Projet VARI 2002-2003
Visio : Un petit outil de détection et d'identification d'objets dans une image

Brian Fraval - brian@fraval.org
http://brian.fraval.org/cnam/

24 juin 2003

Table des matières

Chapitre 1

Introduction

Le projet de fin d'année de l'Unité de Valeur VARI - Valeur d'Accueil et de Reconversion à l'Informatique [1] - de cette année est un programme Ada qui doit permettre de détecter et d'identifier des objets dans une base d'image (Photothèque), à partir d'une image requête. Pour avoir plus d'informations sur ce projet, vous pouvez consulter le site [2] réalisé par Valérie Gouet, enseignant-chercheur au CNAM.

Ce document doit vous permettre de comprendre en détail le sujet du projet, son déroulement et différentes améliorations possibles. Dans un premier temps, il y aura une explication de toutes les étapes de la réalisation du projet. Ensuite une description des structures de données utilisées et des traitements choisis sera effectuée.

Pour vous aider à utiliser le programme, je détaillerai les démarches à effectuer pour tester l'application, en donnant des informations sur les images utilisées, lors du développement et des tests réalisés au cours du projet.

Pour finir, les problèmes non résolus et les améliorations possibles seront expliqués.

1.1 L'objectif du projet

L'objectif de ce projet est de créer un programme Ada qui permet de détecter et d'identifier automatiquement des objets dans une image. A partir d'une image requête passée en entrée au programme, il faudra déterminer si une ou plusieurs images de la photothèque contient un ou plusieurs objets de l'image requête.

Après avoir déterminé les images de la phototèque qui contiennent un ou plusieurs objets de l'image requête, un affichage des informations concernant les images de la photothèque sera réalisé.

1.2 Les programmes utilisés pour réaliser le projet

L'ensemble du projet VISIO a été réalisé sur une machine PIII 450 Mhz 196 Mo avec le système d'exploitation Debian GNU/Linux 2.4.20-686 et le compilateur GNU Ada GNU Ada 95 3.14p-3. L'éditeur utilisé pour écrire les sources de l'application et cette documentation est Emacs.

Afin de réaliser de meilleurs tests, j'ai aussi utilisé une machine Windows XP P4 2.4 GHz 256 Mo avec AdaGIDE v.6.22.10. Cela m'a permis d'être sûr de la portabilité de l'application.

L'application est aussi fonctionnelle sur la machine joule.cnam.fr du CNAM.

Chapitre 2

Les différentes étapes du projet

2.1 Méthode de caractérisation des points

2.1.1 Récupérer le nom de l'image requête

Le programme a besoin d'une image requête, afin de la comparer avec les différentes images de la photothèque. Cette comparaison doit permettre de définir si une ou plusieurs images de la photothèque ressemblent en partie ou en totalité à l'image passée au programme.

Le programme aura donc en argument le nom de l'image requête à analyser. J'ai pris l'initiative d'utiliser cette méthode et elle s'est avérée bonne, puisqu'elle a été ajouté à la documentation du projet [3] sur la page créée par Valérie GOUET.

Pour récupérer les arguments sur la ligne de commande, j'ai fait une recherche sur internet. Cette information est présente dans la FAQ [4] du groupe de discussion fr.comp.lang.ada, maintenue par Samuel TARDIEU.

2.1.2 Chargement de la photothèque

Après avoir récupéré le nom de l'image requête, j'ai essayé de respecter l'algorithme général [5] proposé pour réaliser le projet VISIO. Donc je me suis attaqué à la lecture sur le disque des informations associées à chaque image de la photothèque.

Je me suis servi du paquetage p_cadeau et plus particulièrement de la fonction TailleFichier. Cette fonction permet de savoir combien de ligne contient un fichier. Avec le projet, nous avons un fichier fphototheque.txt qui contient les informations concernant les images de la photothèque. Chaque ligne est utilisée pour décrire une image de la photothèque. Donc il est possible de savoir combien d'image contient la photothèque, avec la fonction TailleFichier, comme ceci :

```
NombreImages :=TailleFichier(NomFichierPhototheque) ;
```

Maintenant que je sais combien d'images contient la photothèque, il est possible de la déclarer, en utilisant le type Phototheque défini dans le paquetage p_cadeau.

```
Ph :Phototheque(1..NombreImages) ;
```

Pour charger la photothèque, j'utilise une fois de plus une procédure disponible dans le paquetage p_cadeau. Cette procédure s'appelle ChargerPhototheque. j'ai modifié cette procédure pour ajouter un mode debug au programme.

ChargerPhototheque(Debug, NomFichierPhototheque, Ph) ;

2.1.3 Nombre de point à analyser dans l'image requête

Les informations disponibles sur les images requêtes, concernent seulement les coordonnées des points d'intérêt. Ces coordonnées sont fourni avec le paquetage p_cadeau pour nous aider dans la réalisation du projet. Donc en plus de chaque image requête, nous avons un fichier du type nom_image.pts qui contient les coordonnées des points d'intérêt.

