

# Backgroundmodel

0.0

Generated by Doxygen 1.7.1

Thu Mar 29 2012 23:05:50



# Contents

<b>1</b>	<b>This is my own main page</b>	<b>1</b>
1.1	Introduction . . . . .	1
<b>2</b>	<b>Module Index</b>	<b>3</b>
2.1	Modules . . . . .	3
<b>3</b>	<b>Data Structure Index</b>	<b>5</b>
3.1	Data Structures . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Module Documentation</b>	<b>9</b>
5.1	Background model functions . . . . .	9
5.1.1	Function Documentation . . . . .	9
5.1.1.1	bg_model_acquire . . . . .	9
5.1.1.2	bg_model_clear . . . . .	10
5.1.1.3	bg_model_free . . . . .	10
5.1.1.4	bg_model_get_colorspace . . . . .	10
5.1.1.5	bg_model_get_history_size . . . . .	10
5.1.1.6	bg_model_history_add . . . . .	10
5.1.1.7	bg_model_history_reset . . . . .	11
5.1.1.8	bg_model_init . . . . .	11
5.1.1.9	bg_model_new . . . . .	11
5.1.1.10	bg_model_pixel_is_background . . . . .	11
5.1.1.11	bg_model_set_colorspace . . . . .	11
5.1.1.12	bg_model_set_history_size . . . . .	12
5.1.1.13	bg_model_set_parameters . . . . .	12
5.1.1.14	bg_model_update . . . . .	12
5.1.1.15	free_circle_buff . . . . .	12

---

5.1.1.16	<a href="#">init_circle_buff</a>	13
5.1.1.17	<a href="#">new_circle_buff</a>	13
5.1.1.18	<a href="#">reset_circle_buff</a>	13
5.1.1.19	<a href="#">set_circle_buff_data_size</a>	13
5.1.1.20	<a href="#">write_circle_buff</a>	13
<b>6</b>	<b>Data Structure Documentation</b>	<b>15</b>
6.1	<a href="#">bg_commands Struct Reference</a>	15
6.2	<a href="#">bg_model Struct Reference</a>	15
6.2.1	Field Documentation	15
6.2.1.1	<a href="#">bg_model_ctx</a>	15
6.2.1.2	<a href="#">buff</a>	15
6.2.1.3	<a href="#">colorspace</a>	15
6.2.1.4	<a href="#">model_type</a>	16
6.3	<a href="#">circle_buff Struct Reference</a>	16
6.3.1	Field Documentation	16
6.3.1.1	<a href="#">buff</a>	16
6.3.1.2	<a href="#">buff_size</a>	16
6.3.1.3	<a href="#">dim</a>	16
6.3.1.4	<a href="#">head</a>	16
6.3.1.5	<a href="#">tail</a>	16
<b>7</b>	<b>File Documentation</b>	<b>17</b>
7.1	<a href="#">bg_model.h File Reference</a>	17
7.1.1	Detailed Description	18
7.1.2	Enumeration Type Documentation	18
7.1.2.1	<a href="#">bg_colorspace</a>	18
7.1.2.2	<a href="#">bg_model_type</a>	18
7.1.2.3	<a href="#">bg_result</a>	18
7.1.3	Function Documentation	18
7.1.3.1	<a href="#">bg_model_reset</a>	18
7.1.3.2	<a href="#">clear_circle_buff</a>	19

# Chapter 1

## This is my own main page

### 1.1 Introduction

Single background pixel model API.

This API allows the user to determine, be provided with the RGB values, if the pixel belongs to the background or not.



# Chapter 2

## Module Index

### 2.1 Modules

Here is a list of all modules:

Background model functions . . . . .	9
--------------------------------------	---





# Chapter 3

## Data Structure Index

### 3.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">bg_commands</a>	15
<a href="#">bg_model</a>	15
<a href="#">circle_buff</a>	16



# Chapter 4

## File Index

### 4.1 File List

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

<a href="#">bg_model.h</a> . . . . .	17
--------------------------------------	----



