



HAL
open science

Utilisation d'images réelles pour la production d'images synthétiques dans un contexte audiovisuel

Jean-Yves Doucet

► **To cite this version:**

Jean-Yves Doucet. Utilisation d'images réelles pour la production d'images synthétiques dans un contexte audiovisuel. Traitement des images [eess.IV]. 1989. dumas-00335950v1

HAL Id: dumas-00335950

<https://dumas.ccsd.cnrs.fr/dumas-00335950v1>

Submitted on 31 Oct 2008 (v1), last revised 31 Oct 2008 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE AGREE

DE GRENOBLE (C.U.E.F.A.)

MEMOIRE

présenté en vue d'obtenir

le **DIPLOME D'INGENIEUR C.N.A.M.**

en

INFORMATIQUE

par

JEAN-YVES DOUCET

Utilisation d'images réelles pour la production d'images synthétiques dans
un contexte audiovisuel.

SOUTENU LE : 26 JUIN 1989

JURY

Président : Mr Claude KAISER

**Membres : Mr Louis BOLLIET
Mr Jacques COURTIN
Mr Philippe BOULLE
Mr Francis MARTINEZ**

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE AGREE

DE GRENOBLE (C.U.E.F.A.)

MEMOIRE

présenté en vue d'obtenir

le **DIPLOME D'INGENIEUR C.N.A.M.**

en

INFORMATIQUE

par

JEAN-YVES DOUCET

Utilisation d'images réelles pour la production d'images synthétiques dans un contexte audiovisuel.

Les travaux relatifs au présent mémoire ont été effectués dans la société **GETRIS IMAGES** sous la direction de Monsieur **F. MARTINEZ**.

REMERCIEMENTS

Je remercie Mr C. KAISER, professeur au Conservatoire National des Arts et Métiers de Paris d'avoir bien voulu présider le jury de ce mémoire, de même que Mr J. COURTIN professeur à l'université des Sciences Sociales de Grenoble, Mr L. BOLLIET professeur émérite à l'université des Sciences Sociales de Grenoble, Mr P. BOULLE ingénieur à la société GETRIS-IMAGES qui ont accepté de participer à ce jury.

Je voudrais remercier Mr MARTINEZ, professeur à l'université Joseph FOURIER de GRENOBLE, Responsable du groupe de Développement de la société GETRIS IMAGES, de l'aide et du soutien technique qu'il m'a apporté durant ce travail ainsi que dans l'élaboration de ce document.

J'exprime toute ma reconnaissance à Mr MIRIBEL, président directeur général de la société GETRIS-IMAGES, pour m'avoir accueilli durant toute cette année et qui m'a permis de travailler dans les meilleures conditions.

J'associerais à ces remerciements tout le personnel de cette société, et surtout Mlle CHALVIN qui m'a initié aux différentes applications ATALIS et Mr MOLLIE qui m'a donné de précieux conseils.

Je tiens à exprimer toute ma gratitude aux enseignants du C.U.E.F.A. sans qui, ce mémoire n'aurait pas pu avoir lieu et tout particulièrement à deux d'entre eux, Messieurs BOLLIET et LIANDRAT pour qui le terme de Formation Continue prend toute son importance.

Je voudrais que ce mémoire soit un signe d'encouragement pour les auditeurs, afin que le maximum d'entre eux entreprennent ce que j'ai eu la chance de réaliser cette année et poursuivent leurs efforts jusqu'au bout.

Je n'obmettrai pas non plus de remercier Mr GUERRE, pour m'avoir laissé consulter à loisirs la banque de données de la société MERLIN-GERIN.

Ce rapport n'est pas le fruit d'un travail personnel et c'est pourquoi je voudrais le partager avec ma femme et mes deux enfants, qui m'ont soutenu pendant toute cette formation et qui ont sacrifié une partie de leurs loisirs pour que ce document soit ce jour entre vos mains.

SOMMAIRE

Introduction

Chapitre 1 : Cadre du projet : la société GETRIS IMAGES

1. Le marché aux images
2. La société GETRIS IMAGES
3. Présentation du système ATALIS
 - 3.1. Présentation matérielle
 - 3.1.1. Description du système ATALIS
 - 3.1.2. Particularités du système ATALIS
 - 3.2. Présentation logicielle
 - 3.3. Présentation de l'équipe

Chapitre 2 : Présentation du sujet, objectifs et contraintes

1. Le pourquoi de cette étude
 - 1.1. Exemple 1
 - 1.2. Exemple 2
 - 1.3. Exemple 3
 - 1.4. Exemple 4
2. Objectifs
3. Contraintes

Chapitre 3 : Etude théorique, expérimentations et choix

1. La couleur en synthèse d'images
 - 1.1. Qu'est ce que la couleur ?
 - 1.2. Espaces de couleurs
 - 1.2.1. Modèles sur la théorie trichromatique
 - 1.2.2. Modèles basés sur la perception
2. La vision artificielle
 - 2.1. Définition du contexte de l'application
 - 2.1.1. Contexte idéal en vision
 - 2.1.2. Notre contexte
 - 2.2. Qu'est ce qu'un système de vision artificielle ?
 - 2.2.1. Histogramme
 - 2.2.2. Saisie d'image
 - 2.2.3. Analyse de l'image
 - 2.2.4. Traitement de l'image

3. Prétraitement

3.1. Traitement point par point

- 3.1.1. Egalisation
- 3.1.1. Couleurs normalisées

3.2. Traitement spatial

- 3.2.1. Filtre passe-bas
- 3.2.2. Autres filtres

3.3. Autres prétraitements

4. Segmentation de l'image

4.1. Seuillage

4.2. Apprentissage

4.3. Seuillage relatif à une couleur

- 4.3.1. Seuillage RVB par moyenne
- 4.3.2. Seuillage RVB par rapport à un ensemble de composantes
- 4.3.3. Seuillage RVB par rapport à un ensemble de prototypes
- 4.3.4. Seuillage TLS

4.4. Seuillage par l'analyse de l'aspect de surface

4.4.1. Définition de la texture

4.4.2. Seuillage par variance

4.4.2.1. Rappels statistiques

4.4.2.2. Mise en œuvre du seuillage par variance

4.4.3. Seuillage par différence d'images

4.4.3.1. Théorie

4.4.3.2. Mise en œuvre

4.4.5. Autres techniques d'analyse de texture

4.5. Analyse du contour

4.5.1. Seuillage par variation de gradient

4.5.1.1. Définition du gradient de la fonction image

4.5.1.2. Pondération du gradient

4.5.1.3. Mise en oeuvre du seuillage par variation de gradient

4.5.2. Suivi de contours sur une image couleur

4.5.2.1. Théorie

4.5.2.2. Exemple

4.5.2.3. Traitement d'une image couleur

4.5.3. Suivi de contours sur une image binaire

4.5.3.1. Par opérateur gradient

4.5.3.2. Autres techniques de suivi de contours

4.5.3.3. Avantages du suivi de contours

5. Post-traitement

6. Notre solution : le logiciel G_EXTRA

Chapitre 4 : Réalisation

1. Configuration en plans images

2. Description de l'interface Homme - Machine

2.1. Généralités

2.2. Zone Menu

2.2.1. Fonctions permanentes et claviers

2.2.2. Description de la zone MENU

2.3. Zone de CLIC

2.3.1. Saisie de zones

2.3.2. Saisie d'informations "images"

3. Description des menus

3.1. Fonctions classiques

3.1.1. Restitution

3.1.2. Sauvegarde

3.1.3. Effacement

3.1.4. Fin

3.2. Extraction

3.3. Détourage

3.4. Correction

3.5. Copie

4. Réalisation de l'interface homme-machine

4.1. Généralités

4.2. Programme principal

4.3. Gestion du curseur

4.3.1. Affichage du curseur

4.3.2. Saisie d'informations

4.4. Affichage des morphologies

4.4.1. COPIER-MORPH

4.4.2. EFFACEMENT-MORPH

4.4.3. AFFICH-MORPH

4.4.4. VALID-MORPH

4.5. Saisie des morphologies

4.6. Gestion de l'action courante

4.7. Gestion des rubriques

5. Fonctions spécifiques du traitement d'images

5.1. Fonctions relatives à l'extraction

5.1.1. Saisie de prototypes

5.1.2. Exécution du seuillage

5.1.2.1. Seuillage RVB 12 bits ou seuillage TLS

5.1.2.2. Seuillage par rapport aux composantes RVB 24 bits

5.1.2.3. Seuillage RVB 24 bits

5.2. Fonctions relatives au détournage

5.2.1. Principe de l'algorithme de ROSENFELD et KAK

5.2.2. Algorithme de tâche

5.2.3. Adaptation de l'algorithme de ROSENFELD et KAK

5.3. Conversion RVB \Leftrightarrow TLS

5.3.1. Formules de conversion

5.3.2. Optimisation RVB \Rightarrow TLS

5.3.3. Optimisation TLS \Rightarrow RVB

5.3.4. Gain en temps de réponse

5.4. Autres fonctions

Conclusion

Annexes

annexe 1 : Caractéristiques du PC CompaQ 386 S

annexe 2 : Caractéristiques de l'application G_EXTRA

annexe 3 : Comparaison des vitesses de traitement des algorithmes d'extraction

annexe 4 : Photographies d'écran

Bibliographie

INTRODUCTION

Le travail décrit dans ce document, consiste à concevoir un système permettant d'utiliser des éléments extraits d'images réelles numérisés en couleur, pour faciliter et améliorer la production d'images synthétiques statiques ou dynamiques, dans un contexte audiovisuel.

Le chapitre I est une brève description de la société GETRIS-IMAGES qui m'a accueilli, de son matériel ainsi que de ses réalisations logicielles.

Le chapitre II définit les objectifs de notre étude. Il montre les différentes utilisations possibles du logiciel à créer. Dans sa seconde partie, il dresse la liste des contraintes à respecter lors de la phase de réalisation, qui sont surtout dues au contexte audiovisuel.

Le chapitre III dresse un inventaire des techniques du traitement d'images. Les méthodes qui nous semblent intéressantes ont été expérimentées et les conclusions par rapport à notre problème, y sont données. A la fin du chapitre, nous présentons notre solution qui est le logiciel **G-EXTRA**.

Le Chapitre IV décrit la réalisation proprement dite de cette étude. Il s'attache à montrer comment nous avons résolu les contraintes de l'interface homme-machine, définies au chapitre II. Nous y détaillons les algorithmes de traitement d'images utilisés dans cette application.

Une conclusion reprend l'ensemble de ce travail. Elle donne aussi les évolutions possibles et souhaitables de ce logiciel.

En annexe, nous montrons par quelques photographies, des exemples de fonctionnement et d'utilisation de notre application.

CHAPITRE 1

CADRE DU PROJET

LA SOCIETE : GETRIS IMAGES

CADRE DU PROJET : LA SOCIETE GETRIS IMAGES

1. Le marché aux images

Depuis dix ans, l'ordinateur s'est emparé du monde de l'image. Une nouvelle science est née, elle a pour nom l'**Infographie**, c'est à dire l'application de l'informatique à la représentation et au traitement de l'image (SEY87).

Le marché qu'elle recouvre est vaste. Il s'évalue déjà en milliards de francs. Il s'étend dans des domaines variés comme (BRA88):

- .modélisation : recherche médicale, architecture ...;
- .simulation : aéronautique, "design" industriel ...;
- .graphisme de gestion : visualisation des données ...;
- .conception de produits industriels appelée couramment C.A.O.;
- .création graphique (DAL87).

La génération d'effets spéciaux, les montages vidéos, les spots publicitaires, les génériques de télévision et bien sûr le dessin animé constituent cette dernière catégorie et sont très friands des "images informatiques".

L'artiste modifie ses images en temps réel, par simples changements d'informations traduits dans l'instant à l'écran. Il crée le graphisme à la main à partir de sa table à digitaliser. L'image est tellement réaliste, qu'il est impossible de différencier les images de synthèses des images réelles.

Quelques sociétés françaises se sont lancées sur cette part de marché porteur. Dans le matériel milieu de gamme, nous pouvons citer en outre XCOM, la GIXI et **GETRIS IMAGES**.

2. LA SOCIETE GETRIS IMAGES

Issu des recherches universitaires, et en particulier de l'institut d'informatique et mathématiques appliquées de GRENOBLE (IMAG), la société GETRIS IMAGES commercialise depuis 1985, un système graphique complet ATALIS (matériels et logiciels), facilement configurable .

Malgré sa jeunesse, cette société s'est imposée dans le monde de l'infographie, en effectuant un compromis entre le temps réel et une qualité d'images, tout en soignant la partie RHM (relations homme-machine), c'est à dire l'interactivité.

Le monde de l'audiovisuel bien sûr et surtout la production d'animation de synthèse en vidéo, celui de la communication sans oublier les industries manipulant les couleurs (textile, vente par catalogue ...), constituent leurs principaux clients.

3. PRESENTATION DU SYSTEME ATALIS

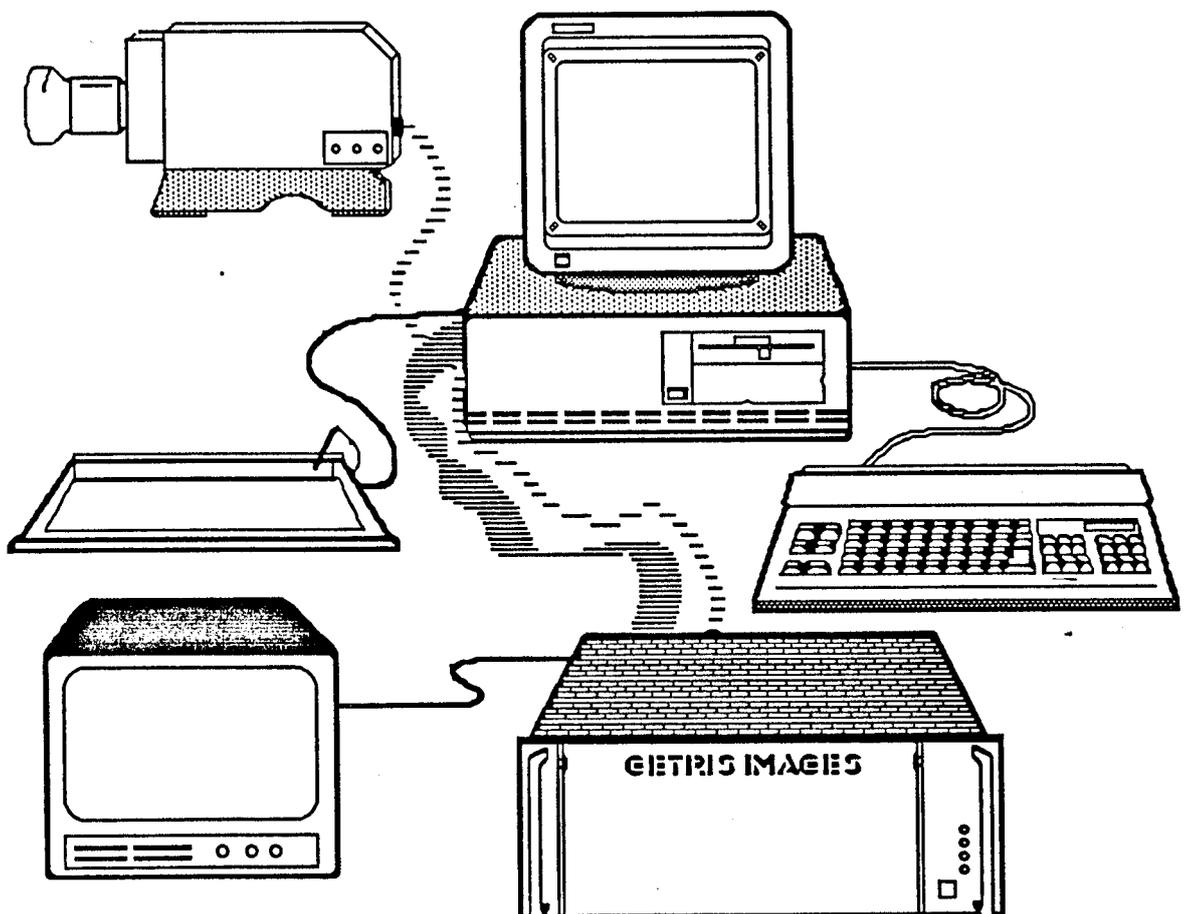
Le système ATALIS permet la création et l'animation des images sur ordinateur du graphisme le plus simple (dessins) à une réalisation sophistiquée (animation , effets visuels...) par l'intermédiaire de menus simples d'emploi.

3.1. PRESENTATION MATERIELLE

3.1.1 Description du système ATALIS

Le système ATALIS se divise en quatre parties (FOU88) (ATA88):

- .une unité de calcul qui exécute les programmes;
- .un rack de synthèse ou unité vidéo = ensemble de cartes spécifiques qui traite les informations numériques;
- .un dispositif de dialogue = tablette ou souris;
- .une unité de visualisation.



Poste ATALIS

Le système ATALIS est piloté par un ordinateur hôte qui est soit un PC/AT type 80286, soit 80386. Du fait de sa lenteur relative, il ne gère que les entrées-sorties, c'est à dire les ordres. Les calculs vectoriels sont réalisés par des processeurs spécialisés type 80287 ou 80387. Les traitements 3D sont effectués sur des processeurs plus performants comme les 68020 ou 68030. Une carte spécifique **GPR**, basé sur un ADSP-2100 (ANALOG DEVICES), permet le traitement rapide du signal, en utilisant les possibilités d'une architecture parallèle (bus spécifique), mémoires caches, opérateurs cablés (accélérateur). Le système obtient ainsi une vitesse de calcul aussi élevée qu'avec les stations de travail spécialisées.

Une carte **GAT**, qui n'est autre qu'une interface parallèle rapide avec mémoire tampon, accroît considérablement le débit de transfert entre le ordinateur hôte et les cartes spécifiques.