Une fois de plus, j'utilise la fonction TailleFichier pour stocker dans une variable le nombre de points présents dans le fichier pts de l'image requête. Ensuite j'utilise la structure DescripteurImage fourni dans le package p_cadeau, pour déclarer une variable qui contiendra la caratérisation de chaque point.

NombrePoints :=TailleFichier(NomFichierRequete&".pts") ;

DescriptionImageRequete : DescripteurImage(1..NombrePoints) ;

2.1.4 Chargement des coordonnées des points d'intérêt

Pour charger tous les points à analyser dans l'image requête, afin de les utiliser lors de l'analyse de la caractérisation de chaque point, j'utilise la procédure ChargeFichierPointsRequete du paquetage p_visio. Cette procédure permet de récupérer tous les coordonnées (x, y) des points d'intérêt et de les enregistrer dans la structure DescripteurImage.

```
for I in Description'range loop
  get(Fd,X);
  get(Fd,Y);
  Get_Line(Fd,Ligne,Lg);
  Description(I).X:=X;
  Description(I).Y:=Y;
end loop;
```

2.1.5 Lire le header de l'image requête

Pour calculer la description associée à chaque point de l'image requête, il faut lire l'image requête dans son ensemble. Cependant, il faut avant toute chose récupérer le header de l'image pour avoir des informations sur le type de l'image, sa taille, le nombre de couleurs, etc..

La lecture de l'image se décompose donc en deux phases. Une première fois pour récupérer le header d'information et une seconde phase pour récupérer les trois matrices R, V, B de l'image.

Une fois de plus, je créé un nouveau paquetage p_imageppm pour y stocker les procédures et les fonctions concernant le travail sur les images ppm.
Pour lire le header, j'utilise la procédure LireHeader qui permet de récupérer le type de l'image PPM (P6 ou P3), la taille de l'image (Hauteur, Largeur) et le nombre de couleur. Je n'utilise pas cette dernière information, mais cela pourrait servir pour une version plus évoluée de VISIO.

La lecture du header d'un fichier PPM n'est pas si simple, car il faut supprimer les lignes vides, les espaces, les tabulations et les commentaires. J'ai donc ajouté au paquetage la fonction SupprimeCommentaires qui permet de ne pas tenir compte des lignes de commentaire et la fonction SupprimeEspace qui permet de supprimer tous les caractères blancs (espace, tabulation, etc..).

2.1.6 Lire l'image requête. Chargement des matrices R, V, B

A partir des informations récupérées dans le header de l'image requête, il est possible de récupérer les trois matrices R pour la couleur rouge, V pour la couleur verte et B pour la couleur bleue. Ces trois matrices permettent de stocker tous les points de l'image requête.

Pour remplir les trois matrices, j'ai créé une procédure ChargeMatriceImage dans le paquetage p_imageppm. Cette procédure utilise le format P6 des images PPM, mais il est possible de la faire évoluer pour utiliser d'autre format.

Comme le format des images requête est du type P6, j'ai utilisé le paquetage sequential_io pour lire les valeurs codées en binaire des couleurs. Ensuite je transforme cette valeur avec Character'Pos(C) pour récupérer la couleur réelle.

Chaque couleur est stockée une à une dans sa matrice respective.

2.1.7 Caractérisation des points

Pour chaque point d'intérêt de l'image requête, il faut récupérer les couleurs présentes dans son voisinage, c'est à dire par une fenêtre carrée centrée en (x,y), et de taille paramètrable R (impaire). Chaque couleur est décrite dans l'espace colorimétrique RVB, et sera stockée dans un vecteur Vcar(x,y) qui contiendra donc R*R*3 grandeurs RVB.

J'ai créé une procédure CaracterisationPoints dans le paquetage p_visio, qui permet de stocker dans le Vcar toutes les couleurs. La valeur de R est positionnée 11, donc j'ai en tout 11 * 11 * 3 = 363 couleurs.

Ces couleurs sont stockées dans le Vcar de la manière suivante : PointR de 1 à 121, PointV de 122 à 242 et PointB de 243 à 363. Cela veut dire qu'il faut stocker dans le Vcar toutes les couleurs rouge, puis les couleurs vertes et pour finir les couleurs bleues. C'est en effet le format utilisé pour la caractérisation des images la notre photothèque.

Si le point d'intérêt se trouve près du bord de l'image et que la fenêtre carrée sort de l'image, alors par convention on choisit de mettre à -1 les éléments correspondants du vecteur caractéristique.

2.2 Méthode de comparaison des points caractérisés

La méthode utilisée pour la comparaison des points est la formule ZNCC - Zero Mean Normalized Cross Correlation [6]. Cette méthode revient à comparer tous les points de l'image requête avec tous les points de chaque image de la photothèque.

