

Inversion

Generated by Doxygen 1.8.6

Thu Feb 11 2016 01:49:27

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

color_table	Structure permettant de stocker les données d'une table de couleur	??
cutting_plane	Structure permettant de stocker les données du plan de coupe	??
image	??
kdtree	Structure permettant de stocker les données des nœuds de l'arbre	??
point	??

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/ arbre.h	
Structures et headers des fonctions pour les kd arbres	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/ classe.h	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/ exit_if.h	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/ image.h	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/ move_type.h	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/ objet.h	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/ point.h	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/ table.h	
Structures et headers des fonctions pour les tables de couleurs	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/ type_obj.h	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/ arbre.c	
Fonctions pour les kd arbres	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/ graphdata.c	
Sauvegarde les temps d execution pour l inversion d une image par methode triviale et kd arbre	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/ image.c	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/ inversion.c	
Inversion d une image par une table de couleur en un kd arbre	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/ table.c	
Fonctions pour les tables de couleurs	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/ test_arbre.c	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/ test_table.c	
Test des differents fonctions de table.c	??
/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/ trivial.c	
Inversion d une image par une table de couleur en utilisant une methode triviale	??

Chapter 3

Data Structure Documentation

3.1 color_table Struct Reference

structure permettant de stocker les données d'une table de couleur

```
#include <table.h>
```

Data Fields

- [color** table](#)
- [boolean owner](#)
- [int count](#)

3.1.1 Detailed Description

structure permettant de stocker les données d'une table de couleur

3.1.2 Field Documentation

3.1.2.1 [int count](#)

3.1.2.2 [boolean owner](#)

3.1.2.3 [color** table](#)

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

- [/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/table.h](#)

3.2 cutting_plane Struct Reference

structure permettant de stocker les données du plan de coupe

```
#include <arbre.h>
```

Data Fields

- [int position](#)

- int [value](#)

3.2.1 Detailed Description

structure permettant de stocker les donnees du plan de coupe

3.2.2 Field Documentation

3.2.2.1 int position

3.2.2.2 int value

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

- [/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/arbre.h](#)

3.3 image Struct Reference

Data Fields

- [Comp_rgb](#) * [debut](#)
- [Comp_rgb](#) * [courant](#)
- int [hauteur](#)
- int [largeur](#)
- int [longueur](#)
- long [taille](#)
- int [dim](#)
- [PFdist](#) [distance](#)

3.3.1 Field Documentation

3.3.1.1 [Comp_rgb](#)* [courant](#)

3.3.1.2 [Comp_rgb](#)* [debut](#)

3.3.1.3 int [dim](#)

3.3.1.4 [PFdist](#) [distance](#)

3.3.1.5 int [hauteur](#)

3.3.1.6 int [largeur](#)

3.3.1.7 int [longueur](#)

3.3.1.8 long [taille](#)

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

- [/media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/image.c](#)

3.4 kdtree Struct Reference

structure permettant de stocker les donnees des noeuds de l arbre

```
#include <arbre.h>
```

Data Fields

- [color_table](#) colorTable
- struct [kdtree](#) * [left_son](#)
- struct [kdtree](#) * [right_son](#)
- [axis](#) sortAxis
- [cutting_plane](#) cuttingPlane

3.4.1 Detailed Description

structure permettant de stocker les donnees des noeuds de l arbre

3.4.2 Field Documentation

3.4.2.1 [color_table](#) colorTable

3.4.2.2 [cutting_plane](#) cuttingPlane

3.4.2.3 struct [kdtree](#)* [left_son](#)

3.4.2.4 struct [kdtree](#) * [right_son](#)

3.4.2.5 [axis](#) sortAxis

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

- /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/[arbre.h](#)

3.5 point Struct Reference

```
#include <point.h>
```

Data Fields

- int [coordx](#)
- int [coordy](#)

3.5.1 Field Documentation

3.5.1.1 int [coordx](#)

3.5.1.2 int [coordy](#)

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

- /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/[point.h](#)

Chapter 4

File Documentation

4.1 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/arbre.h File Reference

Structures et headers des fonctions pour les kd arbres.

```
#include "table.h"
```

Data Structures

- struct [cutting_plane](#)
structure permettant de stocker les donnees du plan de coupe
- struct [kdtree](#)
structure permettant de stocker les donnees des noeuds de l arbre