Suivant l'utilisation, le rack de synthèse est constitué :

- .d'une carte d'acquisition vidéo appelée **GAV**. Elle réalise la saisie d'images à partir d'une caméra;

- .d'un ou plusieurs plans mémoires **GPM** qui contiennent des images 1024*1024 pixels, soit en 24 bits, c'est à dire en 16 millions de couleurs, soit en 12 bits pour 4096 couleurs parmi 16 millions;

- .d'une carte **GCV** qui dialogue avec le ordinateur hôte et qui permet la synchronisation avec une régie vidéo;

- .d'une carte **VTR Contrôleur** . C'est une carte de transfert qui permet les enregistrements sur magnétoscopes (pilotage).

L'unité de visualisation est un moniteur vidéo RVB haute résolution (720x 574 points image), qui convertit les données numériques en signal vidéo.

3.1.2. Particularités du système ATALIS

Le système ATALIS fournit des opérations cablées telles que :

- .translation d'objets dans une image;
- .modification des couleurs par une table de couleur;
- .visibilité d'une couleur ou d'une image (plan mémoire);
- .modification de l'ordonnancement des images;
- .possibilité de fenêtrage sur une image;
- .gel numérique des informations issues de l'acquisition vidéo.

La modification d'une image ne nécessite pas de traitements compliqués au niveau du calculateur hôte, mais seulement l'interprétation et l'exécution d'une seule commande par la machine (gain en rapidité) . Nous ne recalculons pas l'image à chaque affichage. De plus, cette commande est rapide et peut être exécutée pendant les temps morts, qui correspondent au retour de trames (40 millisecondes). Ceci est particulièrement intéressant pour des animations en temps réel. A titre d'exemple, l'animation est créée en utilisant le fenêtrage sur plusieurs zones de l'image, où sont mémorisés les différentes phases du mouvement.

:

3.2. Présentation logicielle

Les principaux logiciels standards, écrits en langage C à partir de la chaîne MICROSOFT™, supportés par ce système sont :

.G_PAINT = palette graphique qui permet à l'artiste de définir l'épaisseur du trait, sa couleur et son aspect (pinceau, aérographe), de créer des figures géométriques telles que le cercle ou le rectangle, de remplir des surfaces, d'améliorer son dessin avec des opérations comme la correction, d'effacer ou de recopier tout ou une partie de celui-ci;

.G_ANIM = Logiciel d'animation 2D multi-acteurs exploitant les possibilités temps réel du système ATALIS telles que le déplacement d'acteurs (translation, fenêtrage), la superposition (profondeur des plans mémoires) et des effets visuels (modification des couleurs temps réel) (GAN88);

.DGS 2.1 de Digital Arts = animation 3D (texture, éclairage scénario d'animation et montage sur magnétoscope ...);

.GTI = Logiciel de traitement d'images (filtre, gradient, moyenne, seuillage ...);

.G_COUL = Logiciel d'analyse et de traitement de couleurs. Les transformations de couleurs sont obtenues par action sur les composantes de teinte, luminance et saturation.

Un des atouts de ce système est de posséder une bibliothèque de haut niveau, permettant de réaliser des fonctions graphiques évoluées et de rendre aisé tous les traitements de dialogue (cf description de l'interface homme-machine).

3.3. Présentation de l'équipe

Le service de recherche et de développement auquel j'appartiens :

- . complète cette bibliothèque

- . crée des logiciels standards

- . étudie de nouveaux systèmes (nouvelle génération de machines) plus performants pour préparer l'avenir.

L'étude que je réalise dans le cadre d'un mémoire du Conservatoire National des Arts et Métiers, doit aboutir à la création d'un logiciel standard pour compléter la gamme existante.

CHAPITRE 2

PRESENTATION DU SUJET

OBJECTIFS

ET

CONTRAINTES

1. LE POURQUOI DE CETTE ETUDE

L'étude décrite ci-après n'est pas une étude "gratuite", c'est à dire qu'elle se doit de répondre à un besoin réel, ressenti par les utilisateurs graphistes. Montrons par quelques exemples simples, les applications futures de ce logiciel.

1.1. Exemple 1

Imaginons la préparation d'un prototype de catalogue vestimentaire. Avant, les séances de prises de vues étaient obligatoires, longues et bien sûr coûteuses (photographe, mannequins, créations de décors, voyages..). Maintenant rien de tout cela! Il suffit pour juger immédiatement les effets, de constituer les différentes vues, en prélevant des morceaux de décors ou de personnages sur d'autres images, et de les assembler judicieusement comme un puzzle!

De même, certains décors sont souvent utilisés (voitures, maisons..). Le travail du graphiste est minimisé lorsque le prélèvement de l'objet est facilement reproduit (mémorisation des différentes opérations).

1.2. Exemple 2

Nouveau contexte. Nous sommes dans une régie-vidéo. Pour permettre au téléspectateur de mieux suivre un reportage, le producteur met en valeur un acteur ou un décor, en isolant celui-ci ou en modifiant sa coloration. Le tour est joué!

1.3. Exemple 3

Le scénario d'un film publicitaire consiste à faire déplacer un véhicule entre une forêt et une montagne. Quoi de plus facile, aujourd'hui! Il suffit au réalisateur de posséder une image représentant le décor. Il isole la montagne sur un premier plan, fait déplacer le véhicule sur le second plan, tandis que l'image intégrale se situe sur le plan le plus profond (le moins visible).

1.4. Exemple 4

Nous sommes confrontés au même problème à ceci près que le décor est lui aussi animé. Le principe reste identique. L'opérateur isole la montagne sur la première image. Sur les images suivantes, il indique simplement les déplacements de la prise de vue. L'outil réalise alors automatiquement l'identification de cette montagne .

2. OBJECTIFS

Deux approches différentes sont à l'origine de l'infographie (DAL87) :

- .la synthèse d'images = dessiner des objets;
- .le traitement d'images = reconnaître des objets.

La synthèse d'images a abouti à la création de logiciels sophistiqués comme la palette graphique qui facilite la vie du graphiste et lui permet des réalisations artistiques irréalisables il y a à peine une dizaine d'années.

Le traitement d'images, quant à lui, permet l'exploitation proprement dite des images. Celles-ci sont obtenues par une caméra classique, numérisées puis stockées en mémoire, où elles subissent différents traitements :

- .reconnaissance de formes;
- .calculs dimensionnels;
- .contrôle de qualité (état des surfaces ..).

Les domaines d'utilisation du traitement d'images sont nombreux :

- .analyse d'images satellites;
- .vision en robotique;
- .recherche médicale ...

Pourquoi ne pas rapprocher ces deux techniques a priori si différentes? Pourquoi ne pas utiliser les principes de bases du traitement d'images pour "étouffer" les possibilités de création de la synthèse d'images? Pourquoi ne pas utiliser une parcelle ("objet") d'une image réelle dans la composition d'une image de synthèse. Pourquoi dessiner une image qui existe déjà? La photo d'un arbre est toujours plus réaliste que le meilleur dessin! Telles sont les questions à l'origine de cette étude.

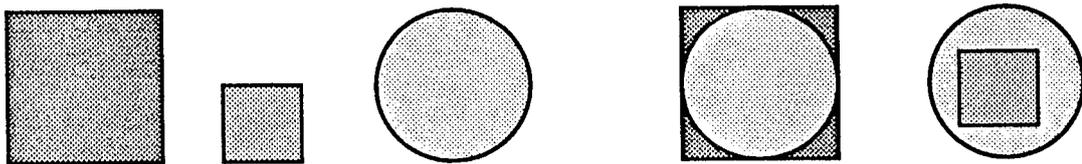
De même ce qu'il est possible de réaliser avec des images réelles doit être aussi réalisable avec des images de synthèses. L'image de synthèse devient alors un panachage ou mixage :

- .de création (il en faudra toujours !);
- .d'objets de synthèse;
- .d'objets réels.

Pour arriver à ce résultat, il faut pouvoir **extraire**, c'est à dire isoler un élément dans une image. Le mot clé de cette étude est laché, il s'agit d'**extraction**.

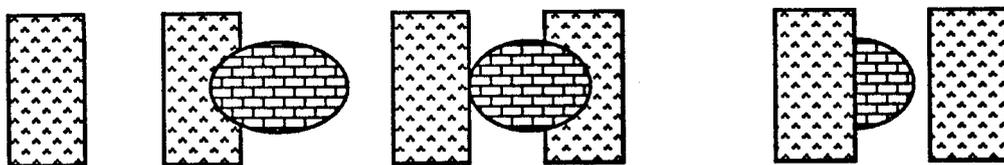
Les possibilités de cette extraction sont nombreuses suivant son utilisation. A partir des exemples donnés précédemment, nous pouvons en dresser une liste.

L'infographiste crée à partir d'images réelles, une base de données d'objets. Ceux-ci sont mémorisés et, sur demande sont insérés dans des images. L'opérateur a la possibilité de jouer sur les *paramètres de profondeur* des objets pour réaliser ses images. De tels procédés sont souvent utilisés en audiovisuel pour la création de décors. Il est possible de même, en jouant sur ces paramètres, d'isoler des objets similaires mais situés à des distances différentes.



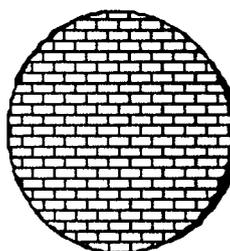
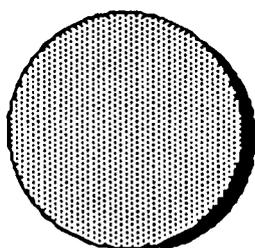
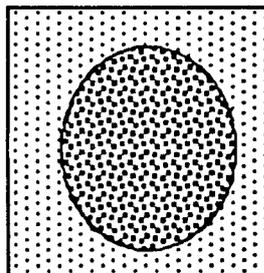
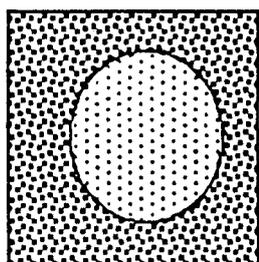
Exemple de combinaisons de décors à partir de trois objets

Il vient tout de suite à l'esprit de créer des séquences d'*animation* en déplaçant l'objet isolé dans l'image..



Exemple d'animations à partir de trois objets

Le graphiste modifie le rendu de l'objet extrait, en jouant sur ses couleurs ou sur sa forme. Pour ce fait, il faut extraire les *paramètres de luminance et de chrominance*. Il est alors possible de conserver l'apparence de l'objet (en particulier son ombre) tout en modifiant sa teinte.



Exemple de modification de rendu d'une image

Un objet n'est pas composé d'une couleur unique dans la majorité des cas. L'analyse de sa répartition de couleurs, autrement dit de sa *texture*, est nécessaire pour extraire ce dernier. Comment différencier un torrent à sec composé de gros galets gris avec le sentier de montagne où se trouve de la pierraille de même couleur mais de plus petite taille?

Le chapitre 3 présente un inventaire des techniques du traitement d'image pour aboutir à l'extraction d'un objet. Il fait une synthèse de ce domaine en pleine innovation, comme le prouvent les dizaines de parutions scientifiques annuelles sur ce sujet.

3.CONTRAINTES

L'objectif qui est la principale contrainte, se résume à ceci. Le produit créé est utilisé dans un *contexte audiovisuel* . L'utilisation des logiciels GETRIS existants, les conversations avec les infographistes et les commerciaux de la société, ainsi que l'expérience de nos premières réalisations (phase prototype) permettent de dresser un inventaire des caractéristiques de ce type de logiciel.

Le graphiste est en général un artiste qui se méfie de l'informatique. L'outil que nous lui proposons, doit être aussi *convivial* que possible. Il doit être *plus ou du moins aussi facile d'emploi* que le pinceau pour un peintre! Le soin porté à l'*ergonomie* est un facteur très important. Le logiciel se doit d'être *souple* . L'opérateur a le droit d'abandonner l'opération courante pour en exécuter une autre de son choix.

Le produit est très *interactif* , pour ce fait les principales règles du RHM sont respectées :

- .l'arborescence dans les menus est limitée;
- .les fonctions communes sont regroupées;
- .minimisation des ordres de validation;
- .les opérations valides et invalides à un instant donné sont différenciées (avec une couleur par exemple).

L'apprentissage est facilité par l'emploi d'*icônes*, utilisant des paradigmes de la vie courante, pour représenter les différentes actions possibles. Le meilleur exemple est l'icône "corbeille" de MACINTOSH™ pour abandonner l'opération courante.

Plus grande est la variété des gens travaillant sur le logiciel, meilleure est sa commercialisation! Si pour les spécialistes de l'infographie le terme gradient, par exemple, signifie quelque chose de concret, il n'en est pas de même pour l'utilisateur lambda. Pas de mystère, les notions théoriques sont *transparentes* pour rendre le logiciel plus *attractif*.

Le travail artistique est une suite d'expérimentations. Il faut plusieurs retouches pour "ajuster" la bonne couleur d'un objet par exemple. Les opérations fréquentes, demandées par l'opérateur, doivent être aussi peu coûteuses que possible en temps. L'exécution, même si elle n'est pas immédiate, reste acceptable. Les algorithmes ainsi que les structures de données sont axés sur la *rapidité* , l'idéal étant une gestion en *temps réel*. Un traitement conséquent en temps d'exécution, est signalé mais non implémenté même si ses résultats sont intéressants.

Le logiciel permet des opérations de correction dans l'élaboration progressive de l'image et, dans la mesure du possible, de *recupérer les erreurs* du graphiste.

En imagerie orientée audiovisuelle, la *qualité de l'image* est importante. Un objet extrait, dont les contours débordent sur le reste de l'image, n'est pas satisfaisant. Par contre, la correction des imperfections presque invisibles à l'écran, au prix de traitements importants, ne sont pas nécessaires (*compromis qualité/ temps d'exécution*).

Un film est composé de 24 images par seconde (de plus en plus 25 images dans les standards d'aujourd'hui). Il faut limiter les opérations fastidieuses comme l'extraction de 24 fois le même objet. Les paramètres permettant l'extraction de ce dernier sont alors réutilisés *automatiquement*, même si la position de celui-ci évolue dans l'espace. Dans le cas contraire, l'outil devient vite fastidieux, et l'expérience le montre, il est souvent délaissé.

Il est souvent difficile pour un non-informaticien, de passer d'un logiciel à un autre. Pour pallier à ceci, les produits de GETRIS IMAGES forment une gamme. Ils sont conçus de façon similaire. Par exemple, les menus affichés à l'écran se ressemblent quelque soit le logiciel, ainsi que les opérations de base comme le décalage d'une image à l'écran. Le produit à concevoir respecte un *standard de logiciel*, c'est à dire une cohérence dans la présentation. Ses fonctions ne sont pas redondantes avec celles des produits déjà existants.

Le logiciel doit être *ouvert*, c'est à dire qu'il est aisé d'intégrer des nouveaux modules de traitement sans remettre en cause la structure du programme.

Le logiciel est "*avare*" dans le nombre de plans mémoires, pour qu'il puisse s'exécuter sur des configurations minimales de travail.

Pour être utilisé en animation, le logiciel n'exploite pas seulement la portion visible de l'écran (720x574 pixels) mais toute la possibilité du plan mémoire (1024x1024 pixels).

Les normes américaines (N.T.S.C.) sont différentes des normes européennes . Pour ce fait, les menus sont limités en hauteur à 480 lignes maximum pour être visibles quelque soit l'écran utilisé.

La réalisation décrite dans le chapitre 4, a pour objectifs de respecter ces contraintes draconiennes!

CHAPITRE 3

ETUDE THEORIQUE

EXPERIMENTATIONS

ET

CHOIX

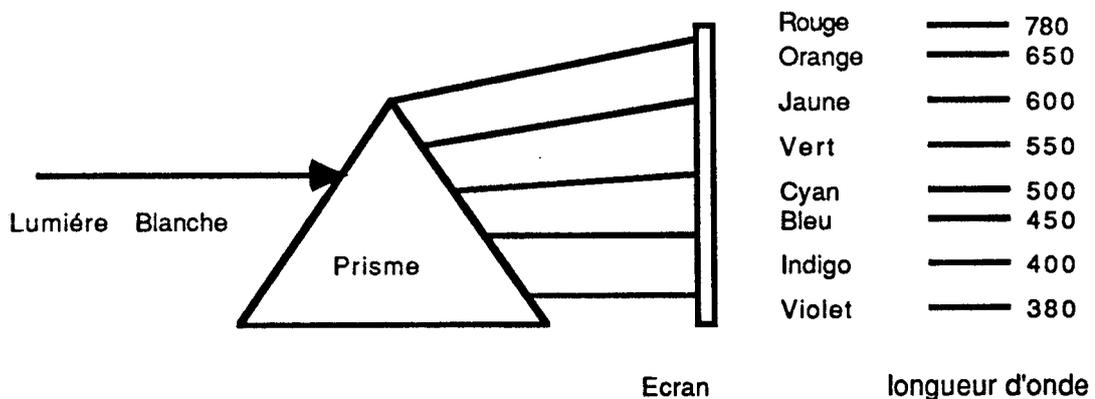
1. LA COULEUR EN SYNTHÈSE D'IMAGES

Avant toute étude, il est nécessaire de définir la notion de couleur, d'espace de couleurs qui sont la base de tous nos traitements.

1.1. Qu'est ce que la couleur?

En synthèse d'images, la notion la plus importante est la couleur. Elle est fondamentale pour la différenciation des objets, elle correspond à la perception naturelle d'une image (ART87). Elle joue un rôle fondamental pour rendre les images réalistes (effets de transparence, dégradés de couleur ...) (PER88). Il est difficile en peu de mots de spécifier une couleur car sa perception est purement psycho-physiologique, elle dépend de chaque individu.