# Chapter 5

## Module Documentation

### 5.1 Background model functions

#### Functions

- struct `circle_buff` \* `new_circle_buff` ()
- void `set_circle_buff_data_size` (int dim, int ndata, struct `circle_buff` \*c\_buff)
- void `init_circle_buff` (struct `circle_buff` \*c\_buff)
- void `write_circle_buff` (struct `circle_buff` \*c\_buff, const double \*data)
- void `reset_circle_buff` (struct `circle_buff` \*c\_buff)
- void `free_circle_buff` (struct `circle_buff` \*\*c\_buff)
- struct `bg_model` \* `bg_model_new` (enum `bg_model_type` model)
- void `bg_model_init` (struct `bg_model` \*bg)
- void `bg_model_clear` (struct `bg_model` \*bg)
- void `bg_model_free` (struct `bg_model` \*\*bg)
- void `bg_model_set_colorspace` (struct `bg_model` \*bg, enum `bg_colorspace` colorspace)
- int `bg_model_get_colorspace` (struct `bg_model` \*bg)
- void `bg_model_set_parameters` (struct `bg_model` \*bg, const char \*param)
- void `bg_model_set_history_size` (struct `bg_model` \*bg, int size)
- int `bg_model_get_history_size` (struct `bg_model` \*bg)
- void `bg_model_history_add` (struct `bg_model` \*bg, int x, int y, int z)
- void `bg_model_history_reset` (struct `bg_model` \*bg)
- void `bg_model_acquire` (struct `bg_model` \*bg)
- void `bg_model_update` (struct `bg_model` \*bg, int x, int y, int z)
- int `bg_model_pixel_is_background` (struct `bg_model` \*bg, int x, int y, int z)

#### 5.1.1 Function Documentation

##### 5.1.1.1 void `bg_model_acquire` ( struct `bg_model` \* *bg* )

Acquire background model for the single pixel using it's collected historical values.

#### Parameters

[in] *bg* pointer to struct `bg_model`

**5.1.1.2 void bg\_model\_clear ( struct bg\_model \* *bg* )**

clear fields of struct [bg\\_model](#)

**Parameters**

[in] *bg* pointer to a struct [bg\\_model](#)

**Note**

if define DEBUG in this function done check that the *bg* is valid

**5.1.1.3 void bg\_model\_free ( struct bg\_model \*\* *bg* )**

free memory allocated from struct [bg\\_model](#)

**Parameters**

[in] *bg* pointer to pointer to struct [bg\\_model](#)

**Note**

if define DEBUG in this function done check that the *bg* is valid

**5.1.1.4 int bg\_model\_get\_colorspace ( struct bg\_model \* *bg* )**

Get colorspace. The only colorspace supported is BG\_COLORSPACE\_RGB.

**Parameters**

[in] *bg* pointer to struct [bg\\_model](#)

**Returns**

pixel colorspace

**5.1.1.5 int bg\_model\_get\_history\_size ( struct bg\_model \* *bg* )**

Get depth of the pixel value history cyclic buffer.

**Parameters**

[in] *bg* pointer to struct [bg\\_model](#)

**Returns**

buffer size

**5.1.1.6 void bg\_model\_history\_add ( struct bg\_model \* *bg*, int *x*, int *y*, int *z* )**

Update the history with a new pixel value. Arguments [*x,y,z*] stay for [R,G,B] values in BG\_COLORSPACE\_RGB.

**Parameters**

[in] *bg* pointer to struct [bg\\_model](#)

**5.1.1.7 void bg\_model\_history\_reset ( struct bg\_model \* *bg* )**

Reset the history and forget all previous pixel values.

**Parameters**

[in] *bg* pointer to struct [bg\\_model](#)

**5.1.1.8 void bg\_model\_init ( struct bg\_model \* *bg* )**

initialize struct [bg\\_model](#) and allocated memory for fields of struct

**Parameters**

[in] *bg* pointer to a struct [bg\\_model](#)

**Note**

if define DEBUG in this function done check that the *bg* is valid

**5.1.1.9 struct bg\_model\* bg\_model\_new ( enum bg\_model\_type *model* ) [read]****Parameters**

[in] *model* - background model type

**Returns**

pointer to struct [bg\\_model](#) and set type of model if OK, else NULL

**5.1.1.10 int bg\_model\_pixel\_is\_background ( struct bg\_model \* *bg*, int *x*, int *y*, int *z* )**

Determine, according to current background model, if a pixel [*x,y,z*] represents background or not. Arguments [*x,y,z*] stay for [R,G,B] values in BG\_COLORSPACE\_RGB

**Returns**

BG\_OK if pixel in model, else BG\_ERROR

**5.1.1.11 void bg\_model\_set\_colorspace ( struct bg\_model \* *bg*, enum bg\_colorspace *colorspace* )**

Set colorspace. The only colorspace supported is BG\_COLORSPACE\_RGB.