Typedefs

- typedef struct [cutting_plane](#) * [cutting_plane](#)
pointeur de structure [cutting_plane](#)
- typedef struct [kdtree](#) * [kdtree](#)
pointeur de structure [kdtree](#)

Functions

create_kdtree

La fonction cree le kd arbre de tri

Parameters

cTable	<i>table de couleur a decouper en kd arbre</i>
seuil	<i>le nombre minimum de couleur dans une feuille de l arbre</i>

Returns

kdtree pointer sur la racine de l arbre

- [kdtree](#) [create_kdtree](#) ([color_table](#), int)

destroy_kdtree

La fonction supprime le kd arbre et ses fils

Parameters

tree	<i>l arbre a detruire</i>
------	---------------------------

- void [destroy_kdtree](#) ([kdtree](#))

kdtree_get_nearest_color

La fonction cherche la couleur la plus proche dans tout l arbre

Parameters

tree	<i>la racine de l arbre de recherche</i>
inputColor	<i>la couleur d origine dont on cherche la plus proche</i>
outputColor	<i>la couleur la plus proche trouvee</i>

- void [kdtree_get_nearest_color](#) ([kdtree](#), [color *](#), [color *](#))

4.1.1 Detailed Description

Structures et headers des fonctions pour les kd arbres.

Author

Durand Valentin TP3 Informatique Promo 2018

Date

Janvier 2016

4.1.2 Typedef Documentation**4.1.2.1 cutting_plane::cuttingPlane pointeur sur structure cutting_plane**

pointeur de structure [cutting_plane](#)

4.1.2.2 kdtree

pointeur de structure kdtree

4.1.3 Function Documentation**4.1.3.1 kdtree create_kdtree (color_table , int)****4.1.3.2 void destroy_kdtree (kdtree)****4.1.3.3 void kdtree_get_nearest_color (kdtree , color * , color *)****4.2 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/classe.h File Reference****Macros**

- #define [CLASSE](#)(c_ident) typedef struct c_ident *c_ident

4.2.1 Macro Definition Documentation

4.2.1.1 `#define CLASSE(c_ident) typedef struct c_ident *c_ident`

4.3 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/exit_if.h File Reference

```
#include <stdio.h>
```

Macros

- `#define EXIT_IF(expr, mess, type)`

4.3.1 Macro Definition Documentation

4.3.1.1 `#define EXIT_IF(expr, mess, type)`

Value:

```
((expr)\
? (fprintf(stderr, ">EXIT:%s:%d: %s %s\n", __FILE__, __LINE__, mess, type), exit(1)) : 0)
```

4.4 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/image.h File Reference

```
#include "objet.h"
#include "classe.h"
#include "point.h"
#include "move_type.h"
```

Typedefs

- `typedef double(* PFdist)(int *, int *, booleen)`

Functions

- [CLASSE](#) ([image](#))
- [image FAIRE_image](#) ()
- [void DEFAIRE_image](#) ([image](#))
- [int image_charger](#) ([image](#), [char *](#))
- [int image_sauvegarder](#) ([image](#), [char *](#))
- [int image_to_stream](#) ([image](#), [FILE *](#))
- [int * image_lire_pixel](#) ([image](#))
- [void image_ecrire_pixel](#) ([image](#), [int *](#))
- [booleen image_pixel_suivant](#) ([image](#))
- [void image_debut](#) ([image](#))
- [booleen image_move_to](#) ([image](#), [point](#))
- [booleen image_pixel_dessus](#) ([image](#))
- [booleen image_pixel_dessous](#) ([image](#))
- [booleen image_pixel_gauche](#) ([image](#))
- [booleen image_pixel_droite](#) ([image](#))

- void [image_set_distance](#) (image, PFdist)
- void [image_initialize](#) (image, int, int, int)
- double [image_distance](#) (image, point, move_type)
- int [image_give_dim](#) (image)
- int [image_give_hauteur](#) (image)
- int [image_give_largeur](#) (image)

4.4.1 Typedef Documentation

4.4.1.1 typedef double(* PFdist)(int *, int *, boolean)

4.4.2 Function Documentation

4.4.2.1 CLASSE (image)

4.4.2.2 void DEFAIRE_image (image)

4.4.2.3 image FAIRE_image ()

4.4.2.4 int image_charger (image , char *)

4.4.2.5 void image_debut (image)

4.4.2.6 double image_distance (image , point , move_type)