Le spectre de la lumière solaire, décomposé par le prisme de NEWTON, donne les couleurs suivantes :



1.2. Espaces de couleurs

1.2.1. Modèles sur la théorie trichromatique

L'écran est composé de points images appelés **pixels** (abréviation de picture élément). Ce sont les plus petites unités visibles d'un écran vidéo. Chaque ligne de l'écran est divisée en un certain nombre de ces points, où est affectée une valeur numérique qui représente l'intensité du signal. La **définition** de l'écran c'est à dire sa **résolution** dépend du nombre de ces points.

L'étude de la perception des couleurs (ROC75) a montré que presque toutes les couleurs peuvent être obtenues par un mélange pondéré de trois couleurs de base, appelées couleurs primaires. Ceci est dû en particulier au phénomène d'intégration dans l'espace et le temps, et au pouvoir séparateur de l'oeil (MAR84). Ce procédé de superposition de couleurs s'appelle la **trichromie**.

La fonction image couleur au pixel $A(x,y)$ devient :

$$g(x,y) = g_1(x,y) + g_2(x,y) + g_3(x,y) \text{ où}$$

$g_1(x,y)$, $g_2(x,y)$ et $g_3(x,y)$ sont les composantes de la couleur dans un espace couleur à trois dimensions

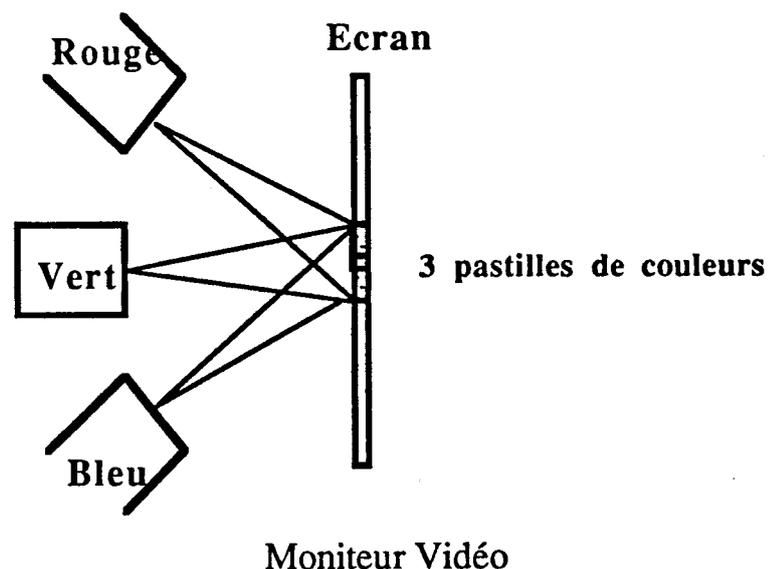
La définition des vecteurs de base est loin d'être unique. Les espaces de couleurs sont isomorphes et le passage de l'un à l'autre est simple. Une décomposition directement utilisable par le matériel est le modèle **RVB** (rouge vert bleu). Le système ATALIS utilise cette représentation où chaque composante a 256 valeurs possibles.

exemple : $R = 0$; $V = 0$; $B = 0 \implies$ Couleur noire

$R = 255$; $V = 255$; $B = 255 \implies$ Couleur blanche

Pour une image en 4096 couleurs (12 bits), les valeurs des composantes couleurs sont des multiples de 16 (4 bits par composante). Une image en 16 millions de couleurs nécessite 8 bits par composante.

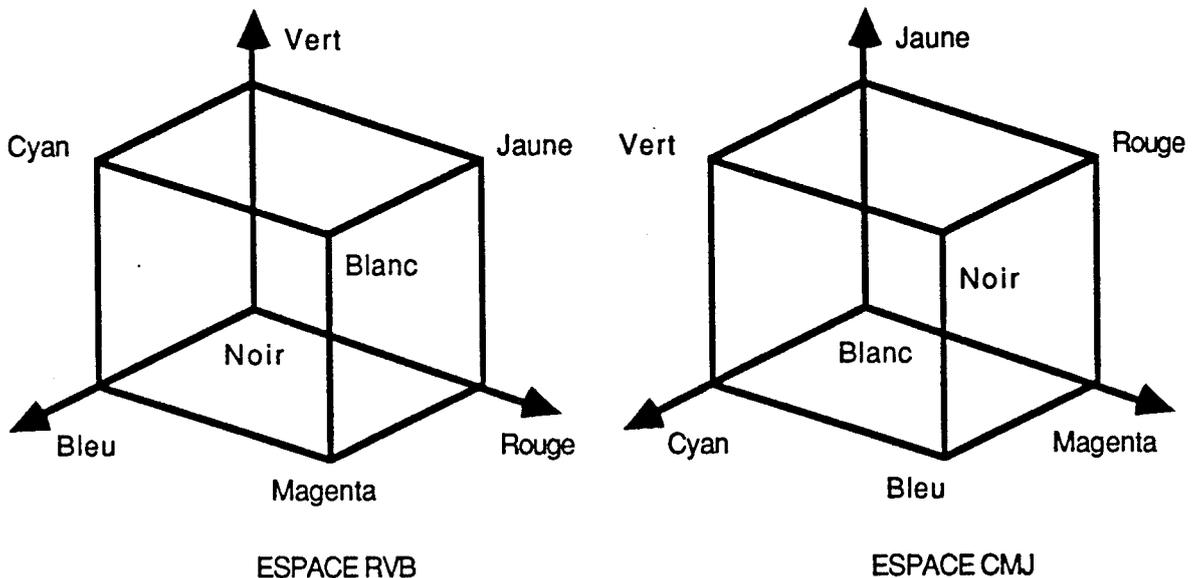
Un pixel de l'écran est en fait un ensemble de trois pastilles phosphorescentes (R,V,B) "bombardées" par trois canons à électrons. Chacun représente une couleur primaire, et son faisceau est proportionnel à l'intensité (poids) de cette couleur. Le contrôleur vidéo intègre directement la couleur. C'est le procédé le plus répandu à l'heure actuelle. Elle correspond à une **synthèse additive** des couleurs (MOL88).



La synthèse additive est un espace peu "naturel" (MAR84). Par exemple, le mélange des trois couleurs de base en RVB donne du blanc au lieu du noir attendu.

Le modèle **CMJ** (pour cyan, magenta et jaune) qui est l'espace inverse de RVB est plus approprié que celui-ci. La transformation d'espace est la suivante.

$$(r,v,b) = (1,1,1) - (c,m,j)$$



D'autres espaces sont issus du modèle RVB, comme par exemple l'espace YIQ, qui dérive du premier par simple produit scalaire. Il a fait l'objet d'une norme par le N.T.S.C. en 1953.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0,30 & 0,59 & 0,11 \\ 0,60 & -0,28 & -0,32 \\ 0,21 & -0,52 & 0,31 \end{bmatrix} \begin{bmatrix} R \\ V \\ B \end{bmatrix}$$

La composante Y, appelée luminance chromatique, correspond à la mesure de la "brillance". C'est le signal reçu par les postes de télévision noir et blanc.

Des modèles basés sur la **synthèse soustractive**, plus proche de notre perception de la couleur sont aussi utilisés, notamment en peinture ou pour les systèmes de reproduction couleur. Ils correspondent à une superposition de filtres colorés. La somme des trois primaires donne bien le noir attendu.

1.2.2. Modèles basés sur la perception

Les systèmes de représentation de couleurs, décrits ci-dessus, ne sont pas pratiques. Il n'y est pas aisé de définir simplement une couleur. Il est facile de concevoir qu'une couleur est la combinaison de trois primaires, mais difficile de prévoir la couleur finale après la variation d'une primaire! Ce type d'opérations est fréquent dans les dégradés de couleurs qui sont en fait des interpolations linéaires entre deux couleurs (CHA84). L'interpolation entre le rouge et le vert en RVB passe par un gris moyen (surprenant !).

D'autres systèmes ont été créés, qui correspondent à l'intuition du système oeil-cerveau de l'homme, c'est à dire à la **perception subjective** de la couleur. Les critères pour les définir sont des mots comme : vif, pastel, clair...

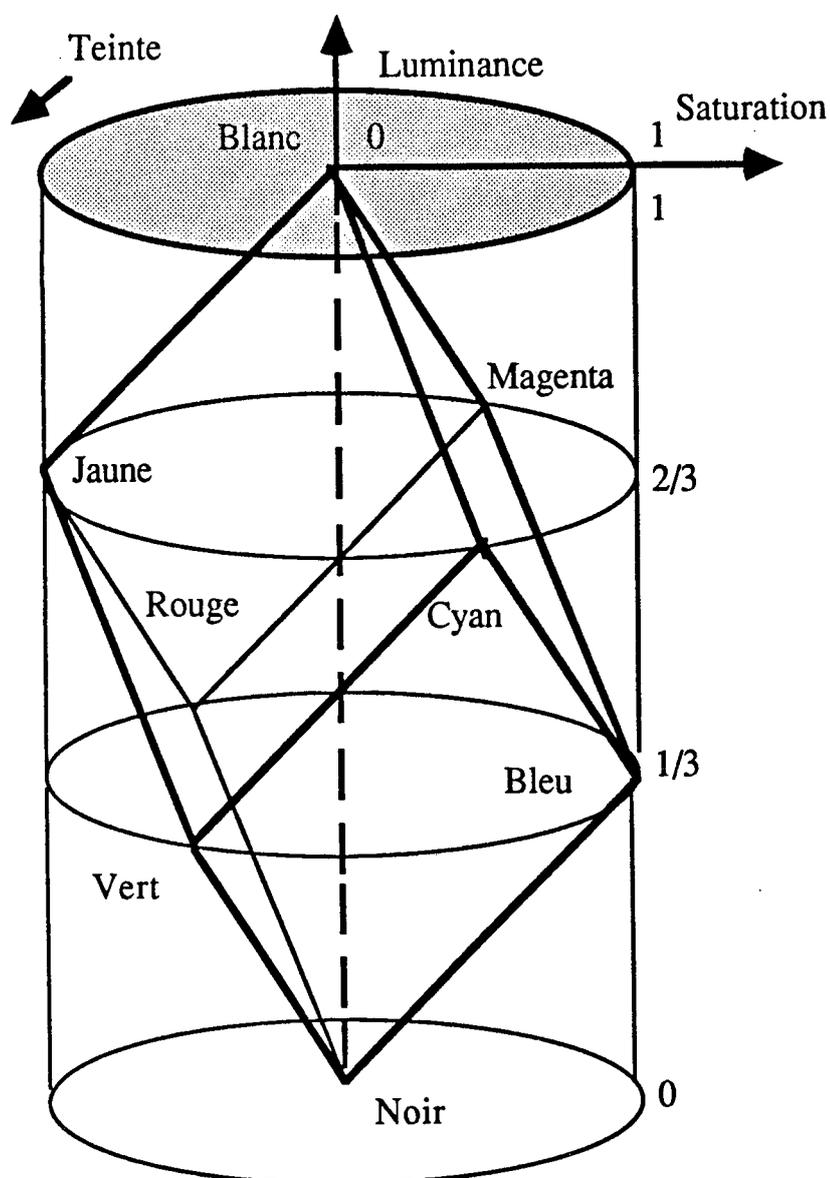
Les grandeurs fondamentales de la **colorimétrie** (PER88) (MAR84)(MOL88) sont les suivantes :

- .la teinte;
- .la luminance;
- .la saturation.

La teinte est la longueur d'onde dominante de la couleur. C'est la perception colorée ou autrement dit "la couleur de la couleur". Elle permet la distinction entre un rouge et un vert (rougeâtre, verdâtre).

La luminance correspond à l'énergie lumineuse reçue ou encore à l'intensité d'excitation visuelle (clair au sombre). Elle exprime la sensation d'éclairage.

La saturation définit la distance entre la couleur et le gris. Elle mesure la proportion de couleur pure par rapport au blanc (sensation de pureté). Elle distingue les couleurs vives des couleurs délavées (ternes). Elle donne par exemple la différence entre le rosé et le rouge.

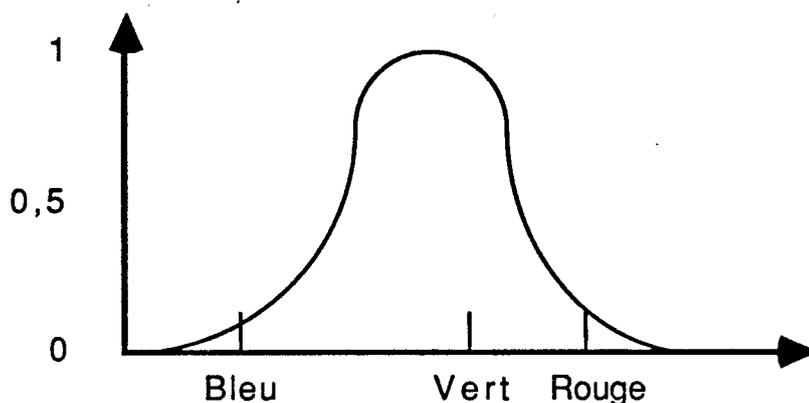


Cube RVB inclus dans le cylindre TLS

L'espace de couleurs issu de ces trois caractéristiques, est l'espace **TLS**. L'angle entre l'origine (couleur bleue) et la couleur courante correspond à la teinte. La distance de cette couleur à l'axe vertical donne la saturation, et la hauteur à ce même axe définit la luminance. L'espace TLS est un espace théorique (non directement applicable sur les machines). Nous sommes obligés d'effectuer une conversion de couleurs dans un système type RVB. Mais attention, nous voyons que certaines couleurs définies dans le cylindre TLS, n'ont pas de correspondance dans le modèle RVB. Prenons par exemple une couleur à même luminance et saturation que le vert mais avec la teinte du jaune.

L'interpolation des couleurs est beaucoup plus naturelle (prévisible) dans ce modèle. Par exemple, un phénomène d'ombrage est réalisé par interpolation entre la couleur et le noir (luminance), celui de brume ou de distance par une interpolation entre la couleur et le gris (saturation). Chaque composante a 256 valeurs possibles.

A éclairage égal, certaines couleurs apparaissent plus lumineuses que d'autres. L'oeil est plus sensible dans le jaune-vert.



Courbe sensibilité relative

Mais ce modèle de couleur, bien que fréquemment utilisé (exemple le logiciel MULTISCOPE développé par CAP SOGÉTI SYSTEMES exploitant les images satellites), est loin d'être unique. Nous pouvons citer en outre l'espace HSV (teinte, saturation et valeur), celui TNB (teinte, noir et blanc) (CHA84) et des espaces plus récents comme LAB et LUV où la notion de dégradés est meilleure (MOL88).

2. LA VISION ARTIFICIELLE

2.1. Définition du contexte de l'application

2.1.1. Contexte idéal en vision

Précisons en quelques phrases le contexte idéal en vision artificielle.

Il est plus aisé d'identifier un objet lorsque celui est connu. C'est une évidence! Les critères servant à l'analyse sont alors plus pertinents et plus facilement affinés par expérience.

- exemples :
- repérage de bâtiments (formes rectangulaires dans des zones rurales);
 - ajustement des paramètres pour détection de route dans le déplacement d'un robot (THO88);
 - agriculture (inventaire des cultures, suivi des zones inondables);
 - aménagement et surveillance du territoire.

Une base de données, composée d'objets de référence est parfois créée. L'image à identifier est comparée à ces modèles.

exemple : analyse de caractères.

L'identification s'effectue à partir de plusieurs images d'où un recouvrement d'informations :

- différents angles dans la prise de vue (images issues du satellite SPOT);
- différentes vues dans le temps (détection d'objets mobiles par exemple).

L'image à traiter est une image simplifiée. Dans la plupart des cas, c'est une image en noir et blanc où la différence d'intensité des pixels se traduit par un dégradé de gris. Elle comporte des objets en nombre limité (l'idéal est bien sûr deux : un objet et un fond !), bien contrastés les uns par rapport aux autres (HAR85).

- Les objets sont constitués de régions homogènes où l'intensité des pixels est plus ou moins constante. Le nombre de ces régions est limité et leurs tailles suffisamment conséquentes pour les analyser.
- Les bords des objets ne sont pas irréguliers. Les contours sont définis par des segments de droite.
- Les régions adjacentes des objets ont des valeurs différentes.
- Le nombre de trous internes (imperfections) est peu important.

exemple : Contrôle de qualité dans la fabrication de biscuits alimentaires (POS87). Le biscuit est de couleur blanche sur une bande transporteuse de couleur noire. Nous ne différencions que deux objets.

Le résultat demandé ne nécessite pas une grande précision. Dans le déplacement de mobile (THO88), la détection des bords de la route (opération difficile dépendant de la nature du terrain) n'est pas recherchée, ce dernier avançant sur le milieu de la chaussée.

Les conditions de prise de vue sont parfaites. L'éclairage est judicieusement étudié pour révéler les bonnes caractéristiques de l'objet. Des traitements sont effectués pour augmenter certains effets (par exemple le contraste ..). La qualité des résultats est fonction de celle de l'image initiale !

L'automatisation des systèmes de vision est facilitée lorsqu'elle ne s'adresse qu'à une seule catégorie d'objets.

exemple : contrôle de soudure

2.1.2. Notre contexte

Le contexte de notre application est loin du cas idéal (malheureusement pour la simplicité de la solution, heureusement pour l'intérêt du travail demandé !).

Nous ne sommes pas maître de l'image à analyser. Sa qualité est même parfois douteuse : image issue de revues... Le contraste laisse à désirer :

-exemple : robe verte sur un champ vert.

Une multitude de zones avec des imperfections constituent l'objet :

-exemple : feuillage en automne.

D'une image à une autre, les objets à extraire sont de nature différente:

-exemple : arbre, maison, ciel

La netteté de l'extraction est recherchée. Un visage humain à qui il manquerait une portion d'oreille n'est pas tolérable.

