocateimage()

```
unsigned char*** allocateimage (
    int size_x,
    int size_y )
```

A kép tárolására használt 3 dimenziós tömb lefoglalása.

Parameters

in	<i>size</i> ↔ <i>_x</i>	a kép oszlopainak száma
in	<i>size</i> ↔ <i>_y</i>	a kép sorainak száma
out	<i>image</i>	a létrehozott 3 dimenziós tömb

Definition at line 35 of file ppm.c.

Referenced by `convolve()`, `detect_edges()`, and `parseword()`.

5.8.2.2 `allocateimage1d()`

```
unsigned char* allocateimage1d (
    int size_x,
    int size_y )
```

5.8.2.3 `checktype()`

```
void checktype (
    char line[],
    PPM_Image * image,
    int * iter_h,
    int * iter_w,
    int * rgb )
```

Eldönti, hogy a megadott sor mit tartalmaz és mit kell vele csinálni. Ha comment akkor kimarad, egyébként szóközőnként fel kell bontani. Ezek lesznek a szavak, amiket utána tovább kell kategorizálni.

Parameters

in	<i>line[]</i>	a sor amit vizsgálni kell
in	* <i>image</i>	PPM_Image kép aminek az <code>image_data</code> részébe kell beírni a pixelt
in	* <i>iter</i> ↔ <i>_h</i>	a sor, amibe írni kell a pixelt
in	* <i>iter</i> ↔ <i>_w</i>	az oszlop, amibe írni kell a pixelt
in	* <i>rgb</i>	megadja, hogy ez melyik szín

See also

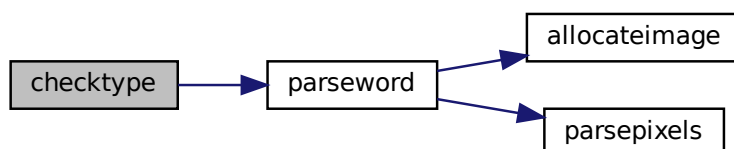
[parseword](#)

Definition at line 150 of file ppm.c.

References parseword().

Referenced by PPM_Parser().

Here is the call graph for this function:



5.8.2.4 freeimage()

```
void freeimage (
    unsigned char *** image,
    int size_x,
    int size_y )
```

A kép tárolására használt 3 dimenziós tömb felszabadítása

Parameters

in	<i>***image</i>	felszabadítandó kép
in	<i>size_x</i>	a kép oszlopainak száma
in	<i>size_y</i>	a kép sorainak száma

Definition at line 19 of file ppm.c.

Referenced by `convolve()`, `main()`, and `pixelsort()`.

5.8.2.5 getpixelcolor()

```
unsigned char getpixelcolor (
    unsigned char * image,
    int x,
    int y,
    int z,
    int size_x )
```

5.8.2.6 parsepixels()

```
bool parsepixels (
    char word[],
    PPM_Image * image,
    int * iter_h,
    int * iter_w,
    int * rgb )
```

egy pixel értékeinek beállítása a kapott szó alapján

A megkapott szót unsigned char-rá alakítja és beírja a képbe. Növeli az aktuális oszlop számát, és ha eléri a sorszélességet az aktuális sor számát is növeli. Az rgb értéket mindig növeli, de ha eléri a 3-at, átírja 0-ra. Ha az aktuális sor értéke megegyezik a kép sorainak számával az azt jelenti, hogy betelt a kép és megszakad a beolvasás.

Parameters

in	<i>word[]</i>	a szó amit pixellé kell átalakítani
in	<i>*image</i>	PPM_Image kép aminek az image_data részébe kell beírni a pixelt
in	<i>*iter_h</i>	a sor, amibe írni kell a pixelt
in	<i>*iter_w</i>	az oszlop, amibe írni kell a pixelt
in	<i>*rgb</i>	megadja, hogy ez melyik szín
out	<i>ongoing</i>	true ha még mindig fér a képbe, false ha betelt

See also

[PPM_Image](#)

Definition at line 78 of file ppm.c.

References `PPM_Image::image_data`, `PPM_Image::maxval`, `PPM_Image::size_x`, and `PPM_Image::size_y`.

Referenced by `parseword()`.

5.8.2.7 parseword()

```
void parseword (
    char word[],
    PPM_Image * image,
    int * iter_h,
    int * iter_w,
    int * rgb )
```

Eldönti, hogy mit kell kezdeni a kapott szóval. Először azt nézi meg, hogy van-e magic. Ha nincs ezt kell beállítani. Majd sorban a sorszámot, oszlopszámot és a maxvalt. Ha megvan az oszlopszám, le lehet foglalni a kép tömbjét. Ezek után a maradék biztosan pixel érték lesz.

