

Rapport

DA CRUZ DALCOL, Etienne.
REZENDE LAIGNIER, Sophia.

Introduction

L'objectif du projet c'est de construire un interprete de requêtes pour l'analyse d'images distant (le client peut envoyer une image au serveur a tout moment). Le client demande les types d'analyse desirés, y compris c'est disponible:

- Un histogramme d'une image montrant le taux de rouge, vert, ou bleu
- Un histogramme combinant les histogrammes des trois couleurs
- Un histogramme présentant le taux de présence de chaque couleur dans une image
- Une representation graphique montrant la repartition d'images selon leur taille, disposée dans un graphique où l'utilisateur choisit un (ou plusieurs) valeur(es) de taille specifique et l'histogramme indique le nombre d'images avec la taille plus petite et plus grande que la valeur choisie.
- Un histogramme montrant la repartition des images selon que la couleur rouge, verte, ou bleu est majoritaire

Développement

Nous avons commencé pour definir un groupe de fonctions pour manipuler les chaînes de caractères necessaires au programme (strhelpers.h). Deux fonctions ont été donnés (int str_split() et int startswith()), et nous avons fait autres deux:

```
char * str_to_lower(char *str);
```

Où toutes les commandes sont passées à minuscules (lowercase), de sorte que l'utilisateur peut utiliser des lettres majuscules et minuscules lors de l'entrée avec les requêtes.

```
int caseless_strcmp(char * str_a, char * str_b);
```

Pour comparer les commandes préétablies et les requêtes fournies pour l'utilisateur.

MODULE IMAGES.H et .C

Dans le projet, nous travaillons avec des images en extension TGA, consisté par deux parties principales: la tête (header) et les informations du bitmap. Donc, on a fait le header:

```
19  /* Un header targa basique. */
20  typedef struct targa_header_ {
21      uint8_t idlength;
22      uint8_t colourmaptype;
23      uint8_t datatypecode;
24      uint8_t useless[9];
25      uint16_t width;
26      uint16_t height;
27      uint8_t desc[2];
28  } targa_header;
29
```

idlength: C'est le nombre de bytes significatifs dans le domaine de l'identification de l'image. Le "idlength" mis à 0 indique qu'il n'y a pas de champ d'identification d'image dans le fichier TGA.

colourmaptype: Indique si le fichier TGA comprend une palette . Une valeur de 0 indique qu'aucune palette est inclus.

datatypecode: Indique le type d' image stockée dans le fichier TGA

Et une structure pour decire les elements d'image utiles au traitement:

```
32  typedef struct image_desc_
33  {
34      char *fname;
35      uint16_t width;
36      uint16_t height;
37      uint8_t * pRed;
38      uint8_t * pGreen;
39      uint8_t * pBlue;
40  } image_desc ;
41
```

Où pRed, pGreen et pBlue sont les références sur le plan mémoire des couleurs.

Pour recevoir et interpreter la message, nous utilisons la fonction:

void readParameters(char * msg, char * responsImagePath,char * error);

Où msg c'est la chaîne de caractères avec la message envoyé au serveur et **responsImagePath** et error sont des messages utilisés pour envoyer l'image ou un error.

Les commandes disponibles :
histogramme

Options disponibles:

-c : reprend la couleur

paramètres:

r : montre histogramme rouge

g : montre histogramme vert

b : montre histogramme bleue

rgb: montre le taux des couleurs

Valeur par défaut :

rgb

-i : image lue

paramètres:

nom de fichier d'image

-f : dossier lit

paramètres:

chemin de répertoire

-s: valeur fourni par l'utilisateur pour analyser la repartition d'images selon leur taille

paramètres:

taille (kb)

Exemple:

histogram -i 1.tga -c r

histogram -f images -s 400 -c rgb

Les autres fonctions sont des fonctions auxiliaires utilisés internement par le module en images.h.

OUTILS UTILISÉS

Sublime text

git

github

QUELQUES RESULTATS SONT TROUVÉS DANS LE DIRECTOIRE \histograms