Nous manipulons des images en couleur. Ceci apparaît comme un plus par rapport à la vision classique. Le principal inconvénient est que la majorité des études se réfèrent à des images en noir et blanc (où même le nombre de niveaux de gris est limité), et qu'il faut par conséquent (si ceci est possible) adapter leurs algorithmes à la couleur.

L'outil proposé a pour cadre l'audiovisuel ce qui implique des calculs s'effectuant en temps réel, ou du moins s'en approchant. Des traitements qui manipulent un nombre important de paramètres (de l'ordre de 300 avec GAGALOWICZ (GAG85) pour l'analyse de texture) et qui nécessitent une réelle puissance de calcul, bien que donnant des résultats intéressants, sont irréalistes dans notre étude.

2.2. Qu'est ce qu'un système de vision artificielle?

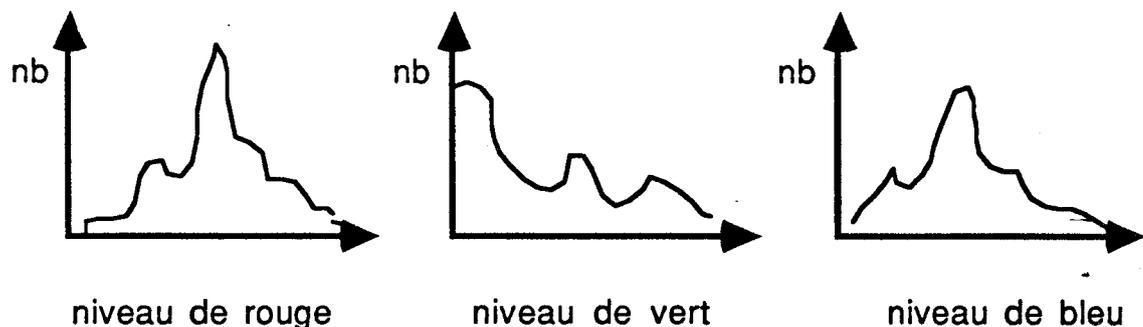
Tout système de vision se décompose invariablement en trois étapes :

- saisie d'image (et restauration) ;
- traitement d'image;
- analyse d'image.

2.2.1. Histogramme

Avant toute chose, il est nécessaire de définir le terme **HISTOGRAMME**, utilisé fréquemment en vision. Mathématiquement, c'est la description probabiliste du premier ordre de l'image, il correspond aux fréquences relatives d'apparition des différentes valeurs que prend une variable. Il se représente sous la forme d'un tableau unidimensionnel sur lequel il est possible de faire tous les traitements statistiques classiques (POS87) . Son allure donne des informations :

- .sur la nature globale (répartition);
- .sur l'état de la surface;
- .sur les traitements d'amélioration possible.



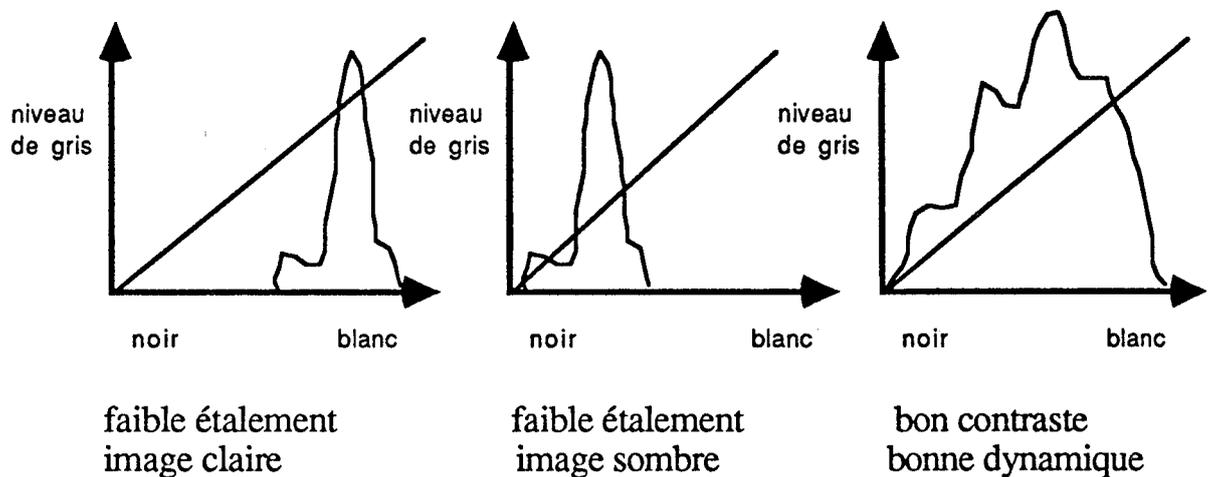
Exemple : 3 histogrammes d'une image couleur dans un système de couleur RVB. A chaque niveau de couleur (R, V ou B), l'histogramme donne le nombre de pixels possédant cette caractéristique.

2.2.2. Saisie d'image

Il s'agit de la transformation dans la mémoire de l'ordinateur, d'un objet analogique en tableaux numériques directement exploitables (CHA86).

Cette étape est de toute première importance. Un bon système de vision permet de représenter de façon satisfaisante, c'est à dire avec suffisamment de pixels, les détails significatifs. C'est la **définition**.

De même, la **dynamique** de l'image, qui se mesure par son histogramme, pour avoir un bon contraste, est étalée. A chaque pixel de l'image est affectée une intensité codée sur 8 bits (image noir et blanc), comprise théoriquement entre 0 et 255. En pratique, les valeurs sont loin de couvrir toute cette dynamique. Des mots comme image claire, sombre, bien contrastée, terne ... (CHA86), la caractérisent bien.



L'opérateur, pendant cette saisie, effectue des réglages prédominants qui agissent sur la dynamique :

- .éclairage;
- .saturation de l'image.

De mauvaises conditions créent une image imparfaite (**bruit**) qui s'avère difficilement exploitable par la suite.

2.2.3. Analyse de l'image

À partir de l'image, des renseignements quantitatifs sont obtenus tels que la longueur, le périmètre, la surface, le centre de gravité... Des affinements successifs (érosion, dilatation, ouverture, fermeture), des comparaisons avec des objets prédéfinis permettent d'identifier l'objet en question. Ces techniques sont regroupées dans la **Reconnaissance de forme**, à mi-chemin entre le traitement d'images et l'intelligence artificielle (arbre de classification ...).

Cette étape est en aval de notre étude. Nous cherchons à isoler l'objet et non à l'identifier.

nota :

érosion : affectation au pixel central de la valeur minimale du voisinage
dilatation : affectation au pixel central de la valeur maximale du voisinage
fermeture: dilatation suivie d'une érosion
ouverture: érosion suivie d'une dilatation

2.2.4. Traitement de l'image

Le traitement de l'image permet la transformation d'une image en une autre image, mettant en exergue certaines informations. Des techniques nombreuses et variées sont employées. Les prochains paragraphes de ce chapitre tentent de les énumérer.

Pour certaines d'entre elles, qui nous paraissent intéressantes, une expérimentation est faite suivie de ses conclusions. Car c'est seulement en pratiquant que nous nous rendons compte de l'efficacité ou de l'inefficacité de tel ou tel traitement sur des images d'origines diverses et de qualités différentes!.

3. PRETRAITEMENT

L'objet pour une raison ou pour une autre (cf paragraphe Saisie de l'image), est difficilement isolable. Un **prétraitement** s'impose avant toute extraction.

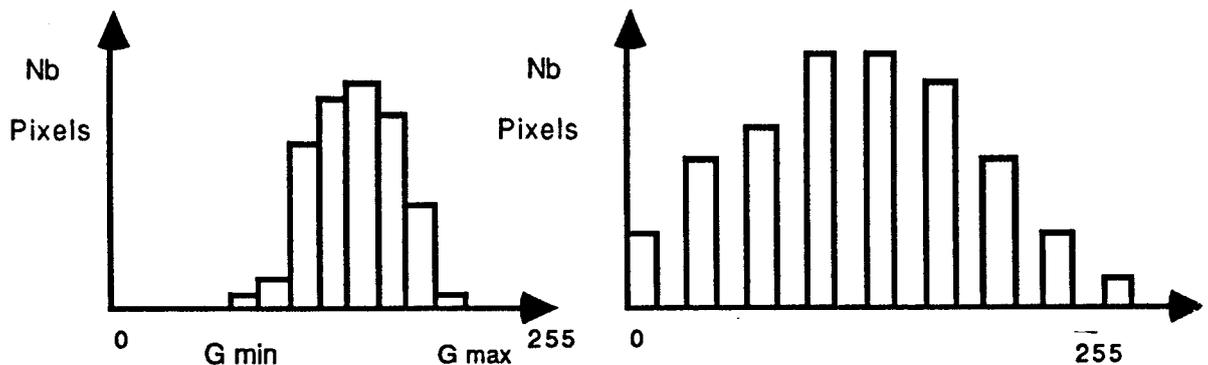
3.1. Traitement point par point

L'intensité initiale du pixel qui détermine sa couleur (niveau de gris ou de rouge-vert-bleu) est modifiée par une fonction mathématique (puissance, logarithme ...).

3.1.1. Egalisation

L'**égalisation** de l'histogramme consiste à trouver une fonction, qui modifie la répartition, pour obtenir une **répartition uniforme** sur l'ensemble de la plage des intensités possibles (CHE85)(CAN81).

La dynamique est ainsi améliorée d'où un meilleur contraste. Cette opération est rendue nécessaire lorsque le spectre de fréquences est concentré dans une bande étroite. Elle est aussi appelée **étalement d'histogramme**.



Histogramme niveaux de gris avant prétraitement concentré entre G min et G max Histogramme après égalisation

La fonction F de transformation de l'intensité du pixel G est alors la suivante :

.si $G < G_{\text{MIN}}$ alors $F(G) = 0$

.si $G > G_{\text{MAX}}$ alors $F(G) = 255$

.si $G_{\text{MIN}} \leq G \leq G_{\text{MAX}}$ alors

$$F(G) = \frac{(G - G_{\text{MIN}}) * 255}{G_{\text{MAX}} - G_{\text{MIN}}}$$

L'adaptation pour une image couleur est immédiate. Le niveau de couleur R,V ou B remplace le niveau de gris dans chaque fonction de transformation.

Cette technique a fait l'objet d'une expérimentation. Bien sûr, la première conclusion est que l'image résultante devient une image peu réaliste. La transformation appliquée aux trois composantes provoquent une distorsion de teintes. Avons-nous vu un arbre bleu ? Mais elle est particulièrement intéressante pour mettre en évidence des objets (en jouant sur les valeurs des différents seuils), aux couleurs similaires, les uns par rapport aux autres.

3.1.2. Couleurs normalisées

Les composantes RVB de l'image ne reflètent pas uniquement les propriétés chromatiques de l'objet mais aussi son intensité lumineuse, c'est à dire son éclaircissement. C'est ainsi qu'un objet de teinte uniforme, présente une grande variété de couleurs, s'il est soumis à plusieurs sources lumineuses ou si un autre objet lui fait de l'ombre(POS87).

Pour retrouver sa teinte d'origine, une fonction de transformation sur chaque composante est appliquée, rendant insensible les changements d'intensités lumineuses. Tout pixel de composantes R,V,B est transformé en composantes FR,FV,FB de la façon suivante :

$$FR = \frac{R}{R + V + B} ; FV = \frac{V}{R + V + B} ; FB = \frac{B}{R + V + B}$$

Le diviseur est un multiple de la luminance. La somme des trois composantes est égale à l'unité d'où le nom de couleurs normalisées.

Ce type de transformation, après différents essais, n'a pas donné les résultats espérés. L'image obtenue est une image de "tendance grisâtre", ne permettant pas d'isoler la teinte initiale de l'objet. Son principal inconvénient à notre avis, est de ne pas conserver la teinte et la saturation des objets.

Une autre fonction de transformation a été essayée. L'information Teinte et Saturation restent inchangées tandis que la Luminance prend une valeur constante (nous ne pouvons plus parler de grandeurs normalisées). Cette technique aboutit à une image plus satisfaisante. Des zones de même teinte mais d'ombrages différents, apparaissent avec des couleurs semblables. Mais l'image est rendue floue donc peu exploitable. Ce type de recherche n'a pas eu de suite.

3.2. Traitement spatial

A la différence du traitement point par point, les fonctions de transformation de la valeur du pixel ne concernent pas uniquement son intensité mais aussi celles des pixels voisins appelés voisinage (CHA86).

Ces opérations se nomment "filtrages". En règle général, le voisinage étudié correspond aux huit pixels voisins.

X-1, Y-1	X,Y-1	X+1, Y-1
X-1, Y	X,Y	X+1, Y
X-1, Y+1	X,Y+1	X+1, Y+1

Voisinage du pixel de coordonnées (x,y)

3.2.2. Filtre passe-bas

Ce filtre s'utilise lorsque l'image est bruitée. Le bruit est souvent situé dans les hautes fréquences du signal et un filtre passe-bas élimine ces dernières (CHE85).

Physiquement, il substitue la valeur du pixel courant par une "moyenne" des pixels voisins.

Les filtres les plus couramment employés sont les suivants :

$$F1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad F2 = \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Le premier filtre (F1) met à contribution de manière identique tous les pixels appartenant au voisinage, tandis que le second (F2) attache autant de valeur au pixel courant que l'ensemble des autres pixels voisins.

Comme il est noté précédemment, ce filtre réalise une moyenne et par conséquent supprime les petits détails. Un effet de **flou** apparaît sur l'image. De même les transitions entre différentes zones s'adoucissent. Le bruit est filtré mais la qualité de l'objet se dégrade! Ceci va à l'encontre de notre objectif (qualité).

Une solution consiste alors à adapter ce filtrage. La transformation du pixel intervient seulement si ce dernier n'est pas trop différent de ses voisins, c'est à dire s'il appartient à la même zone. Par exemple :

.soit $g(x,y)$ la fonction image au pixel A de coordonnées x,y

.soit moy la moyenne du voisinage

.la fonction de transformation $f(x,y)$ du pixel est définie de la façon suivante:

$$f(x,y) = \text{moy si } |g(x,y) - \text{moy}| < T$$

$$f(x,y) = g(x,y) \text{ autrement}$$

T étant un seuil préétabli.

3.2.2. Autres filtres

D'autres filtres sont utilisés en traitement d'images. Le **filtre passe-haut** exacerbe les détails en renforçant le contraste. Il visualise mieux les limites des différentes zones de l'image.

$$F3 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad F4 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad F5 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Le **filtre LAPLACIEN** détecte les bords des objets comme des zones à forte variation de contraste.

$$F6 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

La notion de différentiation, utilisée par ces filtres, est reprise dans le paragraphe sur le seuillage par variation de gradient.

L'inconvénient majeur avec les filtres aux coefficients négatifs, et que leurs utilisations ne conviennent que pour des images en noir et blanc. Nous avons à faire à des soustractions d'images en couleur, qui ne permettent en aucun cas de détecter ni les bords des objets, ni d'augmenter le contraste (cf paragraphe sur le seuillage par différence d'images).

3.3. Autres prétraitements

D'autres prétraitements sont employés. Nous les citons à titre indicatif bien que ne rentrant pas dans le cadre de notre étude:

.passage en fausses couleurs ou pseudo coloration ou table de contraste (CAN81): A chaque niveau d'intensité, une couleur est fixée arbitrairement. Elle permet à l'utilisateur de distinguer en couleurs, des nuances locales difficiles à percevoir en noir et blanc. En effet, l'œil peut différencier des milliers de teintes alors qu'il ne sépare qu'une trentaine de niveaux de gris.

.combinaison d'images : Une image en mémoire correspond à un tableau numérique. Différentes opérations mathématiques sont faites sur ces nombres, qui mettent en valeur les propriétés physiques des éléments constituant l'objet (CAN81)

.superposition d'images à l'écran.

.sous-exposition : Une valeur constante est ajoutée à chaque pixel qui augmente l'intensité d'éclairage et qui compense une sous-exposition. L'opération simule une augmentation du temps de pose.

.sur-exposition : Une valeur constante est retranchée à chaque pixel. Ceci compense une sur-exposition.

.traitement logarithmique : Cette transformation donne une mesure de l'intensité lumineuse.

4. SEGMENTATION DE L'IMAGE

La **segmentation**, étape qui intervient après le prétraitement, sépare l'image en un certain nombre de régions plus ou moins homogènes. Nous **divisons** l'image en **classes** à partir des caractéristiques de celles-ci, appelées **attributs**. Ce sont des décisions qui s'appuient sur les propriétés des pixels, de leur voisinage et de leur arrangement spatial. Il n'existe pas de méthode pour déterminer les attributs les plus pertinents. Ils dépendent du type d'image traitée et souvent ce sont les résultats obtenus qui justifient leur choix (POS87).

Deux techniques sont employées en fonction de la façon dont nous considérons les régions [POS87] :

- une région est un ensemble de points qui présentent des propriétés statistiquement semblables;
- une région est une zone limitée par un contour. Ce dernier est caractérisé par des variations importantes des propriétés des pixels.

La première technique permet de reconstituer des régions. Elle nous a conduit à élaborer les algorithmes de seuillage.

La seconde caractérise des zones homogènes par leur contours [BOU85] (cf paragraphe sur les suivis de contours).

Ces différentes techniques ont été étudiées puis expérimentées selon leur intérêt.