Parameters

in	<i>word[]</i>	a szó amit le kell ellenőrizni
in	<i>*image</i>	PPM_Image kép aminek az image_data részébe kell beírni a pixelt
in	<i>*iter_h</i>	a sor, amibe írni kell a pixelt
in	<i>*iter_w</i>	az oszlop, amibe írni kell a pixelt
in	<i>*rgb</i>	megadja, hogy ez melyik szín

See also

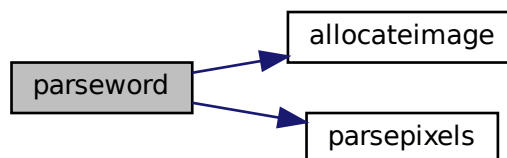
[allocateimage](#)
[parsepixels](#)

Definition at line 111 of file ppm.c.

References [allocateimage\(\)](#), [PPM_Image::image_data](#), [PPM_Image::magic](#), [PPM_Image::maxval](#), [parsepixels\(\)](#), [PPM_Image::size_x](#), and [PPM_Image::size_y](#).

Referenced by [checktype\(\)](#).

Here is the call graph for this function:



5.8.2.8 PPM_Parser()

```
PPM_Image PPM_Parser (
    char filename[] )
```

beolvas egy képet Megnyitja a fájlt, majd létrehozza a [PPM_Image](#) struktúra egy példányát és beállítja a kezdő értékeit. Ezek után maximum 4096 karakter hosszú soronként beolvassa a fájlt és átadja a `checktype` -nak. Végül bezárja a fájlt.

Parameters

in	<code>filename[]</code>	ezt a fájlt fogja megnyitni
----	-------------------------	-----------------------------

See also

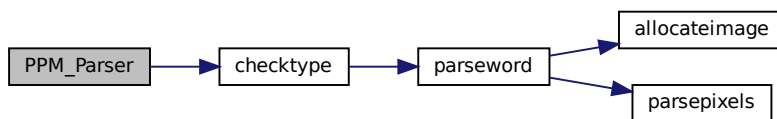
[checktype](#)
[PPM_Image](#)

Definition at line 176 of file ppm.c.

References [checktype\(\)](#), [PPM_Image::magic](#), [PPM_Image::maxval](#), [PPM_Image::size_x](#), and [PPM_Image::size_y](#).

Referenced by [main\(\)](#).

Here is the call graph for this function:



5.8.2.9 PPM_Writer()

```
void PPM_Writer (
    char filename[],
    PPM_Image * image )
```

Fájlba írja a `PPM_Image` tartalmát. A megnyitott fájlba először kiírjuk sorrendben a magic-et, az oszlopok számát, a sorok számát, a maxval-t. Ezek után a pixelek adatait írjuk ki, egy sorba egy pixelt, tehát három számot. Végül bezárjuk a fájlt.

Parameters

in	<code>filename[]</code>	a kimentő fájl neve
in	<code>*image</code>	a kiírandó kép

Definition at line 218 of file `ppm.c`.

References `PPM_Image::image_data`, `PPM_Image::magic`, `PPM_Image::maxval`, `PPM_Image::size_x`, and `PPM_Image::size_y`.

Referenced by `main()`.

5.8.2.10 setpixelcolor()

```
void setpixelcolor (
    unsigned char * image,
    int x,
    int y,
    int z,
    int size_x,
    unsigned char value )
```


Index

- addmath.c
 - clamp, [20](#)
 - max, [20](#)
 - min, [20](#)
 - partition, [21](#)
 - quicksort, [21](#)
- addmath.h
 - clamp, [23](#)
 - max, [24](#)
 - min, [24](#)
 - partition, [24](#)
 - quicksort, [25](#)
 - Sort, [23](#)
- allocatefilter
 - imagefunc.c, [28](#)
 - imagefunc.h, [53](#)
- allocateimage
 - ppm.c, [73](#)
 - ppm.h, [80](#)
- allocateimage1d
 - ppm.h, [81](#)
- allrandom
 - imagefunc.h, [52](#)
- anaglyph3d
 - imagefunc.c, [28](#)
 - imagefunc.h, [54](#)
- b
 - RGB, [15](#)
- blue_x
 - RGB_SHIFT, [16](#)
- blue_y
 - RGB_SHIFT, [16](#)
- change_light
 - imagefunc.c, [29](#)
 - imagefunc.h, [54](#)
- checkinterval
 - imagefunc.c, [30](#)
- checktreshhold_bottom
 - imagefunc.c, [30](#)
- checktreshhold_top
 - imagefunc.c, [31](#)
- checktype
 - ppm.c, [74](#)
 - ppm.h, [81](#)
- clamp
 - addmath.c, [20](#)
 - addmath.h, [23](#)
- contrast
 - imagefunc.c, [32](#)
 - imagefunc.h, [55](#)
- convolve
 - imagefunc.c, [32](#)
 - imagefunc.h, [56](#)
- corrupt
 - imagefunc.c, [33](#)
 - imagefunc.h, [57](#)
- dark
 - imagefunc.h, [52](#)
- detect_edges
 - imagefunc.c, [34](#)
 - imagefunc.h, [58](#)
- diagonal
 - imagefunc.h, [52](#)
- edge
 - imagefunc.h, [52](#)
- edges
 - imagefunc.h, [53](#)
- fewcolors
 - imagefunc.h, [52](#)
- filt
 - Filter, [7](#)
- Filter, [7](#)
 - filt, [7](#)
 - imagefunc.h, [50](#)
 - mult, [7](#)
 - size_x, [8](#)
 - size_y, [8](#)
- freefilter
 - imagefunc.c, [35](#)
 - imagefunc.h, [59](#)
- freeimage
 - ppm.c, [75](#)
 - ppm.h, [82](#)
- g
 - RGB, [15](#)
- getpixelcolor
 - ppm.h, [82](#)
- grayscale
 - imagefunc.c, [36](#)
 - imagefunc.h, [60](#)
- green_x
 - RGB_SHIFT, [17](#)
- green_y
 - RGB_SHIFT, [17](#)