4.4.2.7 void image_ecrire_pixel (image , int *)

4.4.2.8 int image_give_dim (image)

4.4.2.9 int image_give_hauteur (image)

4.4.2.10 int image_give_largeur (image)

4.4.2.11 void image_initialize (image , int , int , int)

4.4.2.12 int* image_lire_pixel (image)

4.4.2.13 boolean image_move_to (image , point)

4.4.2.14 boolean image_pixel_dessous (image)

4.4.2.15 boolean image_pixel_dessus (image)

4.4.2.16 boolean image_pixel_droite (image)

4.4.2.17 boolean image_pixel_gauche (image)

4.4.2.18 boolean image_pixel_suivant (image)

4.4.2.19 int image_sauvegarder (image , char *)

4.4.2.20 void image_set_distance (image , PFdist)

4.4.2.21 int image_to_stream (image , FILE *)

4.5 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/move_type.h File Reference

```
#include "type_obj.h"
```

Macros

- #define [NB_MOVE](#) 4
- #define [RIGHT](#) 0
- #define [UP](#) 1
- #define [LEFT](#) 2
- #define [DOWN](#) 3

Typedefs

- typedef int [move_type](#)

Functions

- long [move_type_compute_step_clock](#) ([move_type](#), [move_type](#))
- int [move_type_compute_dist_sigma_1](#) ([move_type](#), [move_type](#))

4.5.1 Macro Definition Documentation

4.5.1.1 #define [DOWN](#) 3

4.5.1.2 #define [LEFT](#) 2

4.5.1.3 #define [NB_MOVE](#) 4

4.5.1.4 #define [RIGHT](#) 0

4.5.1.5 #define [UP](#) 1

4.5.2 Typedef Documentation

4.5.2.1 typedef int [move_type](#)

4.5.3 Function Documentation

4.5.3.1 int [move_type_compute_dist_sigma_1](#) ([move_type](#) , [move_type](#))

4.5.3.2 long [move_type_compute_step_clock](#) ([move_type](#) , [move_type](#))

4.6 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/objet.h File Reference

```
#include <stdlib.h>
#include <stddef.h>
#include <assert.h>
#include "exit_if.h"
```

Macros

- #define [INSTANCIER](#)(c_ident) c_ident self = malloc(sizeof(struct c_ident));assert(self);
- #define [SELF](#)(champ) (self->champ)

4.6.1 Macro Definition Documentation

4.6.1.1 #define [INSTANCIER](#)(c_ident) c_ident self = malloc(sizeof(struct c_ident));assert(self);

4.6.1.2 #define [SELF](#)(champ) (self->champ)

4.7 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/point.h File Reference

```
#include "classe.h"
#include "move_type.h"
#include "type_obj.h"
```

Data Structures

- struct [point](#)

Macros

- #define [EGAL_POINT](#)(P1, P2) (((P1).coor dx == (P2).coor dx) && ((P1).coor dy == (P2).coor dy))
- #define [COORDX](#)(P) ((P).coor dx)
- #define [COORDY](#)(P) ((P).coor dy)
- #define [IS_RIGHTER_POINT](#)(p1, p2) ((p1).coor dx > (p2).coor dx)
- #define [AFFECTE_POINT](#)(p1, p2)
- #define [MOVE_RIGHT_POINT](#)(p) ((p).coor dx++)
- #define [MOVE_LEFT_POINT](#)(p) ((p).coor dx--)
- #define [MOVE_UP_POINT](#)(p) ((p).coor dy--)
- #define [MOVE_DOWN_POINT](#)(p) ((p).coor dy++)
- #define [NULL_POINT](#)(p) (p == NULL)
- #define [INFINITE_POINT](#)(p) (((p).coor dx == INFINITE) && ((p).coor dy == INFINITE))

Typedefs

- typedef struct [point](#) [Point](#)

Functions

- [CLASSE](#) ([point](#))

4.7.1 Macro Definition Documentation

4.7.1.1 #define [AFFECTE_POINT](#)(p1, p2)

Value:

```
((p1).coor dx = (p2).coor dx, \
  (p1).coor dy = (p2).coor dy)
```

4.7.1.2 `#define COORDX(P) ((P).coordx)`

4.7.1.3 `#define COORDY(P) ((P).coordy)`

4.7.1.4 `#define EGAL_POINT(P1, P2) (((P1).coordx == (P2).coordx) && ((P1).coordy == (P2).coordy))`