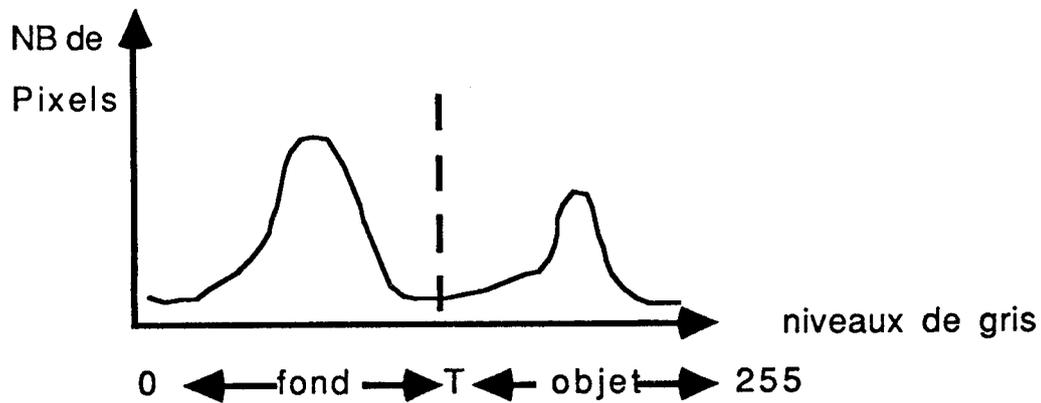
4.1. Seuillage

Le principe du seuillage est simple. Dans une image bimodale, c'est à dire constituée de deux classes d'objets bien définies, l'histogramme fait apparaître deux maxima séparés par un minimum. Ce dernier noté T , correspond au seuil idéal de la segmentation. La fonction de segmentation f est définie de la façon suivante:

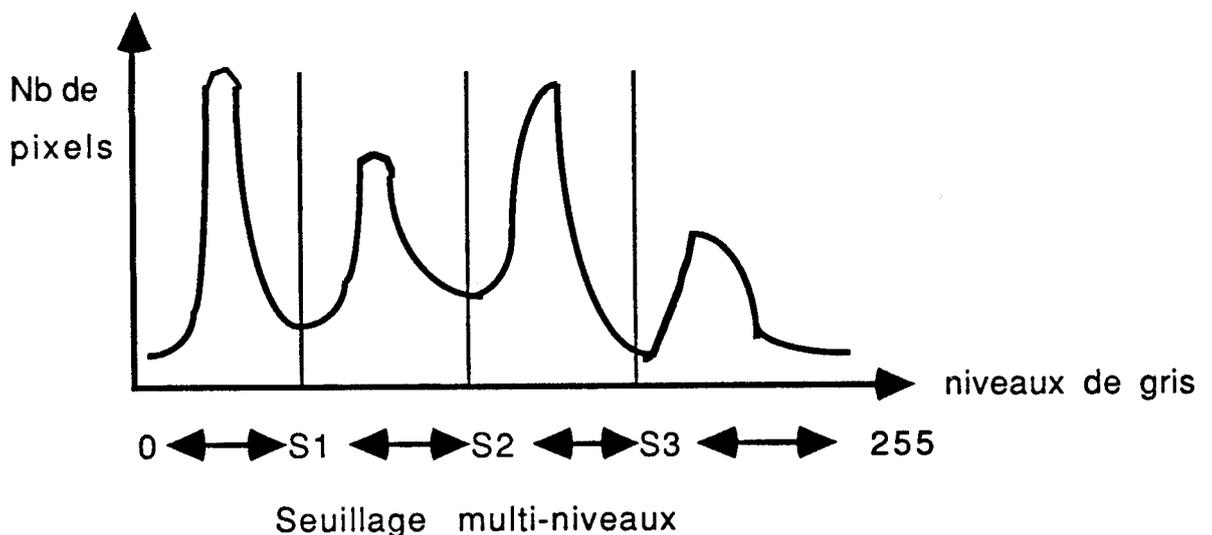
si $g(x,y) > T$ alors $f(x,y) = 255$; (x,y) appartient à l'objet

sinon $f(x,y) = 0$; (x,y) n'appartient pas à l'objet

Ce seuillage s'appelle **binarisation** de l'image. La détermination du seuil T est importante, car un mauvais seuillage entraîne forcément une mauvaise segmentation.



Pour une image comportant plusieurs classes, le principe reste le même. Chaque minimum, compris entre deux maxima définit la frontière entre deux classes.



Il est parfois plus facile de déterminer les maximums que les minimums. Le seuil pris est alors à équidistance de ces maximums. Il est possible de limiter le nombre de ces seuils en réalisant un prétraitement du style filtre passe-bas.

L'adaptation au "monde de la couleur" est trivial. Un objet est alors un ensemble de pixels compris entre deux seuils sur chaque composante. —

Ce type de seuillage, qui convient très bien pour des contrôles de production, n'est pas applicable dans notre étude. L'objet peut être constitué d'un ensemble important de zones homogènes qui entraîne une multiplication de seuils.

4.2. Apprentissage

La technique que nous utilisons est basée sur une phase d'apprentissage. Des points images représentatifs de l'objet, sont étudiés pour déterminer ses attributs. Bien sûr, plus grand est le nombre de pixels de référence, meilleur est la connaissance des caractéristiques de l'objet. Nous réalisons un échantillonnage de l'objet (POS87).

Les objets, en règle générale, ne sont pas composés de zones homogènes. Il faut avoir un pixel de référence par zone ce qui augmente le nombre de pixels prototypes. Pour le limiter, la solution consiste à admettre une **tolérance (écart admissible)** envers les pixels saisis.

Souvent c'est le résultat final qui indique si la sélection des prototypes ainsi que des écarts ont été pertinents (bonne segmentation). Leurs choix sont souvent effectués par expérience en fonction du type d'images traitées.

4.3. Seuillage relatif à la couleur

4.3.1. Seuillage RVB

Le premier seuillage qui vient à l'esprit, est le seuillage issu du modèle de couleurs RVB.

4.3.1. Seuillage RVB par moyenne

Nous calculons la moyenne des pixels prototypes. Un écart admissible est fixé pour chacune des composantes par rapport à celle-ci.

.soient moy_r, moy_v, moy_b la moyenne RVB issue de l'échantillonnage

.soient er, ev et eb l'ecart par rapport aux moyennes respectives

.soient r, v, b les composantes du pixel courant de coordonnées (x, y)

.F la fonction de segmentation est définie de la façon suivante :

$$\begin{aligned}
 - F(x, y) &= (r, v, b) \text{ si } |r - \text{moy}_r| < e_r \\
 &\text{et si } |v - \text{moy}_v| < e_v \\
 &\text{et si } |b - \text{moy}_b| < e_b
 \end{aligned}$$

$$-F(x, y) = 0 \text{ autrement}$$

Cet algorithme, facile à implémenter, "marche" seulement pour quelques cas. Les teintes qui composent l'objet, doivent être très proches les unes des autres, pour que la notion de moyenne soit encore significative. La moyenne entre un rouge et un vert ne représente pas grande chose!

4.3.2 Seuillage RVB par rapport à un ensemble de composantes

Les composantes RVB des prototypes saisis, constituent un ensemble de composantes valides (CV). Une tolérance (respectivement e_r , e_v et e_b), est définie sur cet ensemble. La fonction de segmentation est alors la suivante :

$.F(x,y) = (R,V,B)$ s'il existe 3 composantes (r,v,b) appartenant à CV, tel que

$$|R - r| < e_r \text{ et } |V - v| < e_v \text{ et } |B - b| < e_b$$

$.F(x,y) = 0$ autrement

La réponse au traitement est immédiate. Il suffit de lire l'information "pixel" directement en mémoire puisque celle-ci est déjà sur le modèle de couleurs RVB.

Ce type de seuillage donne de très bons résultats lorsque le traitement s'effectue sur une image 12 bits. Ils sont encore meilleurs, sans augmenter considérablement le temps de traitement, sur une image 24 bits (l'accès au pixel s'effectuant en deux fois), car il n'existe aucune perte de précision sur les couleurs. Il est particulièrement efficace, pour l'extraction d'objets faiblement contrastés, où la différence de couleurs est minime.

Ce traitement est rapide, car il ne nécessite que la comparaison des composantes du pixel traité avec les composantes valides en mémoire (3 fois 256 composantes). Mais le résultat est parfois erroné. Expliquons-nous par l'exemple suivant :

.soient 3 pixels prototypes respectivement (r_1, v_1, b_1) , (r_2, v_2, b_2) et (r_3, v_3, b_3)

.le pixel (r_1, v_2, b_3) est décrété valide bien qu'incorrect.

4.3.3. Seuillage RVB par rapport à un ensemble de prototypes

Ce seuillage ressemble au seuillage précédent mais il travaille sur l'ensemble des prototypes valides PV. La fonction de seuillage est alors la suivante:

$.F(x,y) = (R,V,B)$ s'il existe un prototype (r,v,b) appartenant à PV, tel que

$$|R - r| < e_r \text{ et } |V - v| < e_v \text{ et } |B - b| < e_b$$

$.F(x,y) = 0$ autrement

Pour un seuillage sur une image 12 bits, il est possible de connaître rapidement si le pixel traité correspond à un pixel valide, car le nombre de prototypes différents est limité à 4096. La réponse n'est pas fonction du nombre de prototypes saisis.

Par contre sur une image 24 bits, il est nécessaire de comparer les composantes du pixel traité, avec tous les prototypes saisis (dans le cas le plus défavorable, c'est à dire le pixel n'appartenant pas à la zone). La réponse est proportionnelle au nombre de prototypes saisis, ce qui est vite un facteur très pénalisant. Ce type de traitement est à utiliser en dernier ressort, sur des zones de petites dimensions, avec peu de prototypes et demandant une précision 24 bits.

Un autre inconvénient est la définition des tolérances sur les prototypes sélectionnés. Pour éviter des écarts importants en modèle RVB, qui ne sont plus significatifs pour le graphiste, il faut saisir un grand nombre de pixels de référence (un pixel par région).

4.3.4. Seuillage TLS

Les tolérances sont plus significatives dans le modèle de couleurs TLS et l'opérateur les ajuste plus facilement (cf paragraphe modèles basés sur la perception).

La fonction de segmentation est alors la suivante :

$$.F(x,y) = (T,L,S) \text{ s'il existe un élément } e(t,l,s) \text{ appartenant à PV,} \\ \text{tel que } |T - t| < \epsilon \text{ et } |L - l| < \epsilon \text{ et } |S - s| < \epsilon$$

$$.F(x,y) = 0 \text{ autrement}$$

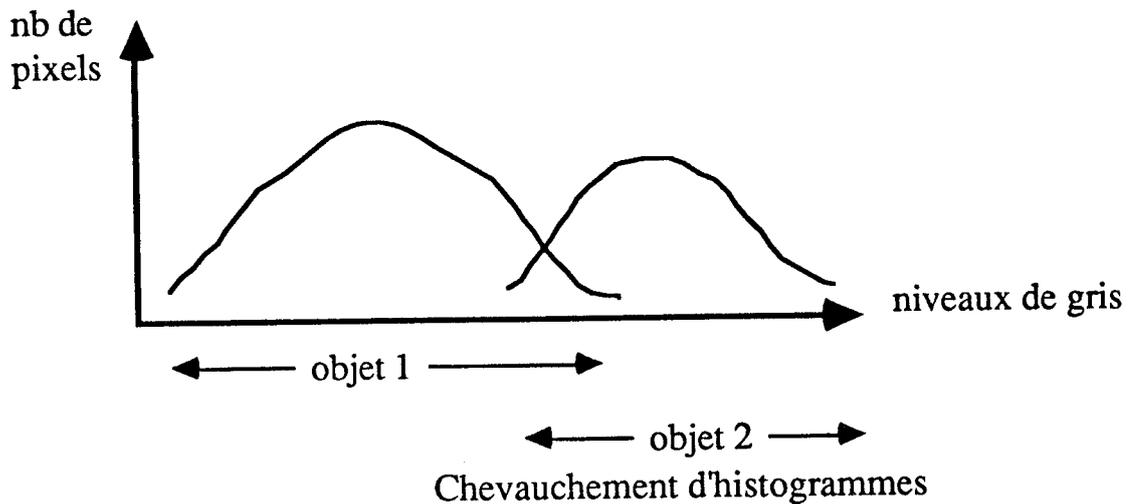
La conversion du modèle RVB (image archivée en mémoire) à celui TLS, n'est pas immédiate. Dans la partie réalisation de l'étude, une "astuce" permet de réduire le temps de conversion (cf paragraphe conversion RVB \Leftrightarrow TLS). Cependant elle nécessite l'analyse de l'image en 12 bits entraînant une perte de précision sensible. Par contre, le traitement est rapide car le nombre de prototypes différents est limité à 4096 comme avec le traitement RVB 12 bits.

Ce type de seuillage est à employer lorsque le contraste est satisfaisant, c'est à dire que l'information "poids forts" de l'image est suffisante, pour distinguer les objets.

4.4 Seuillage par l'analyse de l'aspect de surface

4.4.1. Définition de la texture

Quelque soit le type de seuillage relatif à la couleur adopté, il est impossible de distinguer, dans bien des cas, avec le seuil attribut de couleurs (ou niveaux de gris), un objet sans erreur. L'histogramme de deux objets sur une image peut très bien se "chevaucher". Cette superposition rend impossible alors la détermination d'un seuil correct.



Nous utilisons l'aspect de surface, dit texture pour différencier les objets. Il correspond à la dispersion des niveaux de couleurs dans le voisinage immédiat des pixels (arrangement) (POS87). Il est une propriété fondamentale dans l'analyse des régions pour la segmentation des images. C'est la seconde propriété, après la couleur, pour la discrimination des régions (BOU85).

Nous distinguons deux classes de texture :

- .texture douce caractérisée par des variations significatives des niveaux de couleurs sur un nombre important de pixels (basses fréquences).
- .texture grossière ou granuleuse où les variations sont brutales et rapprochées (gros damier).

4.4.2. Seuillage par variance

4.4.2.1. Rappels statistiques

La couleur d'un pixel est la combinaison des trois couleurs de base: Le Rouge, le Vert et le Bleu. Pour tout point de l'image, nous définissons les événements suivants:

.X_r = événement qui attribue un poids Rouge de 0 à 255 au pixel.

.X_v = événement qui attribue un poids Vert de 0 à 255 au pixel.

.X_b = événement qui attribue un poids Bleu de 0 à 255 au pixel.

Les variables aléatoires associées à ces événements sont identiquement distribuées et indépendantes (POS87).

La loi de ce triplet de variables aléatoires est donc une loi normale qui a pour densité (DUD73) :

$$p(\mathbf{X}) = \frac{\exp [-1/2 (\mathbf{X} - \mathbf{E}(\mathbf{X}))^T \mathbf{S}^{-1} (\mathbf{X} - \mathbf{E}(\mathbf{X}))]}{(2\pi)^{3/2} |\mathbf{S}|^{1/2}}$$

où $\mathbf{E}(\mathbf{X})$ est le vecteur moyenne de la distribution à trois composantes et \mathbf{S} est la matrice de covariance 3x3 de déterminant $|\mathbf{S}|$.

Plus précisément, si cette loi est calculée sur une région "prototype" de p pixels voisins (REI86) :

$$E(X_r) = 1/P \sum_{i=1}^p X_{r_i} \quad E(X_v) = 1/P \sum_{i=1}^p X_{v_i} \quad E(X_b) = 1/P \sum_{i=1}^p X_{b_i}$$

La matrice de covariance devient :

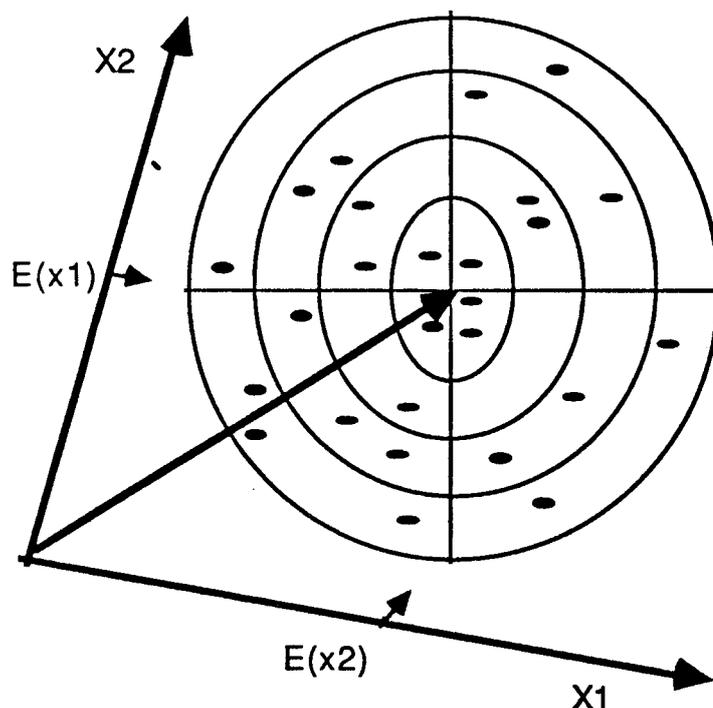
$$S = \begin{bmatrix} (X_r - E(X_r)) & (X_r - E(X_r)) & (X_r - E(X_r)) & (X_v - E(X_v)) & (X_r - E(X_r)) & (X_b - E(X_b)) \\ (X_v - E(X_v)) & (X_r - E(X_r)) & (X_v - E(X_v)) & (X_v - E(X_v)) & (X_v - E(X_v)) & (X_b - E(X_b)) \\ (X_b - E(X_b)) & (X_r - E(X_r)) & (X_b - E(X_b)) & (X_v - E(X_v)) & (X_b - E(X_b)) & (X_b - E(X_b)) \end{bmatrix}$$

La forme du groupement est déterminée en fonction du vecteur moyenne (regroupé autour de celui-ci) et de la matrice de covariance.