2.2.1 Chargement du fichier index des images de la photothèque

Pour chaque image de la photothèque, je récupère les coordonnées (x, y) des points d'intérêt, ainsi que leurs caractérisations. Pour réaliser ce travail, j'ai crée une procédure ChargeFichierIndex qui se trouve dans le package p_imageppm.

2.2.2 Comparaison des points et calcul du score de corrélation

Maintenant que j'ai toutes les informations des points d'intérêt de l'image requête, ainsi que ceux des images de la photothèque, je peux lancer la comparaison. Le résultat de cette comparaison et un score compris entre -1 et 1. 1 indiquant que les points sont fortement corrélés et au contraire -1 indiquant qu'ils sont fortement décorrélés.

La fonction ZNCC [6] se trouve dans le paquetage p_visio.
Cette méthode se décompose en trois parties, la première est le calcul de moyenne des N composantes des deux vecteurs passées à la fonction. La deuxième est le calcul du produit scalaire entre les deux vecteurs. Et la troisième est le calcul des normes euclidiennes des deux vecteurs.

Pour avoir plus d'information sur cette formule, je vous conseille de lire la page d'explication [5] sur le site du projet VISIO.

2.3 Seuillage des scores obtenus

Etant donné que tous les couples de points sont comparés entre deux images, il est évident qu'une grande partie d'entre eux ne se ressemblent pas du tout et sont donc décorrélés. Il est donc facile d'éliminer les scores qui sont inférieur ou égal à un seuil fixé dans l'intervalle [-1..1]. Ce seuil est défini dans le paquetage p_cadeau. Il est fixé à un seuil de 0.8.

Le seuillage des scores obtenus est géré dans la fonction CompareRequeteImagePhototheque du paquetage p_visio. L'algorithme de calcul des scores nous a été donné avec la documentation [5] du projet VISIO.

Pour supprimer les scores qui sont en dessous du seuil fixé dans le paquetage p_cadeau, je modifie la valeur du score par celui du seuil. La valeur du seuil est disponible dans les différents paquetages, donc il est facile de récupérer sa valeur pour ne pas tenir compte des points lors de l'élimination des appariements ambigus.

2.4 Elimination des appariements ambigus

Une fois de plus l'algorithme de l'élimination des appariements ambigus nous a été fourni avec la documentation [5] du projet. Il permet de supprimer l'ambiguité quand à un point sont associés plusieurs correspondants potentiels dans l'autre image.

J'ai ajouté la procédure AppariementsAmbigus dans le paquetage p_visio. Cette procédure permet de supprimer de la matrice Mscores tous les appariements ambigus. Une fois de plus, j'utilise la valeur du seuil définie dans le paquetage p_cadeau.

J'ai ajouté une analyse sur les lignes et les colonnes traitées, pour améliorer la rapidité du programme lors de la suppression des appariements ambibus. Cette information nous a été donnée lors de la présentation du projet par Valérie Gouet.

2.5 Calcul d'un vote pour chaque image de la photothèque

Maintenant que nous avons pour chaque couple (ImageRequete, Imagephototheque) un ensemble de points appariés, il est possible de calculer un vote qui permet de définir qu'elle est l'image de la photothèque qui ressemble le plus à celle de l'image requête. Ce vote représente le nombre de points de l'image de la photothèque appariés avec ceux de l'image requête par rapport au nombre total de points dans l'image de la photothèque.

Ainsi l'image la plus similaire aura un vote plus important que les autres. Le calcul du vote se trouve dans la fonction CompareRequeteImagePhototheque du paquetage p_visio.

2.6 Décision : classement et seuillage des images de la photothèque

Une fois le vote calculé, je classe la liste des images de la photothèque par ordre de vote décroissant. Cela permet d'afficher en premier l'image de la photothèque qui est succeptible de ressembler le plus à l'image requête. Ce classement est réalisé par la procedure OrdreDecroissantDesVotes dans le paquetage p_visio.

Ensuite, je calcule l'écart moyen entre les votes, pour mettre de côté les résultats qui ont un poids important. La fonction EcartMoyen se trouve dans le paquetage p_visio.

2.7 Affichage

L'affichage du résultat est réalisé par la procédure AfficheResultats du paquetage p_visio. Cette procédure utilise la procédure OrdreDecroissantDesVotes et EcartMoyen pour mettre en avant les meilleurs votes.

J'ai utilisé des tirets pour mettre en avant les meilleurs résultats.

Chapitre 3

Les structures de données

Maintenant que nous avons plus d'information sur l'algorithme du programme, je vais expliquer les différentes structures de données choisies pour la réalisation du projet.

3.1 Le paquetage p_cadeau

Ce paquetage nous a été offert au début du projet, pour nous aider dans la réalisation du projet VISIO. Ce paquetage contient les structures les plus importantes du projet. Notamment la définition des points d'une image, la définition de la photothèque, etc..