4.7.1.5 `#define INFINITE_POINT(p) (((p).coordx == INFINITE) && ((p).coordy == INFINITE))`

4.7.1.6 `#define IS_RIGHTER_POINT(p1, p2) ((p1).coordx > (p2).coordx)`

4.7.1.7 `#define MOVE_DOWN_POINT(p) ((p).coordy++)`

4.7.1.8 `#define MOVE_LEFT_POINT(p) ((p).coordx--)`

4.7.1.9 `#define MOVE_RIGHT_POINT(p) ((p).coordx++)`

4.7.1.10 `#define MOVE_UP_POINT(p) ((p).coordy--)`

4.7.1.11 `#define NULL_POINT(p) (p == NULL)`

4.7.2 Typedef Documentation

4.7.2.1 `typedef struct point Point`

4.7.3 Function Documentation

4.7.3.1 `CLASSE (point)`

4.8 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/table.h File Reference

Structures et headers des fonctions pour les tables de couleurs.

Data Structures

- struct [color_table](#)

structure permettant de stocker les donnees d une table de couleur

Typedefs

- typedef int [color](#)

chaque valeur de la composante d une couleur est un entier

- typedef struct [color_table](#) * [color_table](#)

pointeur de structure [color_table](#)

Enumerations

- enum [axis](#) { [red](#), [green](#), [blue](#) }

une axe est une compsanete RGB d une couleur 0 pour l'axe rouge, 1 pour le vert, et 2 pour le bleu

- enum [boolean](#) { [false](#), [true](#) }

booleen vrai faux

Functions

create_color_table

La fonction cree une table a partir d'une image composee d'une seule ligne.

Parameters

img	image a partir de laquelle on cree la table de couleur
-----	--------------------------------------------------------

Returns

colorTable la table de couleur cree Il faut donc allouer le pointeur, initialiser la longueur et remplir le tableau.
Le champ owner est positionne a true dans ce cas.

- [color_table create_color_table](#) (image)

destroy_color_table

La fonction supprime une table de couleur.

Parameters

cTable	la table de couleur a supprimer
--------	---------------------------------

Return values

TRUE	si libere avec owner
FALSE	si libere sans owner Le pointeur ne doit etre libere que si owner est positionne a true. La structure contenant la table doit par contre etre liberee dans tous les cas.

- [boolean destroy_color_table](#) (color_table)

color_table_duplicate

La fonction cree une sous table.

Parameters

cTable	table de couleur mere
offset	indice de debut de la sous table
length	longueur de la sous table

Returns

subTable la sous table de couleur cree Cette fonction initialise son pointeur a partir de celui de la table principale sans faire d'allocation. Le champ owner doit etre positionne a false.

- [color_table color_table_duplicate](#) (color_table, int, int)

color_table_get_color

La fonction donne la couleur rgb a une position precise dans la table de couleur.

Parameters

cTable	table de couleur dans lequel on veut trouver la couleur
index	position dans la table ou on veut la couleur
foundColor	tableau 1D de taille 3 stockant la couleur rgb trouvee

Returns

rien

- void [color_table_get_color](#) (color_table, int, color *)

color_table_get_nb_color

La fonction donne le nombre de couleurs dans la table de couleur.

Parameters

cTable	table dont on veut le nombre de couleur
--------	-----------------------------------------

Returns

int le nombre de couleur

- int [color_table_get_nb_color](#) ([color_table](#))

color_table_sort

La fonction trie les elements d'une table de couleur par ordre croissant en fonction de leurs coodonnees sur un axe

Parameters

cTable	table de couleur a trier
sortingAxis	axe RGB utilise pour le tri

Returns

rien 0 pour l'axe rouge, 1 pour le vert, et 2 pour le bleu

- void [color_table_sort](#) ([color_table](#), [axis](#))

color_table_get_min

La fonction retourne la couleur la plus faible pour un axe rgb donné

Parameters

cTable	table de couleur dont on cherche le minimum
sortingAxis	axe RGB utilise pour chercher le minimum

Returns

la couleur la plus petite trouvee

- [color](#) [color_table_get_min](#) ([color_table](#), [axis](#))

color_table_get_max

La fonction retourne la couleur la plus grande pour un axe rgb donné

Parameters

cTable	table de couleur dont on cherche le maximum
sortingAxis	axe RGB utilise pour chercher le maximum

Returns

la couleur la plus grande trouvee

- [color](#) [color_table_get_max](#) ([color_table](#), [axis](#))

4.8.1 Detailed Description

Structures et headers des fonctions pour les tables de couleurs.