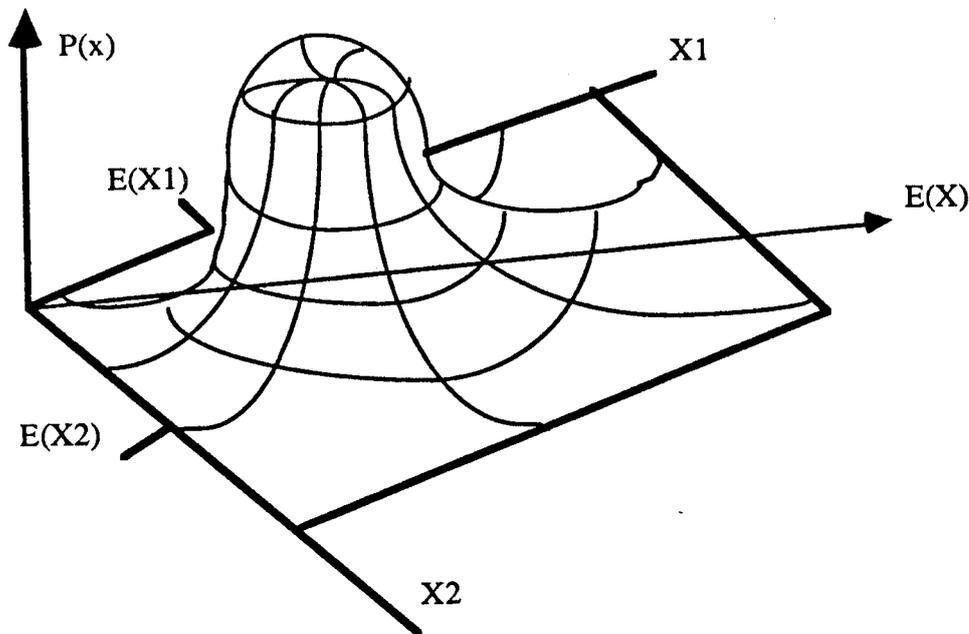
Les lignes d'équidensité sont des hyperellipsoïdes définies par :

$$(X - E(X))^T S^{-1} (X - E(X)) = \text{distance}$$

Elles correspondent à la notion de "distance" du pixel de couleur $X(r,v,b)$ par rapport à la région prototype où a été calculée la matrice de covariance.



Forme de groupements d'observations bidimensionnelles autour du vecteur moyenne en fonction de la matrice de covariance.



Représentation spatiale des lignes d'équidensité d'une fonction de densité de probabilité normale bidimensionnelle.

4.4.2.2. Mise en oeuvre du seuillage par variance

La matrice de covariance est une matrice symétrique. Cette remarque permet de réduire considérablement les temps de calcul dans l'élaboration de la distance d'un pixel à une région.

$$\text{Si la matrice } S^{-1} = \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix} \text{ et } X = \begin{bmatrix} R \\ V \\ B \end{bmatrix}$$

la distance devient $\text{DIST} = 2 [bRV + cRB + eVB] + aR^2 + dV^2 + fB^2$

L'estimation des paramètres statistiques (moyenne et variance) dépend du nombre de prototypes de la région de référence. Le taux d'erreur est inversement proportionnel au nombre de ces points.

Un voisinage trop petit pour l'élaboration de la matrice de covariance, donne une idée fautive sur la distance d'une couleur par rapport à la zone de référence. Le test réalisé sur une région fixée à 25 pixels, donne de bons résultats sans pour autant pénaliser le temps de réponse (la matrice de covariance n'est calculée qu'une seule fois).

Une tolérance d_t est fixée par rapport à la distance. La fonction de segmentation est définie de la façon suivante:

.soit d la distance-couleur du pixel courant à la région

$F(x,y) = (R,V,B)$ si $d < d_t$

$F(x,y) = 0$ autrement

Il suffit de saisir judicieusement un seul point de référence pour réaliser le traitement. Ceci constitue le principal avantage du seuillage par variance. Il donne de bons résultats sur des objets comme le feuillage des arbres (teintes similaires).

L'opération de seuillage est beaucoup plus longue que celle de seuillage RVB, car elle nécessite pour chaque pixel traité 13 multiplications ainsi que 5 additions , en format flottant.

Le seuillage par variance n'est pas applicable dans tous les cas. La matrice de covariance, sur une zone de référence homogène, donne la matrice nulle, qui n'est pas inversible. La distance est donc non définie pour ce prototype.

Sur des objets variant progressivement de teintes, il est possible de réaliser un seuillage par variance adaptative. Une nouvelle distance est définie, à chaque fois qu'une zone testée comporte suffisamment de pixels appartenant à l'objet. Cet algorithme est pénalisant en temps de traitement, car il implique fréquemment le calcul de la matrice de covariance et de son inverse, "gros dévoreur de cycles machines".

Pour des objets possédant plusieurs classes caractéristiques, il est souhaitable de définir plusieurs distances à partir de plusieurs zones de référence.

4.4.3. Seuillage par différence d'images

4.4.3.1. Théorie

Le principe utilisé pour analyser la texture d'un objet, consiste à réaliser une différence entre l'image initiale et l'image translatée de $\partial (\Delta x, \Delta y)$ (POS87).

En étudiant l'histogramme $H(g)$ de cette différence d'images sur N niveaux de gris, nous nous apercevons que lorsque nous sommes sur une région de texture grossière et si ∂ est petit par rapport au damier, l'ensemble des pixels est concentré dans le noir. Il y a peu de variations entre les pixels, et leurs différences sont nulles. Pour les zones très texturées, ∂ est relativement grand et les différences entre pixels sont alors importantes.

L'histogramme est caractérisé par sa moyenne X_1 , son moment d'ordre 2 X_2 ou son entropie X_3 . Après expérimentation, il s'avère que le moment d'ordre 2 permet un meilleur ajustage du seuillage (écarts entre les prototypes plus importants).

$$X_1 = 1/N \sum_{i=1}^N i H(g_i) \quad X_2 = 1/N \sum_{i=1}^N i^2 H(g_i) \quad X_3 = -1/N \sum_{i=1}^N H(g_i) \log g_i$$

4.3.2. Mise en œuvre

Après cette étude théorique, nous avons appliqué la différence d'images sur une image couleur. Cette technique permet de réaliser certes de jolis résultats, mais les histogrammes sont inexploitable. En couleur, la différence n'est pas significative. Nous nous sommes rapprochés alors de la théorie, en utilisant la luminance comme information pixel ce qui nous fait perdre l'information couleur.

Nous admettons une tolérance t_d par rapport à notre caractéristique de l'histogramme. Pour chaque pixel traité, nous calculons son moment d'ordre deux x_2 de différences d'images. La fonction de segmentation est alors la suivante:

$$.F(x,y) = (R,V,B) \text{ si } |X_2 - x_2| < t_d$$

$$.F(x,y) = 0 \text{ autrement}$$

Ce traitement est très "dévoreur" de temps de calcul. Il est fonction de la taille du voisinage. Un voisinage trop petit ne donne pas une bonne caractéristique de la texture de l'objet, par contre un grand voisinage est vite pénalisant.

Pour un voisinage 9×9 , le traitement demande déjà :

- .la lecture de 81 à 162 pixels (fonction de ∂) et le calcul de leur luminance respective (3 additions et une division par pixel)

- .81 opérations de soustraction pour le calcul de la différence

- .de 1 à 255 opérations de sommation et de multiplication pour le calcul de x_2 .

Pour ce type de calcul, il n'est pas nécessaire d'avoir une grande précision. Un traitement avec une image 12 bits est suffisant.

Il est à noter que la qualité du résultat dépend beaucoup de ∂ .

Avec des prototypes pris uniquement sur le centre de l'objet, nous n'obtenons que la partie centrale, tandis qu'avec des prototypes saisis sur les contours, nous avons une bordure de pixels appartenant aux deux objets.

4.4.4. Autres techniques d'analyse de texture

Une des principales voies de recherche dans le traitement d'images est l'analyse de texture. Sans détailler toutes ces techniques, nous pouvons citer :

- .la variation totale de niveaux de gris (POS87) au voisinage de chaque point suivant les directions horizontales et verticales.

- .la matrice des fréquences relatives, où chaque élément de celle-ci indique le nombre de fois, où un couple de points séparés par la distance d , dans la direction ∂ , présente le même niveau de gris (POS87).

- .la texture caractérisée par les pixels ayant un laplacien important.

- .la texture déterminée par les pixels, qui ont un nombre de voisins corrects, par rapport à une caractéristique, supérieur à un seuil.

- .la texture qui prend en compte les pixels, dont le maximum de la somme des gradients de son voisinage est supérieur à un seuil (HAR85).

Il est possible de combiner le seuillage de type couleur avec l'information spatiale (seuillage de type aspect). Un bon exemple est le seuillage TLS-différence d'images. Il améliore la précision du simple seuillage TLS en obtenant quelques pixels supplémentaires du contour de l'objet. Nous pouvons augmenter la tolérance sur TLS sans toutefois extraire des pixels qui n'appartiennent pas à l'objet.

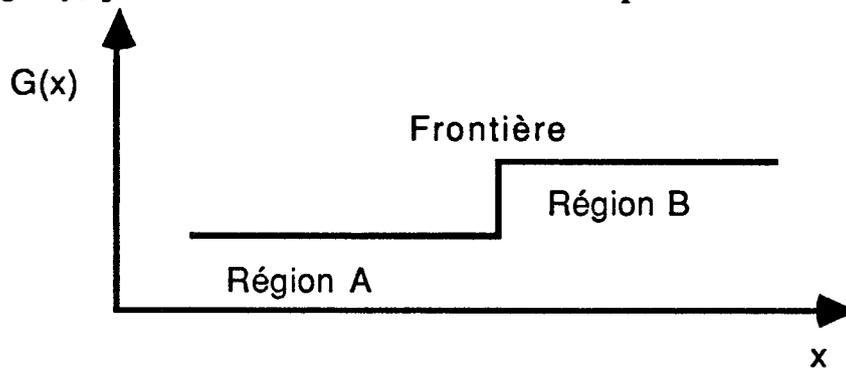
4.5. Analyse du contour

Nous ne nous intéressons non plus à la région mais aux transitions inter-régions, appelées contours .

4.5.1 Seuillage par variation de gradient

4.5.1. Définition du gradient de la fonction image

La frontière entre deux régions, est composée de pixels où la fonction image $g(x,y)$ présente une variation de forte amplitude.



Frontière théorique entre deux régions

Le gradient de la fonction image permet de détecter les contours. Il représente le degré de variation de la fonction image au point considéré (CHE85). Le module φ détermine l'amplitude de la variation de la fonction image et son argument θ la direction de celle-ci.

$$\theta(x,y) = \text{Arctg} \left(\left(\frac{\partial g(x,y)}{\partial y} \right) / \left(\frac{\partial g(x,y)}{\partial x} \right) \right)$$

$$\varphi(x,y) = \sqrt{ \left(\frac{\partial g(x,y)}{\partial x} \right)^2 + \left(\frac{\partial g(x,y)}{\partial y} \right)^2 }$$

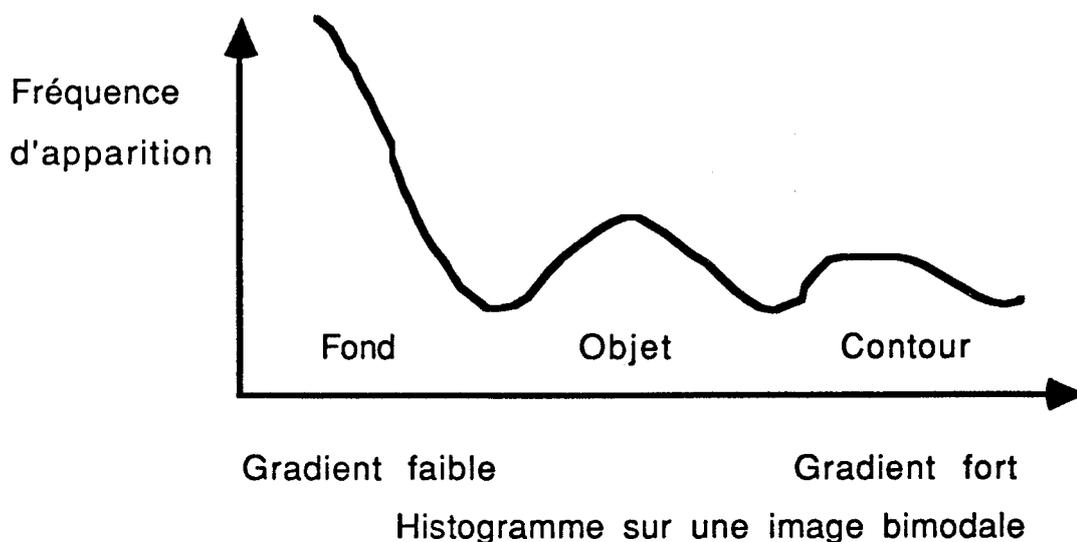
Le gradient permet de dire si un pixel est similaire à son voisinage, donc appartenant à la même région, ou différent s'il appartient au contour [BOU85]. Si la fonction image change brusquement de valeur, les pixels ont alors un gradient fort. Ils appartiennent au contour.

Au point $A(x,y)$, l'estimation des dérivées bidirectionnelles est la suivante :

$$\frac{\partial g(x,y)}{\partial x} = g(x+1,y) - g(x-1,y) = \Delta_o [g(x,y)]$$

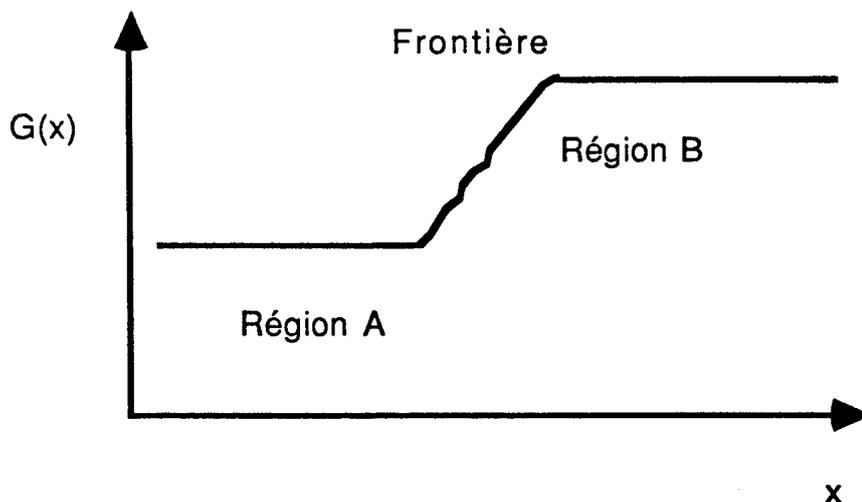
$$\frac{\partial g(x,y)}{\partial y} = g(x,y+1) - g(x,y-1) = \Delta_{\pi/2} [g(x,y)]$$

$\Delta_o [g]$ représente les variations de g dans les directions horizontales et $\Delta_{\pi/2} [g]$ dans les directions verticales.



4.5.2. Pondération du gradient

Avec ces opérateurs différentiels, la détection des contours est peu réaliste. Premièrement, les contours ne sont pas aussi délimités que sur la figure théorique, mais se rapprochent plutôt de celle ci-dessous.



Frontière réelle entre deux régions

Deuxièmement, une image comporte toujours des phénomènes parasites ou des irrégularités de surface [PRE82]. Pour les atténuer, un opérateur de lissage effectuant une "moyenne" avec le voisinage du pixel traité, est combiné avec un opérateur gradient de la façon suivante :

$$\text{Soit la matrice } G(k,l) = \begin{bmatrix} g(k-1,l-1) & g(k,l-1) & g(k+1,l-1) \\ g(k-1,l) & g(k,l) & g(k+1,l) \\ g(k-1,l+1) & g(k,l+1) & g(k+1,l+1) \end{bmatrix}$$

image du voisinage du point $A(k,l)$, nous définissons le produit interne avec une matrice de pondération W de même dimension par la formule suivante

$$G(k,l) * W = \sum_{(i,j) \in D} g(k+i,l+j) \cdot w(i,j) \quad \text{avec } D = \{[-1,1], [-1,1]\}$$

Les opérateurs différentiels sont substitués par

$$\Delta_0[g(k,l)] = G(k,l) * W_0 = GX \quad \text{et} \quad \Delta_{\pi/2}[g(k,l)] = G(k,l) * W_{\pi/2} = GY$$

Plusieurs types de matrices de pondération ont été définis. Parmi ceux-ci, nous avons expérimenté les opérateurs de SOBEL et le gradient lissé qui donnent dans la majorité des cas d'excellents résultats.

$$W_0 = \begin{bmatrix} -1 & +0 & +1 \\ -2 & +0 & +2 \\ -1 & +0 & +1 \end{bmatrix} \quad W_{\pi/2} = \begin{bmatrix} -1 & -2 & -1 \\ +0 & +0 & +0 \\ +1 & +2 & +1 \end{bmatrix}$$

Gradient SOBEL

$$W_0 = \begin{bmatrix} -1 & +0 & +1 \\ -1 & +0 & +1 \\ -1 & +0 & +1 \end{bmatrix} \quad W_{\pi/2} = \begin{bmatrix} -1 & -1 & -1 \\ +0 & +0 & +0 \\ +1 & +1 & +1 \end{bmatrix}$$

Gradient Lissé

4.5.3. Mise en oeuvre du seuillage par variation du gradient

Pour chaque pixel traité, le calcul des trois gradients couleurs est réalisé. Le gradient résultat GR est alors le maximum des gradients. L'opérateur fixe une valeur de gradient G correspondant à la valeur minimale d'un pixel contour. La fonction de segmentation est définie de la façon suivante :

$$.F(x,y) = (R,V,B) \text{ si } GR \geq G$$

$$.F(x,y) = 0 \text{ autrement}$$

Ce seuillage nécessite un nombre important de calculs (une trentaine d'additions, 16 multiplications et 3 opérations racine-carrée), pour chaque pixel analysé. Le calcul peut être optimisé en tenant compte de l'approximation suivante pour le module du gradient :

$$\varphi(x,y) = \text{maximum} (\partial g(x,y)/dx , \partial g(x,y)/dy)$$

Ce seuillage permet de connaître uniquement les contours des objets. Une deuxième étape est alors nécessaire pour récupérer le contenu , c'est à dire les pixels internes des objets.