3.1.1 Structure de la photothèque

La photothèque est un tableau d'image, chaque image est décrite par le nom du fichier contenant l'image, le nom du fichier contenant la description, un label et un vote.

```
type ImagePhototheque is
  record
    NomFichierImage:Caracteres;
    NomFichierIndex:Caracteres;
    Label:Caracteres;
    Vote:Natural:=0;
  end record;

  type Phototheque is array(Integer range <>) of ImagePhototheque;
```

Le nom de l'image ainsi que le nom du fichier qui contient sa description est du type Caracteres. Ce type est composé d'une chaine de caractères (String) et d'une taille (Lg).

```
type Caracteres is
  record
  C:String(1..80):=(others=>' ');
  Lg:Natural range 0..80:=0;
end record;
```

3.1.2 Structures des images

Une image PPM est représentée par un header qui contient les informations sur l'image, le magic number (P3 ou P6), ses dimensions et le maximum de couleurs. Pour le reste de l'image, il est possible de la représenter par 3 tableaux à plusieurs dimensions (Matrice), qui contiennent des couleurs représentées par le type NiveauDeGris (Integer).

```
type T_Magic is (P3,P6);
package Magic_Io is new Enumeration_Io (T_Magic);
subtype NiveauDeGris is integer range -1..255;
type MatriceImage is array(Integer range <>,Integer range <>) of NiveauDeGris;
```

Un point est caractérisé par ses coordonnées (x,y) et par un vecteur caractéristique de type CaracterisationPoint. R est la taille d'un côté de la fenêtre de description d'un point. Pour une image définie par 3 plans couleur RVB, on a donc un vecteur caractéristique de dimension R*R*3 appelé DIM_ESPACE_CARAC.

Le contenu visuel d'une image (de test ou de la photothèque) est décrit par un ensemble de points. C'est le type DescripteurImage.

```
R:constant Positive:=11;
DIM_ESPACE_CARAC:constant Positive:=R*R*3;
subtype NiveauDeGris is integer range -1..255;
type CaracterisationPoint is array(Integer range 1..DIM_ESPACE_CARAC)
of NiveauDeGris;

type Point is
record
X,Y:Natural:=0;
  Vcar:CaracterisationPoint:=(others=>0);
end record;

type DescripteurImage is array(Integer range <>) of Point;
```

Ensuite, nous avons besoin de définir les différents seuils utilisés, lors de l'analyse des scores et de l'élimination des appariements ambigus.

```
subtype ScoreCorrelation is float range -1.0..1.0;
SEUIL_SCORE:constant ScoreCorrelation:=0.8;
```

3.2 Le paquetage p_visio

Le paquetage p_visio est un paquetage que j'ai créé pour stocker les procédures et les fonctions qui serviront pour l'algorithme général du projet VISIO. Ce paquetage est donc l'un des plus importants.

J'ai ajouté le type MatriceMscores qui permet de travailler avec la matrice qui contiendra les appariements.

```
type MatriceMscores is array(Integer range <>,Integer range <>) of Float;
```

3.3 Le paquetage p_imageppm

Le paquetage p_imageppm est un paquetage qui va me servir à stocker les procédures et les fonctions en relation avec les images PPM. Ainsi il sera facile de le remplacer par un autre paquetage, si nous décidons d'utiliser d'autre format pour les images.

Dans le projet visio, je lis les images PPM de deux façons. D'une part en utilisant le paquetage Text_io et d'autre part en utilisant le paquetage Sequential_io afin de lire correctement toutes les valeurs des images PPM binaires (P6).

```
package monio is new sequential_io(character);
```

Chapitre 4

Description des traitements choisis

4.1 Le paquetage p_cadeau

4.1.1 Fonction ToString

```
function ToString(S:Caracteres) return String;
```

Cette fonction permet de passer du type Caracteres au type String, elle nous a été fourni avec le paquetage p_cadeau.

4.1.2 Fonction ToCaracteres

```
function ToCaracteres(S:String) return Caracteres;
```

Cette fonction permet de passer du type String au type Caracteres, elle nous a été fourni avec le paquetage p_cadeau.

4.1.3 Fonction TailleFichier

```
function TailleFichier(NomFichier:String) return Natural;
```

Cette fonction est utilisée pour savoir combien de ligne contient un fichier, je m'en sers plusieurs fois dans le programme VISIO, pour calculer la taille de la photothèque, le nombre de ligne images index, etc.

4.1.4 Procédure ChargePhototheque

```
procedure ChargerPhototheque(Debug: in String;NomFichierPhototheque:String;
Ph:out Phototheque);
```

Cette procédure permet d'enregistrer les informations présentes dans le fichier fphototheque.txt dans une structure Phototheque. Cette structure contient toutes les informations concernant les images de la photothèque.