Author

Durand Valentin TP3 Informatique Promo 2018

Date

Decembre 2015

4.8.2 Typedef Documentation

4.8.2.1 color

chaque valeur de la composante d une couleur est un entier

4.8.2.2 color_table

pointeur de structure [color_table](#)

4.8.3 Enumeration Type Documentation

4.8.3.1 enum axis

une axe est une compsanse RGB d une couleur 0 pour l'axe rouge, 1 pour le vert, et 2 pour le bleu

Enumerator

red
green
blue

4.8.3.2 enum boolean

booleen vrai faux

Enumerator

false
true

4.8.4 Function Documentation

4.8.4.1 `color_table color_table_duplicate (color_table , int , int)`

4.8.4.2 `void color_table_get_color (color_table , int , color *)`

4.8.4.3 `color color_table_get_max (color_table , axis)`

4.8.4.4 `color color_table_get_min (color_table , axis)`

4.8.4.5 `int color_table_get_nb_color (color_table)`

4.8.4.6 `void color_table_sort (color_table , axis)`

4.8.4.7 `color_table create_color_table (image)`

4.8.4.8 `boolean destroy_color_table (color_table)`

4.9 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/include/type_obj.h File Reference

```
#include <limits.h>
```

Macros

- #define `DIM_MAX` 3

Typedefs

- typedef int `angle`

Enumerations

- enum `booleen` { `faux`, `vrai` }

4.9.1 Macro Definition Documentation

4.9.1.1 #define DIM_MAX 3

4.9.2 Typedef Documentation

4.9.2.1 typedef int angle

4.9.3 Enumeration Type Documentation

4.9.3.1 enum booleen

Enumerator

faux

vrai

4.10 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/arbre.c File Reference

Fonctions pour les kd arbres.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "image.h"
#include "arbre.h"
```

Functions

choose_sorting_axis_by_max

La fonction choisi l'axe de coupe avec la plus grande difference entre le min et le max

private

Parameters

<code>cTable</code>	<i>table de couleur dont on cherche l'axe de coupe</i>
---------------------	--------------------------------------------------------

Returns

axis l'axe de coupe choisi

- [axis choose_sorting_axis_by_max](#) ([color_table](#) cTable)

choose_sorting_axis_by_var

La fonction choisi l'axe de coupe avec la plus grande variance

private

Parameters

cTable	table de couleur dont on cherche l'axe de coupe
--------	-------------------------------------------------

Returns

axis l'axe de coupe choisi

- [axis choose_sorting_axis_by_var](#) ([color_table](#) cTable)

choose_axis_position

La fonction trouve le plan de coupe

private

Parameters

cTable	table de couleur dont on cherche le plan de coupe
sortingAxis	l'axe de coupe auquel le plan est orthogonal

Returns

int la position dans la table de couleur de la première couleur supérieure au plan de coupe

- [int choose_axis_position](#) ([color_table](#) cTable, [axis](#) sortingAxis)

create_kdtree

La fonction crée le kd arbre de tri

public

Parameters

cTable	table de couleur à découper en kd arbre
seuil	le nombre minimum de couleur dans une feuille de l'arbre

Returns

kdtree pointer sur la racine de l'arbre

- [kdtree create_kdtree](#) ([color_table](#) cTable, [int](#) seuil)

destroy_kdtree

La fonction supprime le kd arbre et ses fils

public

Parameters

tree	l'arbre à détruire
------	--------------------

- void `destroy_kdtree` (`kdtree` tree)

`get_distance_between_colors`

La fonction calcul la distance entre 2 couleurs

private

Parameters

a	une couleur sous forme d un pointer vers tableau des 3 composantes
b	une couleur sous forme d un pointer vers tableau des 3 composantes

Returns

double la distance entre a et b

- double `get_distance_between_colors` (`color` *a, `color` *b)

`search_nearest_color_in_node`

La fonction cherche la couleur la plus proche dans un noeud

private

Parameters

tree	le noeud dans lequel on fait la recherche
inputColor	la couleur d origine dont on cherche la plus proche
outputColor	la couleur la plus proche trouvee

- void `search_nearest_color_in_node` (`kdtree` tree, `color` *inputColor, `color` *outputColor)

`kdtree_get_nearest_color`

La fonction cherche la couleur la plus proche dans tout l arbre

public

Parameters

tree	la racine de l arbre de recherche
inputColor	la couleur d origine dont on cherche la plus proche
outputColor	la couleur la plus proche trouvee

- void `kdtree_get_nearest_color` (`kdtree` tree, `color` *inputColor, `color` *outputColor)

4.10.1 Detailed Description

Fonctions pour les kd arbres.

Author

Durand Valentin TP3 Informatique Promo 2018

Date

Janvier 2016

4.10.2 Function Documentation

4.10.2.1 `int choose_axis_position (color_table cTable, axis sortingAxis)`

4.10.2.2 `axis choose_sorting_axis_by_max (color_table cTable)`

4.10.2.3 `axis choose_sorting_axis_by_var (color_table cTable)`

4.10.2.4 `kdtree create_kdtree (color_table cTable, int seuil)`

4.10.2.5 `void destroy_kdtree (kdtree tree)`

4.10.2.6 `double get_distance_between_colors (color * a, color * b)`

4.10.2.7 `void kdtree_get_nearest_color (kdtree tree, color * inputColor, color * outputColor)`

4.10.2.8 `void search_nearest_color_in_node (kdtree tree, color * inputColor, color * outputColor)`

4.11 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/graphdata.c File Reference

Sauvegarde les temps d execution pour l inversion d une image par methode triviale et kd arbre.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <time.h>
#include "image.h"
#include "arbre.h"
```