Ce seuillage est particulièrement adapté, lorsque l'objet à extraire est de couleur relativement homogène et repose sur un fond de couleur unie (noire par excellence). Ce type de configuration correspond au contrôle de qualité sur les chaînes de montage et il suffit seulement d'avoir une image avec des niveaux de gris.

Par contre, pour des objets non homogènes (notre cas bien sûr !), ce type de seuillage est inapproprié. Le module du gradient est aussi élevé à l'intérieur des régions que sur les contours.

4.5.2. Suivi de contours sur une image couleur

4.5.2.1. Théorie

La combinaison du module du gradient et de son argument, à partir d'un point du contour, permet de déterminer les contours d'un objet. Cette technique s'appelle **suivi de contours**.

Nous définissons la fonction f en chaque point de l'image de la façon suivante :

$$.f(x,y) = \text{maximum de } (GX,GY) \text{ où } GX \text{ et } GY \text{ sont les dérivées bidirectionnelles de l'opérateur gradient .}$$

L'algorithme, sur une image à analyser où l'objet est noir sur un fond blanc, est le suivant [HOR81] :

.utilisation d'un opérateur gradient en un point du contour.
La direction du gradient est calculée en ce point.

.détermination des successeurs du point courant grâce à la direction du gradient.

.le point suivant choisi est le successeur d'intensité maximum du gradient .

.itération jusqu'à la détection complète du contour.

Chaque pixel a trois successeurs possibles, déterminés d'après le signe des dérivées bidirectionnelles GX et GY de l'opérateur gradient . Le tableau ci-dessous les indique.

Gy \ Gx	< 0	= 0	> 0
< 0	f(i-1, j) f(i-1, j+1) f(i, j-1)	f(i-1, j) f(i-1, j) f(i-1, j-1)	f(i, j+1) f(i-1, j+1) f(i-1, j)
= 0	f(i-1, j-1) f(i, j-1) f(i+1, j-1)	?	f(i+1, j+1) f(i, j+1) f(i-1, j+1)
> 0	f(i, j-1) f(i+1, j-1) f(i+1, j)	f(i+1, j-1) f(i+1, j) f(i+1, j+1)	f(i+1, j) f(i+1, j+1) f(i, j+1)

coordonnées des points à tester

Dans le cas de conflit , c'est à dire où $GX=GY=0$, le successeur du point est alors le point milieu des trois points testés.

4.5.2. Exemple

Le tableau suivant est l'illustration du suivi du contour sur une image codée sur 8 niveaux de gris. Chaque noeud représente les coordonnées d'un pixel et chaque arc, l'intensité du pixel lui succédant. Le contour est donné par les traits en gras.

L'opérateur gradient utilisé est le gradient de SOBEL sur un voisinage de 8 pixel, permettant un meilleur compromis entre le lissage, la détection des micro-contours et le temps de calcul.

Ce type de traitement oblige l'opérateur à saisir au moins un pixel du contour. Pour chaque pixel traité, nous utilisons 3 fois le nombre d'opérations définies dans le seuillage par variation de gradient. Mais à l'inverse de ce dernier, seuls interviennent les pixels du contour. Le traitement est alors très rapide.

Ce traitement est efficace lorsque les objets sont de couleurs relativement bien unies et contrastées. Dans le cas contraire, un changement de couleur interne est analysé comme une transition inter-région.

4.5.3. Suivi de contours sur une image binaire

4.5.3.1. Par opérateur gradient

Pour éviter le défaut précédent, il est possible de combiner les opérations de filtrage (meilleur contraste), les opérations de segmentation (mise en évidence de régions), et le suivi de contours.

Avec les différentes techniques de segmentation, nous obtenons quelques pixels du contour même si cette bande est irrégulière. La fonction de binarisation suivante est ensuite appliquée à l'image segmentée:

$$.f(x,y) = 1 \text{ si } g(x,y) \neq 0 \text{ après segmentation}$$

$$.f(x,y) = 0 \text{ autrement}$$

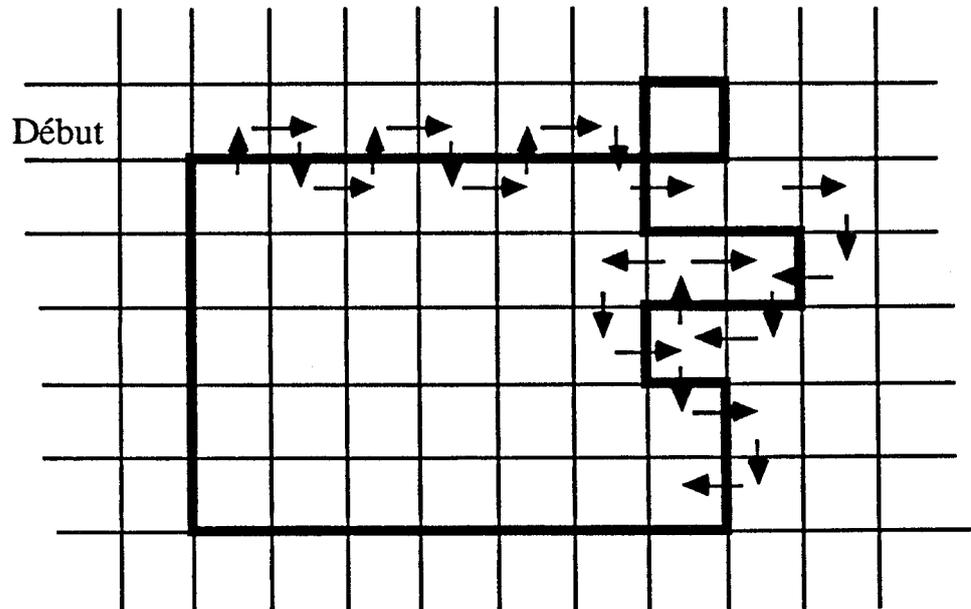
Nous pouvons utiliser avec réussite le suivi du contour sur cette image prétraitée, par l'opérateur gradient expliqué précédemment. Par rapport à une image en couleurs, nous ne calculons qu'un seul gradient, d'où un gain de temps très appréciable (facteur 3).

4.5.3.2. Autres techniques du suivi de contours

Sur des images binaires, les techniques du suivi de contours sont nombreuses, variées et souvent plus performantes qu'avec l'opérateur gradient.

L'algorithme de DUDA et HART (ICA87) consiste à partir d'un pixel appartenant au contour :

- .de tourner à gauche si le pixel appartient à l'objet;
- .de tourner à droite si le pixel est extérieur à l'objet



Le suiveur ne prend que deux directions, la méthode est très rapide. Deux inconvénients à cette méthode. Les pixels reliés selon une direction diagonale sont ignorés tandis que d'autres sont visités deux fois.

Le deuxième algorithme étudié est celui de ROSENFELD et KAK (ICA87). Il consiste à partir du premier point du contour, à balayer ses voisins dans le sens trigonométrique. Le premier voisin noir rencontré est le point suivant du contour.

$X-1, Y-1$ ←	$X, Y-1$ ←	$X+1, Y-1$
$X-1, Y$ ←	X, Y	$X+1, Y$
$X-1, Y+1$ →	$X, Y+1$ →	$X+1, Y+1$

Cet algorithme est lui aussi simple à mettre en œuvre. Il ne nécessite au maximum que huit tests pour déterminer le successeur d'un point. Cet algorithme est retenu pour la recherche de contours. Il est décrit plus amplement dans le chapitre Réalisation.

4.5.3.3. Avantages du suivi de contours

A partir du contour ainsi calculé, il suffit de récupérer les pixels intérieurs de l'objet. Des algorithmes de remplissage de tâches, couramment utilisés en synthèse d'images (DUD73), sont alors nécessaires .

Le principal avantage du suivi de contours est d'obtenir l'ensemble des pixels d'un objet à partir de son enveloppe. Ceci est souvent impossible sur un objet complexe, par simple prototypage (nombreuses imperfections dans l'objet traité).

5. POST-TRAITEMENT

L'image résultante malgré le traitement d'images reste parfois incorrecte. Cette dernière a besoin d'être retouchée, automatiquement voir manuellement pour les objets complexes. Les techniques les plus courantes sont les suivantes :

- .l'utilisation de filtres décrits dans le prétraitement;
- .la suppression des pixels isolés (pixels parasites);
- .l'homogénéisation qui consiste à l'affectation au pixel, de la couleur la plus fréquente avoisinante.

La couleur des pixels situés sur la bordure des objets, correspond à la moyenne entre la couleur de l'objet et celle de la région voisine. Nous intégrons les pixels du contour en leur associant une information supplémentaire qui donne un effet de semi-transparence sur les bords. Cette technique appelée *technique du flou* s'applique particulièrement bien sur des machines qui attribuent 4 composantes au pixel, comme ce sera le cas avec la nouvelle gamme GETRIS :

- .3 composantes couleurs classiques RVB;
- .une composante dite canal P appelé *facteur de transparence* (pondération du pixel par rapport aux pixels de mêmes coordonnées se trouvant sur les autres plans mémoires).

6. Notre solution : le logiciel G-EXTRA

Le logiciel d'extraction G-EXTRA est un logiciel capable d'extraire une portion d'image, en combinant le seuillage d'images par écart sur des prototypes et la détermination du contour d'un objet. La première technique est utilisée seule ou sert de prétraitement à la seconde (image binaire).

Les traitements s'effectuent sur des zones définies par l'opérateur.

Un prétraitement, appelé étalement d'histogramme s'applique à la demande sur ces dernières.

Les opérations de seuillage sont mémorisables pour permettre d'être reproduites.

Les techniques de seuillage conservées, qui sont un bon compromis entre temps de traitement / qualité d'extraction sont les suivantes :

- .seuillage TLS;
- .seuillage RVB en 12 bits ;
- .seuillage composantes 24 bits;
- .seuillage RVB 24 bits.

Le dernier seuillage proposé ne s'utilise que sur des objets difficiles.

L'opérateur a le choix de l'emplacement de l'objet extrait. Plusieurs modes d'insertion sont proposés:

- .l'objet est recopié quelque soit l'état de la zone destination (mode standard)
- .les pixels sont copiés que sur des pixels valides de la zone destination (mode stencil)
- .les pixels sont copiés que sur des pixels invalides de la zone destination (mode non stencil)

L'opérateur a la possibilité de modifier une ou plusieurs composantes-TLS de l'objet pour en modifier le rendu. Pour ce fait , le logiciel permet de choisir facilement une couleur par l'intermédiaire d'une palette de couleurs.

Il s'avère quelquefois plus aisé d'extraire le complément de l'objet que l'objet lui-même. Ce choix est offert à l'opérateur.

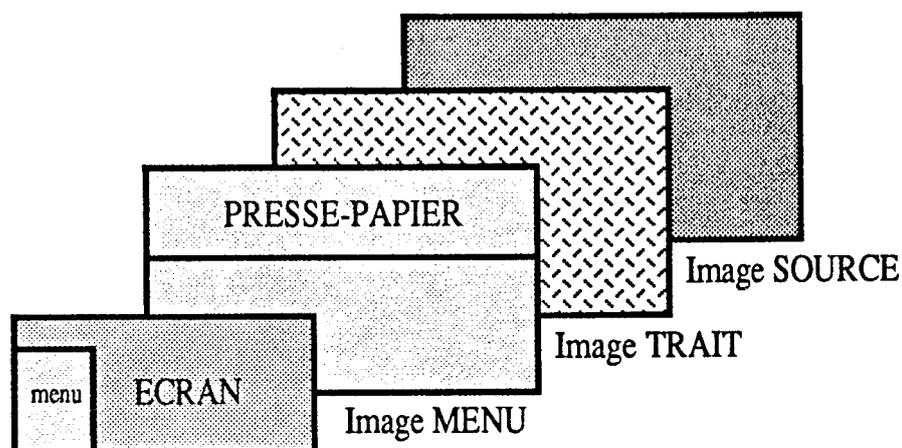
Une suppression des pixels isolés, comme post-traitement, permet d'améliorer la qualité de l'image.

Sur des images difficiles, des retouches manuelles corrigent l'objet.

CHAPITRE 4

REALISATION

1. CONFIGURATION EN PLANS IMAGES



Architecture matérielle G_EXTRA

Le premier plan se nomme plan **MENU**. Les différents menus de l'application G_EXTRA apparaissent en bas à gauche. Sur cette image est définie une vue particulière, appelée **PRESSE-PAPIER**.

Le deuxième plan sert au traitement proprement dit de l'objet (extraction) d'où le nom de **TRAIT**.

Le dernier plan, dit **SOURCE**, mémorise l'objet initial non traité.

Les plans sont superposés. Le plan **MENU** est le plus prioritaire (niveau de profondeur le plus bas), ce qui permet au menu courant d'être toujours visible.

La partie visible des plans se situe dans la portion appelée **ECRAN**. Les deux derniers plans sont mobiles, c'est à dire que l'opérateur a la possibilité de les translater par rapport à l'écran, afin *d'afficher toute l'étendue du plan*. Ces plans sont solidaires pour garder la cohésion entre l'objet traité et l'objet initial.

La vue **PRESSE-PAPIER** (GLI89) est une vue non manipulable. Elle sert à sauvegarder momentanément des portions d'image. Elle se gère comme une pile avec un système de feuilles de sauvegarde. Les plans images sont soit des plans 12 bits, soit des plans 24 bits. Il est nécessaire que le **PRESSE-PAPIER** (plan **MENU**) ait la configuration la plus élevée, pour que cette opération puisse s'effectuer correctement. Le **PRESSE-PAPIER** est placé dans la partie haute du plan **MENU**, ce qui le rend invisible à l'écran. Il est dimensionné au maximum pour sauvegarder 450K pixels.

La *configuration n'est pas figée*, elle est fonction des ressources disponibles. Le logiciel G-EXTRA en profite au mieux pour extraire les objets. La configuration minimale de travail est de trois plans mémoires.

Config. SOURCE	Config. TRAIT	Nombre de Plans
24 Bits	24 Bits	6
24 Bits	12 Bits	5
12 Bits	12 Bits	3

DETERMINATION DE LA CONFIGURATION

2. DESCRIPTION DE L'INTERFACE HOMME-MACHINE

Ce paragraphe décrit la philosophie des relations homme-machine employées par le logiciel G-EXTRA. Elle s'inspire des concepts issus du logiciel G-PAINT (cf présentation logicielle). Il ne s'agit pas d'un manuel d'utilisation, il explique comment les contraintes liées aux RHM, décrites précédemment ont été respectées.

2.1. Généralités

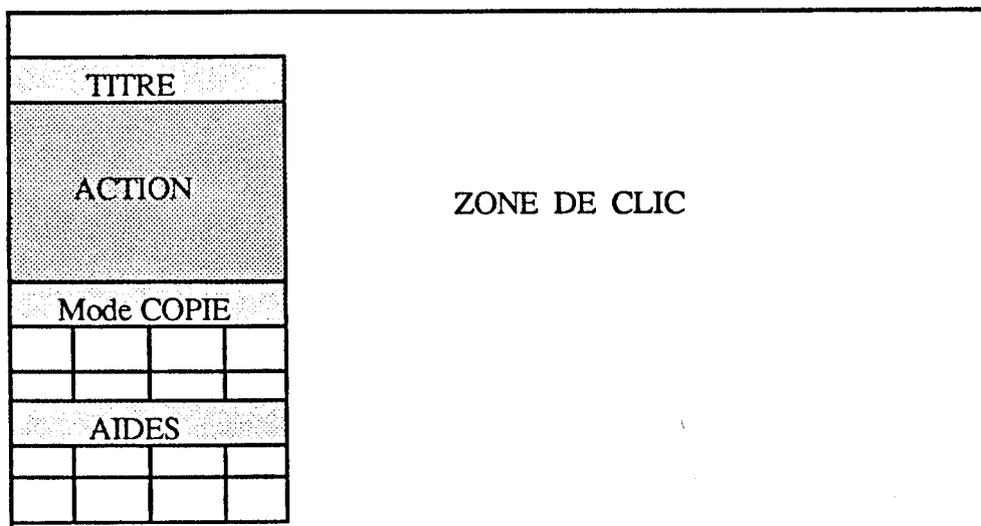
Quelque soit l'action, l'écran se présente de la façon suivante :

- .une zone **menu**;
- .une zone de **clic**.

Tout traitement est identifié par une icône. Une icône a trois états possibles qui sont *identifiables par une couleur*, par exemple :

- .rouge = l'icône est non valide;
- .vert = l'icône est valide mais non sélectionnée;
- .jaune = l'icône est sélectionnée.

Suivant le type de traitement, le curseur est représenté par un symbole différent. A titre d'exemple, le dessin du curseur est un poisson pour la saisie d'un prototype.



DESCRIPTION D'UN MENU

Pour une plus grande *interactivité*, et pour utiliser au mieux la bibliothèque de haut niveau qu'est G_DIAL (GDI89), le logiciel G_EXTRA emploie pour l'interface Homme - Machine, les outils de dialogue graphique qui sont :

- .les *choix interactifs* qui permettent à l'utilisateur de sélectionner une réponse parmi une liste d'entités représentées sous forme d'icônes;
- .les "*boutons*" qui ne sont que des choix ne comportant qu'une seule entité;
- .les *ascenseurs* qui récupèrent dans un intervalle de valeur une information numérique;
- .les *choix de fichiers* qui offrent la possibilité de choisir un fichier à partir d'un répertoire donné;
- .les *messages interactifs* qui affichent un message sur le moniteur vidéo, et proposent à l'utilisateur la sélection d'une réponse.
- .les *valeurs interactives* qui saisissent une chaîne de caractères, affichée sur le moniteur vidéo, à partir du clavier du calculateur hôte.