4.2 Le paquetage p_visio

4.2.1 Procédure ChargeFichierPointsRequete

```
procedure ChargeFichierPointsRequete(Debug: in String;
NomFichierRequetePoints:String; Description:out DescripteurImage);
```

Cette procédure permet de récupérer les points de l'image requête pour les stocker dans la structure DescripteurImage. Elle permet de récupérer les coordonnées des points d'intérêts ainsi que leurs caractérisations.

4.2.2 Procédure CaracterisationPoints

```
procedure CaracterisationPoints(P:in out Point;MR,MV,MB:MatriceImage);
```

Cette procédure permet de récupérer la caractérisation de tous les points d'intérêt d'une image requête.

4.2.3 Procédure AppariementsAmbigus

```
procedure AppariementsAmbigus(Mscores:in out MatriceMscores);
```

Cette procédure est utilisée pour éliminer tous les appariements ambigus. J'ai ajouté un test pour savoir si les lignes et les colonnes étaient déjà analysées. Cette information a été donnée lors de la présentation du projet par Valérie GOUET.

4.2.4 Fonction ZNCC

```
function ZNCC(P1,P2:Point) return Float;
```

La fonction ZNCC renvoie un score, compris entre -1 et 1, qui permet de définir si les points de deux images sont fortement corrélés ou non.

4.2.5 Fonction CompareRequeteImagePhototheque

```
function CompareRequeteImagePhototheque(Requete:DescripteurImage;
IPh:DescripteurImage) return Natural;
```

Cette fonction fait appelle à la fonction ZNCC et à la procédure AppariementsAmbigus, pour calculer un vote pour chaque image de la photothèque.

4.2.6 Procédure CompareRequetePhototheque

```
procedure CompareRequetePhototheque(Debug: in String; Requete:DescripteurImage;
Ph:in out Phototheque);
```

Cette procédure permet de comparer les points de l'image requête avec ceux des images de la photothèque.

4.2.7 Procédure OrdreDecroissantDesVotes

```
procedure OrdreDecroissantDesVotes(Ph:in out Phototheque);
```

Cette procédure permet de classer par ordre décroissant des votes obtenus, les informations sur les images de la photothèque.

4.2.8 Fonction EcartMoyen

```
function EcartMoyen(Ph:Phototheque) return float;
```

Calcul de l'écart moyen pour afficher les meilleurs résultats obtenus.

4.2.9 Procédure AfficheResultats

```
procedure AfficheResultats(Ph:in out Phototheque);
```

Affiche les résultats. Cette procédure utilise la procédure OrdreDecroissantDesVotes et la fonction EcartMoyen.

4.3 Le paquetage p_imageppm

4.3.1 Fonction SupprimeCommentaires

```
function SupprimeCommentaires(Chaine : String) return String;
```

Cette fonction permet de supprimer les commentaires lors de la lecture du header de l'image PPM.

4.3.2 Fonction SupprimeEspace

```
function SupprimeEspace(Chaine: in String) return String;
```

Cette fonction permet de supprimer les espaces ou les tabulations présents en début de chaine. Elle est utilisée pour lire le header des fichiers PPM.

4.3.3 Procédure LireHeader

```
procedure LireHeader(Debug: String; NomFichier: in String;
MagicNumber: out T_Magic; Largeur,Hauteur: out Positive;
Couleur: out NiveauDeGris; NbLines: in out Integer);
```

Cette procédure permet de lire de header des fichiers PPM, pour récupérer les informations de l'image, le magic number, la largeur, la hauteur et le nombre de couleur.

4.3.4 Procédure ChargeMatriceImage

```
procedure ChargeMatriceImage(Debug: in String; NomFichier: in String;
MR,MV,MB: in out MatriceImage; LignesHeader: in Integer);
```

Cette procédure permet de récupérer les trois matrices R,V,B de l'image requête.

4.3.5 Procédure ChargeFichierIndex

```
procedure ChargeFichierIndex(NomFichierIndex:String;
Description:out DescripteurImage);
```

Cette procédure permet de charger les coordonnées des points d'intérêt, ainsi que la caractérisation de ces points.

Chapitre 5

Comment tester le programme

5.1 Les démarches à effectuer

5.1.1 Connexion sur les machines du CNAM

Pour tester le programme VISIO, il faut se connecter sur les machines du CNAM, vlad.cnam.fr puis joule.cnam.fr. Ensuite, le programme se trouve dans le répertoire PROJET à la racine de mon compte (/etudiants/deptinfo/f/fraval_b/).

```
brian@fraval:~$ ssh fraval_b@vlad.cnam.fr -t ssh joule
fraval_b@vlad.cnam.fr's password:
fraval_b@joule's password:
Last login: Fri Jun 20 00:15:36 2003 from vlad.cnam.fr
Sun Microsystems Inc.   SunOS 5.9       Generic May 2002
@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#
Des informations pour les eleves sont disponibles dans les pages WEB de la DSI :
http://www.cnam.fr/eleves/
@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#
@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@#@@#@#

xset: Command not found
Fri Jun 20 00:15:58 CEST 2003
ENVIRONNEMENT JAVA 1.2.2 CHARGE
Environnement modifie pour utiliser gnat 3.14...
bash-2.02$ pwd
/etudiants/deptinfo/f/fraval_b
bash-2.02$ cd PROJET/
```