**Parameters**

[in] *bg* pointer to struct [bg\\_model](#)

[in] *colorspace* pixel colorspace

#### 5.1.1.12 void `bg_model_set_history_size` ( struct `bg_model` \* `bg`, int `size` )

Set depth of the pixel value history cyclic buffer.

##### Parameters

[in] `bg` pointer to struct `bg_model`

[in] `size` buffer size

#### 5.1.1.13 void `bg_model_set_parameters` ( struct `bg_model` \* `bg`, const char \* `param` )

Set underlying model parameters using param string.

##### Parameters

[in] `bg` pointer to struct `bg_model`

[in] `param` command for set model param

##### Note

param can be "ngaussian %d"  
"nth mean %d %f, ..., %f",  
"nth weight %d %f",  
"nth matrix %d %f, ..., %f",  
"epsilon %f"  
"nmean %d",  
"nth mean %d %f, ..., %f"

#### 5.1.1.14 void `bg_model_update` ( struct `bg_model` \* `bg`, int `x`, int `y`, int `z` )

Update background model using a new pixel value. Arguments [x,y,z] stay for [R,G,B] values in BG\_COLORSPACE\_RGB.

##### Parameters

[in] `bg` pointer to struct `bg_model`

#### 5.1.1.15 void `free_circle_buff` ( struct `circle_buff` \*\* `c_buff` )

free memory allocated for struct `circle_buff` and set `c_buff` to NULL

##### Parameters

[in] `c_buff` pointer to pointer to struct `circle_buff`

##### Note

if define DEBUG in this function done check that the `c_buff` is valid



**5.1.1.16 void init\_circle\_buff ( struct circle\_buff \* c\_buff )**

initialize struct circle buff and allocates memory for struct fields

**Parameters**

[in] *c\_buff* pointer to struct [circle\\_buff](#)

**Note**

if define DEBUG in this function done check that the *c\_buff* is valid

**5.1.1.17 struct circle\_buff\* new\_circle\_buff ( ) [read]**

*new\_circle\_buff*

**Returns**

pointer to struct [circle\\_buff](#) if OK, else return NULL

**5.1.1.18 void reset\_circle\_buff ( struct circle\_buff \* c\_buff )**

*reset\_circle\_buff* set the tile and head of struct [circle\\_buff](#) to zero

**Parameters**

[in] *c\_buff* pointer to struct [circle\\_buff](#)

**Note**

if define DEBUG in this function done check that the *c\_buff* is valid

**5.1.1.19 void set\_circle\_buff\_data\_size ( int dim, int ndata, struct circle\_buff \* c\_buff )**

*set\_circle\_buff\_data\_size* set data dimension and buffer size from *c\_buff*

**Parameters**

[in] *dim* data dimension

[in] *ndata* circle buffer size

[in] *c\_buff* pointer to struct [circle\\_buff](#)

**Note**

if define DEBUG in this function done check that the *c\_buff* is valid

**5.1.1.20 void write\_circle\_buff ( struct circle\_buff \* c\_buff, const double \* data )**

*write\_circle\_buff* write data to struct [circle\\_buff](#)

**Parameters**

[in] *c\_buff* pointer to struct [circle\\_buff](#)

[in] *data* pointer to samples

**Note**

if define DEBUG in this function done check that the *c\_buff* and *data* is valid

# Chapter 6

## Data Structure Documentation

### 6.1 `bg_commands` Struct Reference

#### Data Fields

- `const char * bg_cmd_name`
- `int len`

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

- [bg\\_model.h](#)

### 6.2 `bg_model` Struct Reference

#### Data Fields

- enum [bg\\_colorspace](#) `colorspace`
- enum [bg\\_model\\_type](#) `model_type`
- `void * bg_model_ctx`
- struct [circle\\_buff](#) \* `buff`