- h
 - HSL, 9
- horizontal
 - imagefunc.h, 52
- HSL, 8
 - h, 9
 - imagefunc.h, 50
 - l, 9
 - s, 9
- hsl2rgb
 - imagefunc.c, 36
 - imagefunc.h, 60
- hsl2rgbcolor
 - imagefunc.c, 37
 - imagefunc.h, 61
- hsl_l
 - imagefunc.h, 53
- hue_shift
 - imagefunc.c, 37
 - imagefunc.h, 61
- idx
 - Sort, 18
- image_data
 - PPM_Image, 10
- imagefunc.c
 - allocatefilter, 28
 - anaglyph3d, 28
 - change_light, 29
 - checkinterval, 30
 - checktreshold_bottom, 30
 - checktreshold_top, 31
 - contrast, 32
 - convolve, 32
 - corrupt, 33
 - detect_edges, 34
 - freefilter, 35
 - grayscale, 36
 - hsl2rgb, 36
 - hsl2rgbcolor, 37
 - hue_shift, 37
 - invert, 38
 - mirror_diagonal, 38
 - mirror_horizontal, 39
 - mirror_vertical, 39
 - pixelsort, 40
 - rgb2hsl, 41
 - rgb_shift, 42
 - rotate, 43
 - rotate_vertical, 43
 - set_white, 44
 - setfilter, 44
 - sharp_grayscale, 45
 - sinecolor_shift, 46
 - sortcopy, 46
- imagefunc.h
 - allocatefilter, 53
 - allrandom, 52
 - anaglyph3d, 54
 - change_light, 54
 - contrast, 55
 - convolve, 56
 - corrupt, 57
 - dark, 52
 - detect_edges, 58
 - diagonal, 52
 - edge, 52
 - edges, 53
 - fewcolors, 52
 - Filter, 50
 - freefilter, 59
 - grayscale, 60
 - horizontal, 52
 - HSL, 50
 - hsl2rgb, 60
 - hsl2rgbcolor, 61
 - hsl_l, 53
 - hue_shift, 61
 - invert, 62
 - landscape, 52
 - macro, 52
 - man, 53
 - mirror_diagonal, 62
 - mirror_horizontal, 63
 - mirror_type, 50, 52
 - mirror_vertical, 63
 - none, 52
 - pixelsort, 64
 - pixelsort_preset, 51, 52
 - ps_option_type, 51, 52
 - ps_type, 51, 53
 - psnone, 52
 - PsOptions, 51
 - ran, 53
 - RGB, 51
 - rgb2hsl, 65
 - RGB_SHIFT, 51
 - rgb_shift, 66
 - rgb_sum, 53
 - set_black, 67
 - setfilter, 67
 - sharp_grayscale, 68
 - sinecolor_shift, 68
 - sortcopy, 69
 - vertical, 52
- interval
 - PsOptions, 12
- interval_max
 - PsOptions, 12
- interval_min
 - PsOptions, 12
- invert
 - imagefunc.c, 38
 - imagefunc.h, 62
- l
 - HSL, 9
- landscape

