Projet PPM Viewer - Traitement d'Images

Informations du Projet

Université : Université de Yaoundé 1
Département : Informatique
Niveau : INF L2
Année académique : 2025-2026
Module : INF231 - Technique de Conception d'Algorithmes et Structures de Données
Travail Pratique : TP n°1 - Traitement d'images avec les fichiers et tableaux

---

Membres de l'Équipe (TPE1)

Nom	Tâche(s) Effectuée(s)
LONTSI TCHINDA ARTHUR	Tâche 1 - Foncer/Éclaircir les pixels selon dominante
KENDO DJOSSEU CARELLE	Tâche 2 - Conversion en noir et blanc + README
AMOUGOU FANJIP MAEVA	Tâche 3 - Création du négatif d'image
MEKOUDJA MENGAM DIANE	Tâche 4 - Affichage de la taille + Interface ppmviewer
NCHARE NJUTAP RONALD	Tâche 5 - Découpage d'image
FEZEU DOMGOUO BRAYAN	Tâche 6 - Filtre médian

Description du Projet

Ce projet consiste en une application de traitement d'images au format PPM (Portable PixMap) écrite en langage C. Le format PPM permet de stocker des images non compressées en écrivant la liste des pixels avec leurs couleurs en RGB (Red, Green, Blue).

L'application offre une interface en ligne de commande permettant d'effectuer diverses opérations de traitement d'images.

Structure du Code

Types de Données Définis

c

```c
typedef struct {
    int r, g, b;
} pixels;

typedef struct {
    char format[2];
    int largeur, hauteur, max;
    pixels tab[255][255];
} images;
```

Fonctions Principales

1. **generer_image()** - Génère une image PPM avec un dégradé de couleurs
2. **largeur_hauteur()** - Affiche les dimensions d'une image
3. **foncer_eclaicir()** - Fonce ou éclaircit les pixels selon leur dominante
4. **gris()** - Convertit une image en niveaux de gris
5. **negatif()** - Crée le négatif d'une image
6. **couper()** - Découpe une partie spécifique d'une image
7. **filtre_median()** - Applique un filtre médian sur l'image

---

Fonctionnalités Implémentées

1. Foncer ou Éclaircir (Tâche 1 - LONTSI TCHINDA ARTHUR)

Commande : `dom c val fichier.ppm`

Description : Fonce ou éclaircit tous les pixels ayant une dominante rouge, bleue ou verte.

- `c` : couleur dominante (R, G ou B)
- `val` : valeur à ajouter (positif pour foncer) ou retrancher (négatif pour éclaircir)
- La dominante d'un pixel est la couleur primaire avec la valeur la plus élevée

Fonctionnement :

- Lecture du fichier image.ppm
- Identification de la couleur dominante pour chaque pixel
- Ajout/soustraction de la valeur spécifiée
- Vérification des limites (0-255)
- Sauvegarde dans Image_dom.ppm

2. Conversion en Noir et Blanc (Tâche 2 - KENDO DJOSSEU CARELLE)

Commande : `gris fichier.ppm`

Description : Convertit l'image en niveaux de gris en utilisant la moyenne des trois composantes RGB.

Fonctionnement :

- Lecture du fichier source
- Pour chaque pixel : calcul de la moyenne (R+G+B)/3
- Application de cette valeur aux trois composantes
- Sauvegarde dans Image_gris.ppm

3. Créer le Négatif (Tâche 3 - AMOUGOU FANJIP MAEVA)

Commande : `neg fichier.ppm fichier_resultat.ppm`

Description : Crée le négatif d'une image en inversant toutes les couleurs.

Fonctionnement :

- Lecture du fichier source
- Pour chaque composante : nouvelle_valeur = 255 - ancienne_valeur
- Sauvegarde dans Image3.ppm

4. Afficher la Taille (Tâche 4 - MEKOUDJA MENGAM DIANE)

Commande : `size fichier.ppm`

Description : Affiche les dimensions de l'image (largeur × hauteur) en pixels.

Fonctionnement :

- Lecture de l'en-tête du fichier PPM
- Extraction de la largeur et hauteur
- Affichage au format : `largeur x hauteurpx`

Interface PPMViewer : Cette tâche inclut également la mise en place de l'interface en ligne de commande avec le prompt `PPMVIEWER>` et la gestion des commandes.

5. Découper une Image (Tâche 5 - NCHARE NJUTAP RONALD)

Commande : `cut fichier.ppm l1 l2 c1 c2 fichier_resultat.ppm`

Description : Découpe et sauvegarde une partie de l'image entre les lignes l1 et l2 et les colonnes c1 et c2.

Contraintes :

- $l1 < l2 \leq$ hauteur
- $c1 < c2 \leq$ largeur

Fonctionnement :

- Validation des paramètres
- Extraction de la zone spécifiée
- Sauvegarde dans Image_resultat.ppm

6. Filtre Médian (Tâche 6 - FEZEU DOMGOUO BRAYAN)

Commande : `fil fichier.ppm fichier_resultat.ppm`

Description : Applique un filtre médian en remplaçant chaque couleur d'un pixel par la valeur médiane des couleurs des 8 pixels voisins.