Functions

- `int get_exec_time_tree (char *fichierImg, char *fichierTable, int threshold)`
- `int get_exec_time_trivial (char *fichierImg, char *fichierTable)`
- `int main (int argc, char *argv[])`

4.11.1 Detailed Description

Sauvegarde les temps d execution pour l inversion d une image par methode triviale et kd arbre.

Author

Durand Valentin TP3 Informatique Promo 2018

Date

Janvier 2016

4.11.2 Function Documentation

4.11.2.1 `int get_exec_time_tree (char * fichierImg, char * fichierTable, int threshold)`

4.11.2.2 `int get_exec_time_trivial (char * fichierImg, char * fichierTable)`

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

4.12 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/image.c File Reference

```
#include <stdio.h>
#include <time.h>
#include "image.h"
#include "point.h"
#include "type_obj.h"
```

Data Structures

- struct [image](#)

Macros

- #define [Debut SELF](#)(debut)
- #define [Courant SELF](#)(courant)
- #define [Largeur SELF](#)(largeur)
- #define [Hauteur SELF](#)(hauteur)
- #define [Longueur SELF](#)(longueur)
- #define [Taille SELF](#)(taille)
- #define [Distance SELF](#)(distance)
- #define [Dim SELF](#)(dim)
- #define [DEPLACEMENT](#)(i, j) ([Dim](#)*((i)*[Largeur](#)+(j)))
- #define [MODE_LECTURE](#) "r"
- #define [MODE_ECRITURE](#) "w"

Typedefs

- typedef unsigned char [Comp_rgb](#)
- typedef int(* [PFsauvegarde](#))(image, FILE *)

Functions

- [image FAIRE_image](#) ()
- void [DEFAIRE_image](#) (image self)
- void [image_initialize](#) (image self, int dim, int largeur, int hauteur)
- int [image_charger](#) (image self, char *nom)
- int [image_sauvegarder](#) (image self, char *nom)
- int [image_to_stream](#) (image self, FILE *id)
- int * [image_lire_pixel](#) (image self)
- int [image_get_comp](#) (image self, point p, int comp)
- void [image_ecrire_pixel](#) (image self, int *tab)
- boolean [image_pixel_suivant](#) (image self)
- void [image_debut](#) (image self)
- boolean [image_move_to](#) (image self, point p)
- boolean [image_pixel_dessus](#) (image self)
- boolean [image_pixel_dessous](#) (image self)
- boolean [image_pixel_gauche](#) (image self)
- boolean [image_pixel_droite](#) (image self)

- double `image_distance` (`image self`, `point p`, `move_type move`)
- void `image_set_distance` (`image self`, `PFdist fct_dist`)
- int `image_give_dim` (`image self`)
- int `image_give_largeur` (`image self`)
- int `image_give_hauteur` (`image self`)