5.1.2 Compilation du programme

Normalement le programme est déjà compilé, mais pour en être sûr il vaut mieux lancer de nouveau la compilation du programme. Cela permet de s'assurer qu'il n'y a pas d'erreur lors de la compilation.

Pour automatiser la compilation du programme, j'ai crée un Makefile. Donc il suffit de lancer la commande suivante :

```
bash-2.02$ make
gnatmake -gnatf visio
gcc -c -gnatf visio.adb
gcc -c -gnatf p_cadeau.adb
gcc -c -gnatf p_imageppm.adb
gcc -c -gnatf p_visio.adb
gnatbind -x visio.ali
gnatlink visio.ali
```

5.1.3 Lancer le programme

Pour lancer le programme, il faut préciser qu'elle est l'image requête. Bien entendu, je test s'il y a bien une image requête passée en argument au programme.

Lancer la commande suivante :

```
bash-2.02$ ./visio

USAGE :: ./visio <fichier requete> (use -d or -D for mode debug)
```

Comme vous pouvez le constater, il y a un mode debug [7] qui permet de suivre le travail réalisé par le programme. Ce mode peut être activé en ajoutant -d ou -D à la ligne de commande.

Voici un exemple d'utilisation du programme en mode debug :

```
bash-2.02$ ./visio -d irequetes_files/chazal.ppm

-------------------------------------------
Le mode debug est activé.
-------------------------------------------

1) Nombre d'image dans la phototheque :  11

2) Chargement de la phototèque... OK

3) Nombre de points à analyser dans l'image requete :  298

4) Chargement des points(x,y) de l'image requete... OK

5) Lire le header du fichier requete... OK

P6 400 300 255 4

6) Caracterisation des points de l'image requete.

7) Compare les points de l'image requete avec ceux des images de la photothèque
```

```
1- Lecture du fichier iphototheque_files/cereales.ppm.ind,
   puis compare les points avec l'image requete... OK
2- Lecture du fichier iphototheque_files/logo_cnn.ppm.ind,
   puis compare les points avec l'image requete... OK
3- Lecture du fichier iphototheque_files/logo_france3.ppm.ind,
   puis compare les points avec l'image requete... OK
4- Lecture du fichier iphototheque_files/logo_planete.ppm.ind,
   puis compare les points avec l'image requete... OK
5- Lecture du fichier iphototheque_files/plateau_tf1.ppm.ind,
   puis compare les points avec l'image requete... OK
6- Lecture du fichier iphototheque_files/poisson.ppm.ind,
   puis compare les points avec l'image requete... OK
7- Lecture du fichier iphototheque_files/texture_lavande.ppm.ind,
   puis compare les points avec l'image requete... OK
8- Lecture du fichier iphototheque_files/texture_mer.ppm.ind,
   puis compare les points avec l'image requete... OK
9- Lecture du fichier iphototheque_files/texture_piscine.ppm.ind,
   puis compare les points avec l'image requete... OK
10- Lecture du fichier iphototheque_files/tournesol.ppm.ind,
   puis compare les points avec l'image requete... OK
11- Lecture du fichier iphototheque_files/vase.ppm.ind,
   puis compare les points avec l'image requete... OK


-------------------------------------------
Resultat de la recherche :
-------------------------------------------
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :  25%
Mer                 iphototheque_files/texture_mer.ppm, vote :  23%
-------------------------------------------
Piscine             iphototheque_files/texture_piscine.ppm, vote :   4%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Poisson coloré      iphototheque_files/poisson.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Tournesol           iphototheque_files/tournesol.ppm, vote :   0%
Vase chilien        iphototheque_files/vase.ppm, vote :   0%
```

Si le mode debug n'est pas utilisé, il n'y aura pas tous les messages d'information au cours de l'exécution du programme. Seul le résultat de l'analyse de l'image requête, avec celles de la photothèque, sera affiché.

```
bash-2.02$ ./visio irequetes_files/chazal.ppm


-------------------------------------------
Resultat de la recherche :
```

```
-------------------------------------------
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :  25%
Mer                 iphototheque_files/texture_mer.ppm, vote :  23%
-------------------------------------------
Piscine             iphototheque_files/texture_piscine.ppm, vote :   4%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Poisson coloré      iphototheque_files/poisson.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Tournesol           iphototheque_files/tournesol.ppm, vote :   0%
Vase chilien        iphototheque_files/vase.ppm, vote :   0%
```