#### 6.2.1 Field Documentation

##### 6.2.1.1 `void* bg_model::bg_model_ctx`

pointer to background model context

##### 6.2.1.2 `struct circle_buff* bg_model::buff`

buffer for data storage

##### 6.2.1.3 `enum bg_colorspace bg_model::colorspace`

type of modeling pixels

#### 6.2.1.4 enum `bg_model_type` `bg_model::model_type`

model type

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

- [bg\\_model.h](#)

## 6.3 `circle_buff` Struct Reference

### Data Fields

- double \* `buff`
- int `head`
- int `tail`
- int `buff_size`
- int `dim`

### 6.3.1 Field Documentation

#### 6.3.1.1 `double*` `circle_buff::buff`

buffer for storage data

#### 6.3.1.2 `int` `circle_buff::buff_size`

`buff_size` size of buffer

#### 6.3.1.3 `int` `circle_buff::dim`

`dim` dimation of data

#### 6.3.1.4 `int` `circle_buff::head`

number write element

#### 6.3.1.5 `int` `circle_buff::tail`

number recorded of elements

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

- [bg\\_model.h](#)

# Chapter 7

## File Documentation

### 7.1 `bg_model.h` File Reference

#### Data Structures

- struct `bg_commands`
- struct `circle_buff`
- struct `bg_model`

#### Defines

- `#define NUM_BG_EM_COMMAND 5`
- `#define NUM_BG_K_MEANS_COMMAND 2`

#### Enumerations

- enum `bg_colorspace` { `BG_COLORSPACE_RGB` }
- enum `bg_model_type` { `BG_MODEL_MOG`, `BG_MODEL_K_MEAN` }
- enum `bg_result` { `BG_ERROR = 1`, `BG_OK` }

#### Functions

- struct `circle_buff` \* `new_circle_buff` ()
- void `set_circle_buff_data_size` (int dim, int ndata, struct `circle_buff` \*c\_buff)
- void `init_circle_buff` (struct `circle_buff` \*c\_buff)
- void `clear_circle_buff` (struct `circle_buff` \*c\_buff)
- void `write_circle_buff` (struct `circle_buff` \*c\_buff, const double \*data)
- void `reset_circle_buff` (struct `circle_buff` \*c\_buff)
- void `free_circle_buff` (struct `circle_buff` \*\*c\_buff)
- struct `bg_model` \* `bg_model_new` (enum `bg_model_type` model)
- void `bg_model_init` (struct `bg_model` \*bg)
- void `bg_model_clear` (struct `bg_model` \*bg)
- void `bg_model_free` (struct `bg_model` \*\*bg)
- void `bg_model_set_colorspace` (struct `bg_model` \*bg, enum `bg_colorspace` colorspace)

- int [bg\\_model\\_get\\_colorspace](#) (struct [bg\\_model](#) \*bg)
- void [bg\\_model\\_set\\_parameters](#) (struct [bg\\_model](#) \*bg, const char \*param)
- void [bg\\_model\\_set\\_history\\_size](#) (struct [bg\\_model](#) \*bg, int size)
- int [bg\\_model\\_get\\_history\\_size](#) (struct [bg\\_model](#) \*bg)
- void [bg\\_model\\_history\\_add](#) (struct [bg\\_model](#) \*bg, int x, int y, int z)
- void [bg\\_model\\_history\\_reset](#) (struct [bg\\_model](#) \*bg)
- void [bg\\_model\\_acquire](#) (struct [bg\\_model](#) \*bg)
- void [bg\\_model\\_update](#) (struct [bg\\_model](#) \*bg, int x, int y, int z)
- void [bg\\_model\\_reset](#) (struct [bg\\_model](#) \*bg)
- int [bg\\_model\\_pixel\\_is\\_background](#) (struct [bg\\_model](#) \*bg, int x, int y, int z)

### 7.1.1 Detailed Description

A brief file description.

More descriptive text goes here.

### 7.1.2 Enumeration Type Documentation

#### 7.1.2.1 enum [bg\\_colorspace](#)

type of modeling pixels

#### 7.1.2.2 enum [bg\\_model\\_type](#)

model type

#### Enumerator:

***BG\_MODEL\_MOG*** Mixture of Gaussians model

***BG\_MODEL\_K\_MEAN*** K means model

#### 7.1.2.3 enum [bg\\_result](#)

API functions returned values

### 7.1.3 Function Documentation

#### 7.1.3.1 void [bg\\_model\\_reset](#) ( struct [bg\\_model](#) \* *bg* )

Reset the model to initial state. This resets underlying mathematical model to initial state too.