4.12.1 Macro Definition Documentation

4.12.1.1 `#define Courant SELF(courant)`

4.12.1.2 `#define Debut SELF(debut)`

4.12.1.3 `#define DEPLACEMENT(i, j) (Dim*((i)*Largeur+(j)))`

4.12.1.4 `#define Dim SELF(dim)`

4.12.1.5 `#define Distance SELF(distance)`

4.12.1.6 `#define Hauteur SELF(hauteur)`

4.12.1.7 `#define Largeur SELF(largeur)`

4.12.1.8 `#define Longueur SELF(longueur)`

4.12.1.9 `#define MODE_ECRITURE "w"`

4.12.1.10 `#define MODE_LECTURE "r"`

4.12.1.11 `#define Taille SELF(taille)`

4.12.2 Typedef Documentation

4.12.2.1 `typedef unsigned char Comp_rgb`

4.12.2.2 `typedef int(* PFsauvegarde)(image, FILE *)`

4.12.3 Function Documentation

4.12.3.1 `void DEFAIRE_image (image self)`

4.12.3.2 `image FAIRE_image ()`

4.12.3.3 `int image_charger (image self, char * nom)`

4.12.3.4 `void image_debut (image self)`

4.12.3.5 `double image_distance (image self, point p, move_type move)`

4.12.3.6 `void image_ecrire_pixel (image self, int * tab)`

4.12.3.7 `int image_get_comp (image self, point p, int comp)`

4.12.3.8 `int image_give_dim (image self)`

4.12.3.9 `int image_give_hauteur (image self)`

- 4.12.3.10 `int image_give_largeur (image self)`
- 4.12.3.11 `void image_initialize (image self, int dim, int largeur, int hauteur)`
- 4.12.3.12 `int* image_lire_pixel (image self)`
- 4.12.3.13 `booleen image_move_to (image self, point p)`
- 4.12.3.14 `booleen image_pixel_dessous (image self)`
- 4.12.3.15 `booleen image_pixel_dessus (image self)`
- 4.12.3.16 `booleen image_pixel_droite (image self)`
- 4.12.3.17 `booleen image_pixel_gauche (image self)`
- 4.12.3.18 `booleen image_pixel_suivant (image self)`
- 4.12.3.19 `int image_sauvegarder (image self, char * nom)`
- 4.12.3.20 `void image_set_distance (image self, PFdist fct_dist)`
- 4.12.3.21 `int image_to_stream (image self, FILE * id)`

4.13 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/inversion.c File Reference

Inversion d une image par une table de couleur en un kd arbre.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include "image.h"
#include "arbre.h"
```

Functions

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

4.13.1 Detailed Description

Inversion d une image par une table de couleur en un kd arbre.

Author

Durand Valentin TP3 Informatique Promo 2018

Date

Janvier 2016 On decoupe la table de couleur dans un arbre afin de reduire le nombre de couleur a comparer

4.13.2 Function Documentation

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

4.14 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/table.c File Reference

Fonctions pour les tables de couleurs.