5.2 Les images utilisées

FIG. 5.1 – Cereales table

```
-------------------------------------------
Resultat de la recherche :
-------------------------------------------
Poisson coloré      iphototheque_files/poisson.ppm, vote :   6%
-------------------------------------------
Tournesol           iphototheque_files/tournesol.ppm, vote :   3%
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   2%
Vase chilien        iphototheque_files/vase.ppm, vote :   1%
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Mer                 iphototheque_files/texture_mer.ppm, vote :   0%
Piscine             iphototheque_files/texture_piscine.ppm, vote :   0%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
```

FIG. 5.2 – Chazal

```
-------------------------------------------
Resultat de la recherche :
-------------------------------------------
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :  25%
Mer                 iphototheque_files/texture_mer.ppm, vote :  23%
-------------------------------------------
Piscine             iphototheque_files/texture_piscine.ppm, vote :   4%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Poisson coloré      iphototheque_files/poisson.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Tournesol           iphototheque_files/tournesol.ppm, vote :   0%
Vase chilien        iphototheque_files/vase.ppm, vote :   0%
```

FIG. 5.3 – Lavande arbre

```
---------------------------------------------
Resultat de la recherche :
---------------------------------------------
Mer                 iphototheque_files/texture_mer.ppm, vote :  86%
---------------------------------------------
Piscine             iphototheque_files/texture_piscine.ppm, vote :  60%
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :  28%
Vase chilien        iphototheque_files/vase.ppm, vote :   1%
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Poisson coloré      iphototheque_files/poisson.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Tournesol           iphototheque_files/tournesol.ppm, vote :   0%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
```

FIG. 5.4 – Militaire cnn

```
-------------------------------------------
Resultat de la recherche :
-------------------------------------------
Poisson coloré      iphototheque_files/poisson.ppm, vote :   3%
-------------------------------------------
Tournesol           iphototheque_files/tournesol.ppm, vote :   1%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :   0%
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Mer                 iphototheque_files/texture_mer.ppm, vote :   0%
Piscine             iphototheque_files/texture_piscine.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Vase chilien        iphototheque_files/vase.ppm, vote :   0%
```

FIG. 5.5 – Piscine mer logo

```
-------------------------------------------
Resultat de la recherche :
-------------------------------------------
Mer                 iphototheque_files/texture_mer.ppm, vote :  20%
-------------------------------------------
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :  12%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Poisson coloré      iphototheque_files/poisson.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Piscine             iphototheque_files/texture_piscine.ppm, vote :   0%
Tournesol           iphototheque_files/tournesol.ppm, vote :   0%
Vase chilien        iphototheque_files/vase.ppm, vote :   0%
```

FIG. 5.6 – Piscine mer

```
---------------------------------------------
Resultat de la recherche :
---------------------------------------------
Mer                 iphototheque_files/texture_mer.ppm, vote :  21%
---------------------------------------------
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :  10%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Poisson coloré      iphototheque_files/poisson.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Piscine             iphototheque_files/texture_piscine.ppm, vote :   0%
Tournesol           iphototheque_files/tournesol.ppm, vote :   0%
Vase chilien        iphototheque_files/vase.ppm, vote :   0%
```

FIG. 5.7 – Poisson etagere

```
---------------------------------------------
Resultat de la recherche :
---------------------------------------------
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :   0%
Poisson coloré      iphototheque_files/poisson.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Mer                 iphototheque_files/texture_mer.ppm, vote :   0%
Piscine             iphototheque_files/texture_piscine.ppm, vote :   0%
Tournesol           iphototheque_files/tournesol.ppm, vote :   0%
Vase chilien        iphototheque_files/vase.ppm, vote :   0%
```

FIG. 5.8 – Porte tournesol

```
-------------------------------------------
Resultat de la recherche :
-------------------------------------------
Mer                 iphototheque_files/texture_mer.ppm, vote :  46%
-------------------------------------------
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :  38%
Vase chilien        iphototheque_files/vase.ppm, vote :   1%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Poisson coloré      iphototheque_files/poisson.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Piscine             iphototheque_files/texture_piscine.ppm, vote :   0%
Tournesol           iphototheque_files/tournesol.ppm, vote :   0%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
```

FIG. 5.9 – ppda

```
-------------------------------------------
Resultat de la recherche :
-------------------------------------------
Piscine             iphototheque_files/texture_piscine.ppm, vote :   4%
-------------------------------------------
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :   2%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Poisson coloré      iphototheque_files/poisson.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Mer                 iphototheque_files/texture_mer.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Tournesol           iphototheque_files/tournesol.ppm, vote :   0%
Vase chilien        iphototheque_files/vase.ppm, vote :   0%
```