Fonctionnement :

- Pour chaque pixel et chacune de ses composantes RGB
- Collecte des valeurs des 8 voisins
- Calcul de la médiane
- Remplacement de la valeur actuelle

Commandes Disponibles

Commande	Description
`dom c val fichier.ppm`	Foncer/éclaircir selon dominante
`gris fichier.ppm`	Convertir en niveaux de gris
`size fichier.ppm`	Afficher les dimensions
`cut fichier.ppm l1 l2 c1 c2 resultat.ppm`	Découper une partie
`neg fichier.ppm resultat.ppm`	Créer le négatif
`fil fichier.ppm resultat.ppm`	Appliquer filtre médian
`help`	Afficher la liste des commandes
`clear`	Effacer l'écran
`quit`	Quitter l'application

Compilation et Exécution

Compilation

```bash
gcc ppmviewer.c -o ppmviewer
```

Exécution

```bash
./ppmviewer
```

Au lancement, le programme affiche :

```
****WELCOME DEAR USER****
PPMVIEWER>
```

Exemple d'Utilisation

```
PPMVIEWER> size image.ppm
100 x 205px

PPMVIEWER> dom R 50 image.ppm
Image_dom cree avec succes!!!
Operation effectuee!

PPMVIEWER> gris image.ppm
Image_gris.ppm cree avec succes!!!
Operation effectue

PPMVIEWER> neg image.ppm resultat.ppm
Image en negatif avec succes!!!

PPMVIEWER> help
      LISTE DES COMMANDES:

       quit
       clear
       dom c val fichier.ppm
       gris fichier.ppm
       size fichier.ppm
       cut fichier.ppm l1 l2 c1 c2 fichier_resultat.ppm
       neg fichier.ppm fichier_resultat.ppm
       fil fichier.ppm fichier_resulata.ppm

PPMVIEWER> quit
```

Organisation du Développement

Répartition des Tâches

Le travail a été organisé de manière modulaire, chaque membre de l'équipe étant responsable d'une fonction spécifique :

1. **LONTSI TCHINDA ARTHUR** a développé la fonction `foncer_eclaicir()` qui gère l'ajustement de luminosité selon la couleur dominante du pixel.
2. **KENDO DJOSSEU CARELLE** a implémenté la fonction `gris()` pour la conversion en niveaux de gris et a rédigé ce document README.
3. **AMOUGOU FANJIP MAEVA** a créé la fonction `negatif()` qui inverse les couleurs de l'image.
4. **MEKOUDJA MENGAM DIANE** a développé la fonction `largeur_hauteur()` et a mis en place toute l'interface utilisateur (boucle principale, parsing des commandes, gestion du prompt).
5. **NCHARE NJUTAP RONALD** a implémenté la fonction `couper()` pour le découpage d'images.
6. **FEZEU DOMGOUO BRAYAN** a travaillé sur la fonction `filtre_median()` pour l'application du filtre médian.

Méthodologie

- Développement modulaire avec des fonctions indépendantes
- Tests individuels de chaque fonction
- Intégration progressive dans l'interface principale
- Gestion des erreurs (fichiers non trouvés, paramètres invalides)

Fichiers du Projet

- `ppmviewer.c` : Code source principal
- `image.ppm` : Fichier d'image d'exemple (généré)
- `README.md` : Ce fichier de documentation

Fichiers Générés

L'application génère différents fichiers selon les opérations :

- `Image_dom.ppm` : Image après foncer/éclaircir
- `Image_gris.ppm` : Image en niveaux de gris
- `Image3.ppm` : Image en négatif
- `Image_resultat.ppm` : Image découpée

Format PPM

Le programme traite des fichiers au format P3 (ASCII). Structure d'un fichier PPM :

```
P3
# commentaire
largeur hauteur
valeur_max
r1 g1 b1  r2 g2 b2  r3 g3 b3 ...
```

Exemple :

```
P3
# exemple d'image
4 4
255
3 8 25  0 0 0  0 0 0  15 0 15
0 0 0  0 15 7  0 0 0  255 0 18
0 0 0  0 0 0  0 15 7  0 0 0
15 0 15  0 0 0  0 0 0  66 0 4
```

Limites et Contraintes

- Taille maximale des images : 255×255 pixels (limitation du tableau statique)
- Format supporté : PPM P3 uniquement
- Les fichiers doivent être nommés `image.ppm` pour certaines fonctions
- Certaines fonctions lisent toujours depuis `image.ppm` (à améliorer)

Consignes Respectées

✅ Application écrite en C
✅ Traitement d'images au format PPM
✅ Six opérations implémentées
✅ Interface en ligne de commande avec prompt `ppmviewer>`
✅ Commandes selon le format spécifié
✅ Commande `quit` pour quitter
✅ Commande `help` pour l'aide
✅ Travail en groupe TPE1
✅ Dépôt GitHub avant le 10 Octobre 2025 à 20H
✅ README détaillé avec répartition des tâches

Date de Soumission

Vendredi 10 Octobre 2025 à 20H00

Références

- Spécification du format PPM : https://www.adobe.com/ca_fr/creativecloud/file-types/image/raster/ppm-file.html
- Énoncé du TP : TP_Serie1.pdf

Fin du Document