#### Parameters

[in] *bg* pointer to struct [bg\\_model](#)

**7.1.3.2 void clear\_circle\_buff ( struct circle\_buff \* *c\_buff* )**

clear\_circle\_buff clear fields in struct circle buff and set data dimension and size to zero

**Parameters**

[in] *c\_buff* pointer to struct [circle\\_buff](#)

**Note**

if define DEBUG in this function done check that the *c\_buff* is valid

# Index

- Background model functions, 9
- bg\_model.h
  - BG\_MODEL\_K\_MEAN, 18
  - BG\_MODEL\_MOG, 18
- BG\_MODEL\_K\_MEAN
  - bg\_model.h, 18
- BG\_MODEL\_MOG
  - bg\_model.h, 18
- bg\_colorspace
  - bg\_model.h, 18
- bg\_commands, 15
- bg\_model, 15
  - bg\_model\_acquire, 9
  - bg\_model\_clear, 9
  - bg\_model\_ctx, 15
  - bg\_model\_free, 10
  - bg\_model\_get\_colorspace, 10
  - bg\_model\_get\_history\_size, 10
  - bg\_model\_history\_add, 10
  - bg\_model\_history\_reset, 10
  - bg\_model\_init, 11
  - bg\_model\_new, 11
  - bg\_model\_pixel\_is\_background, 11
  - bg\_model\_set\_colorspace, 11
  - bg\_model\_set\_history\_size, 11
  - bg\_model\_set\_parameters, 12
  - bg\_model\_update, 12
  - buff, 15
  - colorspace, 15
  - free\_circle\_buff, 12
  - init\_circle\_buff, 12
  - model\_type, 15
  - new\_circle\_buff, 13
  - reset\_circle\_buff, 13
  - set\_circle\_buff\_data\_size, 13
  - write\_circle\_buff, 13
- bg\_model.h, 17
  - bg\_colorspace, 18
  - bg\_model\_reset, 18
  - bg\_model\_type, 18
  - bg\_result, 18
  - clear\_circle\_buff, 18
- bg\_model\_acquire
  - bg\_model, 9
- bg\_model\_clear
  - bg\_model, 9
- bg\_model\_ctx
  - bg\_model, 15
- bg\_model\_free
  - bg\_model, 10
- bg\_model\_get\_colorspace
  - bg\_model, 10
- bg\_model\_get\_history\_size
  - bg\_model, 10
- bg\_model\_history\_add
  - bg\_model, 10
- bg\_model\_history\_reset
  - bg\_model, 10
- bg\_model\_init
  - bg\_model, 11
- bg\_model\_new
  - bg\_model, 11
- bg\_model\_pixel\_is\_background
  - bg\_model, 11
- bg\_model\_reset
  - bg\_model.h, 18
- bg\_model\_set\_colorspace
  - bg\_model, 11
- bg\_model\_set\_history\_size
  - bg\_model, 11
- bg\_model\_set\_parameters
  - bg\_model, 12
- bg\_model\_type
  - bg\_model.h, 18
- bg\_model\_update
  - bg\_model, 12
- bg\_result
  - bg\_model.h, 18
- buff
  - bg\_model, 15
  - circle\_buff, 16
- buff\_size
  - circle\_buff, 16
- circle\_buff, 16
  - buff, 16
  - buff\_size, 16
  - dim, 16
  - head, 16
  - tail, 16



---

clear\_circle\_buff  
    bg\_model.h, 18

colorspace  
    bg\_model, 15

dim  
    circle\_buff, 16

free\_circle\_buff  
    bg\_model, 12

head  
    circle\_buff, 16

init\_circle\_buff  
    bg\_model, 12

model\_type  
    bg\_model, 15

new\_circle\_buff  
    bg\_model, 13

reset\_circle\_buff  
    bg\_model, 13

set\_circle\_buff\_data\_size  
    bg\_model, 13

tail  
    circle\_buff, 16

write\_circle\_buff  
    bg\_model, 13