FIG. 5.10 – Tournesols

```
-------------------------------------------
Resultat de la recherche :
-------------------------------------------
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :   0%
Poisson coloré      iphototheque_files/poisson.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Mer                 iphototheque_files/texture_mer.ppm, vote :   0%
Piscine             iphototheque_files/texture_piscine.ppm, vote :   0%
Tournesol           iphototheque_files/tournesol.ppm, vote :   0%
Vase chilien        iphototheque_files/vase.ppm, vote :   0%
```

FIG. 5.11 – Vase cuisine

```
-------------------------------------------
Resultat de la recherche :
-------------------------------------------
Boîte de céréales   iphototheque_files/cereales.ppm, vote :   0%
Image CNN           iphototheque_files/logo_cnn.ppm, vote :   0%
Image France 3      iphototheque_files/logo_france3.ppm, vote :   0%
Image Planète       iphototheque_files/logo_planete.ppm, vote :   0%
Image TF1           iphototheque_files/plateau_tf1.ppm, vote :   0%
Poisson coloré      iphototheque_files/poisson.ppm, vote :   0%
Champ de lavande    iphototheque_files/texture_lavande.ppm, vote :   0%
Mer                 iphototheque_files/texture_mer.ppm, vote :   0%
Piscine             iphototheque_files/texture_piscine.ppm, vote :   0%
Tournesol           iphototheque_files/tournesol.ppm, vote :   0%
Vase chilien        iphototheque_files/vase.ppm, vote :   0%
```

Chapitre 6

Conclusion

6.1 Les problèmes non résolus

L'algorithme complet du programme a été réalisé, mais j'ai des problèmes sur les résultats obtenus. Normalement, je devrais avoir un taux de vote de confiance (vote) de 100%, pour les images de la photothèque qui ressemblent le plus à l'image requete.

Je n'ai pas réussi à résoudre ce problème pour l'instant.

6.2 Les améliorations possibles

Pour améliorer le programme VISIO, je pourrai ajouter le traitement de plusieurs type d'image et notamment un traintement sur des images compressées comme les images jpeg. Cela permettrai d'utiliser des images beaucoup moins grosses. Je n'ai pas eu le temps d'utiliser la librairie SimpleJpeg_Lib de Patrice FREYDIERE, mais j'ai commencé à l'analyser.

Ensuite, il est possible d'améliorer la méthode de caractérisation des points, pour notamment gérer la rotation des images. Il est par exemple possible de caractériser un point et son voisinage en calculant l'histogramme couleur de la région.

Pour avoir plus d'informations sur les évolutions possibles de ce programme, je vous conseille de lire la page consacrée à la formation DEA ESTC - Option CAM - Filière Vision par Ordinateur[9] proposée par Valérie GOUET.

Chapitre 7

Remerciements

Je tiens à remercier toutes les personnes de la mailing liste Ada France (ada-france@ada-france.org), ainsi que toutes les personnes du forum de discussion fr.comp.lang.ada, pour leurs aides précieuses sur le language Ada.

Et plus particulièrement Gilles DEMAILLY qui m'a beaucoup aidé sur la lecture des fichiers PPM, ainsi que Patrice FREYDIERE qui m'a fourni des informations sur la librairie SimpleJpeg_Lib [8].

J'aimerai aussi remercier Valérie GOUET, Alexandre TOPOL, et Mohamed HALLAB qui ont su être à mon écoute pendant toute la réalisation de ce projet.

Et pour finir, je remercie Frédéric NIAS, pour sa correction humouristique et consciencieuse de ce document.

Bibliographie

[1] Site de l'unité de valeur VARI - Valeur d'Acceuil et de Reconversion à l'Informatique.
http:// deptinfo.cnam.fr/ Enseignement/ CycleProbatoire/ Vari/

[2] Page d'information concernant le projet de fin d'année de l'unité de valeur VARI - Valeur d'Accueil et de Reconversion à l'Informatique.
http:// cedric.cnam.fr/ ~gouet/ Vari/

[3] Page d'information sur la programmation du projet.
http:// cedric.cnam.fr/ ~gouet/ Vari/ programmation.html

[4] FAQ - Frequently Asked Questions du groupe de discussion fr.comp.lang.ada
http:// www.rfc1149.net/ fcla/

[5] Explication de l'algorithme du projet.
http:// cedric.cnam.fr/ ~gouet/ Vari/ principe.html

[6] Zero Mean Normalized Cross Correlation - Méthodes de comparaison des points caractérisés.
http:// cedric.cnam.fr/ ~gouet/ Vari/ principe.html

[7] Gestion du mode debug dans l'application VISIO
http:// www.adapower.com/ reuse/ debug.html

[8] Patrice Freydiere auteur de la librairie SimpleJpeg_Lib
http:// pfreydiere.free.fr/ SimpleJpegLib.html

[9] DEA ESTC - Option CAM - Filière Vision par Ordinateur.
http:// cedric.cnam.fr/ ~gouet/ FVO/