```
#include <stdio.h>
#include <stdlib.h>
#include "image.h"
#include "table.h"
```

Functions

create_color_table

La fonction cree une table a partir d'une image composee d'une seule ligne.

public

Parameters

<code>img</code>	<i>image a partir de laquelle on cree la table de couleur</i>
------------------	---------------------------------------------------------------

Returns

*colorTable la table de couleur cree Il faut donc allouer le pointeur, initialiser la longueur et remplir le tableau.
Le champ owner est positionne a true dans ce cas.*

- [color_table create_color_table \(image img\)](#)

destroy_color_table

La fonction supprime une table de couleur.

public

Parameters

<code>cTable</code>	<i>la table de couleur a supprimer</i>
---------------------	----------------------------------------

Return values

<code>TRUE</code>	<i>si libere avec owner</i>
<code>FALSE</code>	<i>si libere sans owner Le pointeur ne doit etre libere que si owner est positionne a true. La structure contenant la table doit par contre etre liberee dans tous les cas.</i>

- [boolean destroy_color_table \(color_table cTable\)](#)

color_table_duplicate

La fonction cree une sous table.

public

Parameters

<code>cTable</code>	<i>table de couleur mere</i>
<code>offset</code>	<i>indice de debut de la sous table</i>
<code>length</code>	<i>longueur de la sous table</i>

Returns

subTable la sous table de couleur cree Cette fonction initialise son pointeur a partir de celui de la table principale sans faire d'allocation. Le champ owner doit etre positionne a false.

- `color_table color_table_duplicate (color_table cTable, int offset, int length)`

color_table_get_color

La fonction donne la couleur rgb a une position precise dans la table de couleur.

public

Parameters

cTable	<i>table de couleur dans lequel on veut trouver la couleur</i>
index	<i>position dans la table ou on veut la couleur</i>
foundColor	<i>tableau 1D de taille 3 stockant la couleur rgb trouvee</i>

Returns

rien

- `void color_table_get_color (color_table cTable, int index, color *foundColor)`

color_table_get_nb_color

La fonction donne le nombre de couleurs dans la table de couleur.

public

Parameters

cTable	<i>table dont on veut le nombre de couleur</i>
--------	------------------------------------------------

Returns

int le nombre de couleur

- `int color_table_get_nb_color (color_table cTable)`

color_table_sort

La fonction trie les elements d'une table de couleur par ordre croissant en fonction de leurs coordonnees sur un axe

public

Parameters

cTable	<i>table de couleur a trier</i>
sortingAxis	<i>axe RGB utilise pour le tri</i>

Returns

rien

- `void color_table_sort (color_table cTable, axis sortingAxis)`

color_table_get_min

La fonction retourne la couleur la plus faible pour un axe rgb donne

public

Parameters

cTable	table de couleur dont on cherche le minimum
sortingAxis	axe RGB utilise pour chercher le minimum

Returns

la couleur la plus petite trouvee

- [color](#) [color_table_get_min](#) ([color_table](#) cTable, [axis](#) sortingAxis)

color_table_get_max

La fonction retourne la couleur la plus grande pour un axe rgb donné

public

Parameters

cTable	table de couleur dont on cherche le maximum
sortingAxis	axe RGB utilise pour chercher le maximum

Returns

la couleur la plus grande trouvee

- [color](#) [color_table_get_max](#) ([color_table](#) cTable, [axis](#) sortingAxis)

4.14.1 Detailed Description

Fonctions pour les tables de couleurs.

Author

Durand Valentin TP3 Informatique Promo 2018

Date

Decembre 2015

4.14.2 Function Documentation

4.14.2.1 [color_table](#) [color_table_duplicate](#) ([color_table](#) cTable, int offset, int length)

4.14.2.2 void [color_table_get_color](#) ([color_table](#) cTable, int index, [color](#) * foundColor)

4.14.2.3 [color](#) [color_table_get_max](#) ([color_table](#) cTable, [axis](#) sortingAxis)

4.14.2.4 [color](#) [color_table_get_min](#) ([color_table](#) cTable, [axis](#) sortingAxis)

4.14.2.5 int [color_table_get_nb_color](#) ([color_table](#) cTable)

4.14.2.6 void [color_table_sort](#) ([color_table](#) cTable, [axis](#) sortingAxis)

4.14.2.7 [color_table](#) [create_color_table](#) ([image](#) img)

4.14.2.8 boolean [destroy_color_table](#) ([color_table](#) cTable)

4.15 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/test_arbre.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "image.h"
#include "arbre.h"
```

Functions

- void [printTree](#) ([kdtree](#) t)
- int [main](#) (int argc, char *argv[])

4.15.1 Function Documentation

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

4.15.1.2 void [printTree](#) ([kdtree](#) t)

4.16 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/test_table.c File Reference

test des differents fonctions de [table.c](#)

```
#include <stdio.h>
#include <stdlib.h>
#include "image.h"
#include "table.h"
```

Functions

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

4.16.1 Detailed Description

test des differents fonctions de [table.c](#)

Author

Durand Valentin TP3 Informatique Promo 2018

Date

Decembre 2015

4.16.2 Function Documentation

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

4.17 /media/psf/Home/Desktop/Prog C/ProjetC-Inversion/src/trivial.c File Reference

Inversion d une image par une table de couleur en utilisant une methode triviale.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include "image.h"
#include "table.h"
```

Functions

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

4.17.1 Detailed Description

Inversion d'une image par une table de couleur en utilisant une méthode triviale.

Author

Durand Valentin TP3 Informatique Promo 2018

Date

Décembre 2015 Avec cette méthode on calcule la distance entre chaque pixel de l'image et toutes les couleurs de la table de couleur successivement

4.17.2 Function Documentation

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