- imagefunc.h, [52](#)
- macro
 - imagefunc.h, [52](#)
- magic
 - PPM_Image, [10](#)
- main
 - main.c, [71](#)
- main.c
 - main, [71](#)
- mainpage.md, [19](#)
- man
 - imagefunc.h, [53](#)
- max
 - addmath.c, [20](#)
 - addmath.h, [24](#)
- maxval
 - PPM_Image, [10](#)
- merge
 - PsOptions, [13](#)
- min
 - addmath.c, [20](#)
 - addmath.h, [24](#)
- mirror_diagonal
 - imagefunc.c, [38](#)
 - imagefunc.h, [62](#)
- mirror_horizontal
 - imagefunc.c, [39](#)
 - imagefunc.h, [63](#)
- mirror_type
 - imagefunc.h, [50](#), [52](#)
- mirror_vertical
 - imagefunc.c, [39](#)
 - imagefunc.h, [63](#)
- mult
 - Filter, [7](#)
- none
 - imagefunc.h, [52](#)
- parsepixels
 - ppm.c, [75](#)
 - ppm.h, [82](#)
- parseword
 - ppm.c, [76](#)
 - ppm.h, [83](#)
- partition
 - addmath.c, [21](#)
 - addmath.h, [24](#)
- pixelsort
 - imagefunc.c, [40](#)
 - imagefunc.h, [64](#)
- pixelsort_preset
 - imagefunc.h, [51](#), [52](#)
- ppm.c
 - allocateimage, [73](#)
 - checktype, [74](#)
 - freeimage, [75](#)
 - parsepixels, [75](#)
 - parseword, [76](#)
 - PPM_Parser, [77](#)
 - PPM_Writer, [78](#)
- ppm.h
 - allocateimage, [80](#)
 - allocateimage1d, [81](#)
 - checktype, [81](#)
 - freeimage, [82](#)
 - getpixelcolor, [82](#)
 - parsepixels, [82](#)
 - parseword, [83](#)
 - PPM_Image, [80](#)
 - PPM_Parser, [84](#)
 - PPM_Writer, [85](#)
 - setpixelcolor, [85](#)
- PPM_Image, [10](#)
 - image_data, [10](#)
 - magic, [10](#)
 - maxval, [10](#)
 - ppm.h, [80](#)
 - size_x, [11](#)
 - size_y, [11](#)
- PPM_Parser
 - ppm.c, [77](#)
 - ppm.h, [84](#)
- PPM_Writer
 - ppm.c, [78](#)
 - ppm.h, [85](#)
- ps_option_type
 - imagefunc.h, [51](#), [52](#)
- ps_type
 - imagefunc.h, [51](#), [53](#)
- psnone
 - imagefunc.h, [52](#)
- PsOptions, [11](#)
 - imagefunc.h, [51](#)
 - interval, [12](#)
 - interval_max, [12](#)
 - interval_min, [12](#)
 - merge, [13](#)
 - pstype, [13](#)
 - treshold, [13](#)
 - treshold_bottom_max, [13](#)
 - treshold_bottom_min, [14](#)
 - treshold_top_max, [14](#)
 - treshold_top_min, [14](#)
- pstype
 - PsOptions, [13](#)
- quicksort
 - addmath.c, [21](#)
 - addmath.h, [25](#)
- r
 - RGB, [15](#)
- ran
 - imagefunc.h, [53](#)
- red_x
 - RGB_SHIFT, [17](#)

- red_y
 - RGB_SHIFT, 17
- RGB, 15
 - b, 15
 - g, 15
 - imagefunc.h, 51
 - r, 15
- rgb2hsl
 - imagefunc.c, 41
 - imagefunc.h, 65
- RGB_SHIFT, 16
 - blue_x, 16
 - blue_y, 16
 - green_x, 17
 - green_y, 17
 - imagefunc.h, 51
 - red_x, 17
 - red_y, 17
- rgb_shift
 - imagefunc.c, 42
 - imagefunc.h, 66
- rgb_sum
 - imagefunc.h, 53
- rotate
 - imagefunc.c, 43
- rotate_vertical
 - imagefunc.c, 43
- s
 - HSL, 9
- set_black
 - imagefunc.h, 67
- set_white
 - imagefunc.c, 44
- setfilter
 - imagefunc.c, 44
 - imagefunc.h, 67
- setpixelcolor
 - ppm.h, 85
- sharp_grayscale
 - imagefunc.c, 45
 - imagefunc.h, 68
- sinecolor_shift
 - imagefunc.c, 46
 - imagefunc.h, 68
- size_x
 - Filter, 8
 - PPM_Image, 11
- size_y
 - Filter, 8
 - PPM_Image, 11
- Sort, 18
 - addmath.h, 23
 - idx, 18
 - value, 18
- sortcopy
 - imagefunc.c, 46
 - imagefunc.h, 69
- src/addmath.c, 19
 - src/addmath.h, 22
 - src/imagefunc.c, 26
 - src/imagefunc.h, 47
 - src/main.c, 70
 - src/ppm.c, 72
 - src/ppm.h, 78
- treshold
 - PsOptions, 13
- treshold_bottom_max
 - PsOptions, 13
- treshold_bottom_min
 - PsOptions, 14
- treshold_top_max
 - PsOptions, 14
- treshold_top_min
 - PsOptions, 14
- value
 - Sort, 18
- vertical
 - imagefunc.h, 52