2.2. Description de la zone MENU

Dans le menu respectant les normes N.T.S.C. (hauteur < 480 pixels), nous distinguons trois parties :

- .une partie **action**;
- .une partie **mode de copies**;
- .une partie **aides**.

2.2.1. Fonctions permanentes et " claviers"

La majorité des traitements nécessite la détermination d'une ou de deux zones. Il s'agit de la zone origine qui contient l'objet initial, et de la zone de destination qui désigne l'emplacement de l'objet traité.

Le champ "**mode de copies**" définit le type des zones pour les opérations de copies :

- a) pour la zone origine :
 - .copie de toute la zone;
 - .copie de l'objet;
 - .copie du complémentaire de l'objet.

- b) pour la zone destination (cf notre solution : le logiciel G-EXTRA):
 - .copie standard;
 - .copie stencil;
 - .copie non stencil.

De plus, l'opérateur connaît la phase de saisie en cours (origine ou destination), par l'intermédiaire d'une icône sélectionnée dans le champ mode de copies.

Le champ "**aides**" offre comme son nom l'indique, des aides à l'opérateur. Les touches F1 à F8 du clavier reprennent ces fonctions, dans le même ordre que ci-dessous. Ceci est surtout utile pour faire réapparaître le menu lorsque celui-ci a été rendu invisible! Ces fonctions permettent :

- .de déplacer les images TRAIT et SOURCE. Cette action commence par un clic sur l'icône "déplacement" et se poursuit jusqu'à un nouveau clic sur celle-ci. Les plans, pendant ce traitement, suivent le déplacement du curseur qui est symbolisé par une main;

- .de recentrer ces mêmes plans par rapport à l'écran;

- .de rendre visible ou invisible le plan MENU pour évaluer les résultats des traitements sur un écran complet;

- .de valider le mode MAJ ou de l'annuler. Cette fonction est la **plus importante de ce RHM**. Ce mode visualise les résultats de l'action courante, automatiquement et ceci après chaque modification du contexte du traitement (exemple : modification de la tolérance pour les prototypes). Le résultat n'est pas définitif, l'utilisateur a la possibilité *d'annuler l'action à tout moment*, ce qui a pour effet de restaurer l'image initiale;

- .de sélectionner comme zone origine ou destination tout le plan mémoire;

.de réafficher le fantôme de la zone origine après une opération d'annulation ;

.d'annuler l'opération courante avec restitution de l'image, si le mode MAJ est présent;

.de valider l'opération courante.

Les champs "mode de copies" et "aides" sont constamment présents à l'écran car ils regroupent des *fonctions communes* à plusieurs actions.

D'autres fonctions d'aides sont accessibles à l'utilisateur. Mais celles-ci ne sont exécutables qu'à partir du clavier. Il s'agit :

- .de la visibilité / de l'invisibilité du plan TRAIT (touche ALT+F1);
- .de la visibilité / de l'invisibilité du plan SOURCE (touche ALT + F2).

2.2.2. Description du champ ACTION

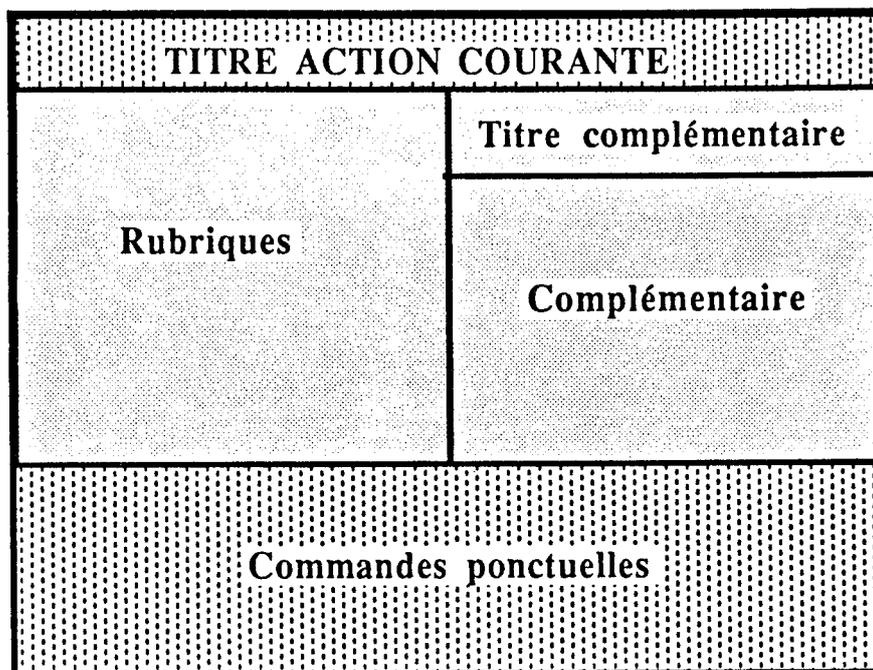
C'est avec ce champ, que l'opérateur :

- . choisit le traitement courant dans le menu principal;
- . exécute une action particulière de ce traitement.

Trois parties composent la zone ACTION :

- .le champ **TITRE**;
- .le champ **RUBRIQUE** et **COMPLEMENTAIRE**;
- .le champ **COMMANDES PONCTUELLES**.

a) La fin de l'action courante s'effectue à *n'importe quel moment* choisi par l'utilisateur, par un clic dans le titre de l'action. Si le traitement n'a pas été validé et le mode MAJ était aussi présent, la fin provoque la *restauration des images* telles qu'elles étaient avant le début de l'action.



DESCRIPTION DU CHAMP ACTION

b) Le champ RUBRIQUE permet à l'utilisateur de *connaître par plusieurs icônes, toutes les options valides relatives à l'état du traitement courant*. Pour modifier une des options, il suffit de cliquer sur l'icône correspondante. Son état est sélectionné, et dans la partie COMPLÉMENTAIRE, s'affiche le titre de la catégorie, ainsi que les différentes valeurs possibles. La sélection d'un nouveau choix dans la partie complémentaire réactualise la zone rubrique.

L'exemple suivant explicite la notion de rubrique. Nous voyons les quatre options en cours pour l'extraction :

.l'extraction s'effectue suivant un delta par rapport aux prototypes saisis, égal à plus ou moins 5. C'est cette option (couleur sélectionnée) que l'opérateur est en train de modifier, en se servant de l'ascenseur défini dans la zone complémentaire;

.le type d'algorithme est le seuillage TLS;

.l'utilisateur n'a pas demandé d'automatisation;

.nous effectuons un prétraitement de type égalisation d'histogramme.

± 5	PRECISION
TLS	
	
PRE	

EXEMPLE : RUBRIQUE EXTRACTION

c) Le dernier champ **COMMANDES PONCTUELLES** modifie par des actions ponctuelles, le contexte du traitement. L'utilisateur clique sur l'icône désirée pour réaliser l'action. L'action se termine immédiatement, comme l'effacement des prototypes saisis, ou après la validation d'une autre icône, comme la saisie de ces mêmes prototypes.

2.3. Zone de "CLIC"

L'opérateur saisit dans cette zone les différentes informations relatives à l'image traitée, par déplacement du curseur (exemple: saisie des informations couleurs relatives aux différents prototypes).

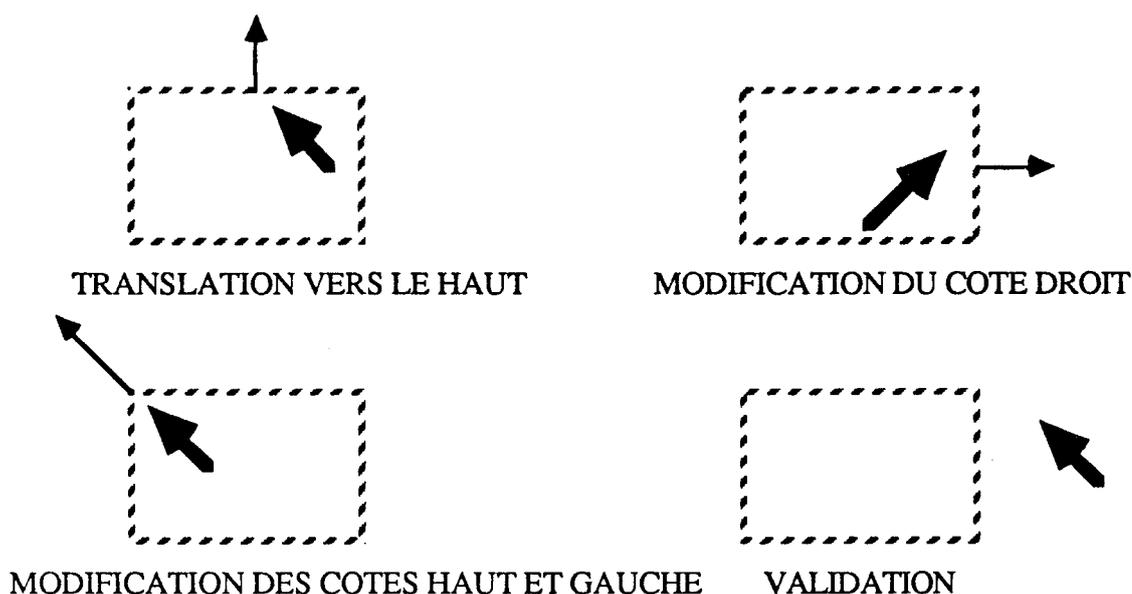
2.3.1. Saisie de zones

Les zones origines et destinations sont de forme rectangulaire.

La zone origine est saisie par appui du dispositif de dialogue sur le point bas-gauche de la région, avec relâché dans le point haut-droit de celle-ci. Le *fantôme de la zone* apparaît. Jusqu'à sa validation, l'utilisateur déforme et déplace à souhait cette zone (*ajustement*). Ceci est très important en mode automatique pour "recaler" la zone initiale sur le nouveau plan image (image translaturée ou "zoomée"). La validation s'effectue par un "clic" à l'extérieur de la zone, ou par la fonction d'aides appelée validation. Le curseur a la forme d'un crochet pendant cette opération.

La saisie de la zone destination prend par défaut le même emplacement que celle initiale. Cette zone n'est pas déformable. Sa validation correspond aussi à la validation de l'action en cours.

Ces zones sont appelées **morphologies** (GDI89).



EXEMPLE DE SAISIE DE MORPHOLOGIE

La bibliothèque graphique effectue cette saisie .

En mode MAJ , les zones modifiées sont sauvegardées dans le PRESSE-PAPIER. Si cela est impossible (place insuffisante), le logiciel informe l'utilisateur par un signal sonore couramment appelé "bip". Il peut réduire la zone ou passer outre (*souplesse d'utilisation*) . Dans ce cas, le mode MAJ est abandonné et la validation de la zone entraîne sa destruction irrémédiable .

2.3.2. Saisie d'informations "images"

La saisie des informations "images" s'effectue simplement par appui du dispositif de dialogue sur la zone de "clic", après la sélection de la zone origine (exemple: saisie d'un point interne à l'objet). L'action n'est réalisée que si le curseur se trouve à l'intérieur de la zone, matérialisée par un fantôme (*protection*). Dans le cas contraire , un signal sonore retentit. Le dessin du curseur est différent entre l'intérieur et l'extérieur de la zone.

3. DESCRIPTION DES MENUS

Ce paragraphe décrit l'organisation en menus de l'application G-EXTRA. Le menu principal fait apparaître 8 fonctions qui sont (cf annexe 4 photographie 1):

- .une action de restitution;
- .une action de sauvegarde;
- .une action d'effacement;
- .une action de fin;
- .une action de seuillage;
- .une action de détournage;
- .une action de retouche;
- .une action de copie.

3.1. Fonctions classiques

Les quatre premières sont des fonctions classiques des logiciels ATALIS. Elles sont seulement adaptées à l'application G-EXTRA.

3.1.1. Restitution

Cette fonction effectue le chargement d'un fichier "image", sur le plan TRAIT ou SOURCE, dans une zone destination.

3.1.2. Sauvegarde

Cette fonction a le rôle inverse de la précédente. Elle réalise la sauvegarde d'une portion d'image ou de l'image complète. Cette dernière se trouve :

- .soit sur le plan TRAIT. Il s'agit alors de la sauvegarde d'un objet extrait;
- .soit sur les deux plans TRAIT et SOURCE. Il s'agit alors d'une sauvegarde vidéo. Nous pouvons ainsi insérer un objet dans un contexte, après son extraction ou après avoir modifié son rendu.

3.1.4. Effacement

C'est avec cette fonction, que l'opérateur efface une partie des images TRAIT ou SOURCE. L'effacement correspond à la réécriture de la zone concernée avec une couleur invisible qui est le noir. Ceci a pour effet, lorsqu'il s'agit du plan TRAIT, de faire apparaître l'image SOURCE en arrière plan.

3.1.5. Fin

La fin du logiciel s'effectue en cliquant sur l'icône FIN du menu principal. Une *demande de confirmation* apparaît. Dans l'affirmative, le logiciel propose d'exécuter les autres logiciels de la gamme ATALIS. Dans le cas contraire, le logiciel G-EXTRA se poursuit.

3.2. Extraction

Cette fonction réalise l'extraction d'un objet par une opération de seuillage (cf annexe 4 photographie 1 et photographie 2).

Les paramètres de l'extraction, donnés dans la zone rubrique et directement modifiables par l'opérateur, sont :

- .la précision de l'extraction;
- .le type d'algorithme qui sert à l'extraction :
 - .seuillage TLS;
 - .seuillage RVB 12 bits;
 - .seuillage RVB 24 bits (par rapport aux prototypes);
 - .seuillage RVB 24 bits (par rapport aux composantes).
- .le mode d'automatisation ;
- .l'utilisation ou non de la phase de prétraitement (égalisation d'histogramme).

L'ajout d'un nouvel algorithme de seuillage correspond *simplement* à l'insertion d'une autre entité, dans le complémentaire associé aux types.

L'utilisateur a la possibilité de mémoriser les différents paramètres relatifs à une extraction dans un fichier dit de sauvegarde. Bien sûr, à la demande, il peut mettre à jour le contexte du traitement en récupérant ces paramètres. Le mode d'automatisation autorise ou non les opérations de sauvegarde et de restitution.

Les commandes ponctuelles proposées sont les suivantes :

- .la saisie d'un ou plusieurs prototypes sur la zone origine;
- .l'annulation des prototypes précédemment saisis;
- .la gestion de l'opération de sauvegarde des paramètres d'extraction (ajout ou suppression d'un enregistrement);
- .la gestion de l'opération de restitution (déplacement dans le fichier restitution);
- .la gestion du prétraitement : sélection d'un canal RVB pour le traitement de l'égalisation d'histogramme.

Hormis la phase de saisie de prototypes, seul le plan de travail TRAIT est visible.

Le changement dans le mode d'automatisation permet :

- . soit de sélectionner le nom du fichier pour la sauvegarde ou la restitution;
- . soit de fermer ces mêmes fichiers.

Tous les modes proposés dans le champ "mode de copies" sont possibles.

3.3. Détourage

Cette fonction réalise le détourage d'un objet, après l'opération de seuillage décrite ci-dessus (cf annexe 4 photographie 1 et photographie 3). Celle-ci a permis d'obtenir un contour sommaire de l'objet.

Un seul paramètre est ajustable. Il s'agit du sens de l'opération de détourage (sens trigonométrique ou sens des aiguilles d'une montre). Il permet, lorsque l'algorithme de détourage a atteint une des limites de la zone origine, de repartir dans l'autre sens, sans obliger l'utilisateur à une nouvelle indication d'un point du contour (*souplesse d'utilisation*).

Après la saisie de la zone origine, les commandes ponctuelles sont valides. Elles proposent :

.une action pour déterminer le premier point du contour. L'utilisateur indique dans la zone origine, une direction avec le dispositif du dialogue. Le logiciel G-EXTRA fait alors le reste, c'est à dire qu'il examine dans cette direction les pixels, jusqu'à ce qu'il rencontre le premier appartenant à l'objet. Ce pixel est dit "valide", car il est différent de la couleur noire (couleur invisible). Le détourage commence alors dès le relâché du dispositif de dialogue. D'autres points de contour peuvent être donnés pendant ce traitement de la même façon. Le contour est dessiné sur le plan TRAIT par un tracé de pixels blancs. Si le curseur rencontre des pixels valides, avant le relâché du dispositif de dialogue, le tracé du contour s'effectue manuellement

.une action d'effacement. Si pour une raison ou une autre (mauvaise étape de segmentation), le tracé du contour est incorrect, le logiciel G-EXTRA propose une action de correction. L'utilisateur déplace le curseur jusqu'à rencontrer la zone de contour à effacer. Après relâché du dispositif de dialogue, l'effacement du contour commence, et ceci jusqu'à l'interruption par un clic dans l'icône "effacement". L'effacement est réalisé par la réécriture des pixels appartenant au contour, à partir de ceux lus dans le plan SOURCE.

.une action de sélection d'un point appartenant à l'objet. Cette action provoque la récupération totale des pixels appartenant à l'objet (cf fonctions relatives au détourage) et valide les modes de copies. Ceux-ci sont limités, il s'agit de l'extraction de l'objet ou de son complément.

3.4. Correction

Cette fonction "retouche" les objets seuillés ou détourés par les traitements décrits dans les deux paragraphes précédents.

La sélection des pixels à retoucher est faite par déplacement du dispositif de dialogue sur le plan TRAIT.

Deux entités composent la partie rubrique :

- .le mode de retouche;

- .la taille de la retouche.

L'utilisateur a la possibilité soit d'effacer des pixels n'appartenant pas à l'objet, soit de compléter l'objet par des pixels lus sur le plan SOURCE.

Il fixe aussi la taille de la brosse, c'est à dire le nombre de pixels voisins retouchés, par pixel sélectionné avec le dispositif de dialogue.

Le plan SOURCE n'est visible qu'en mode récupération et qu'avec le dispositif de dialogue enfoncé.

3.5. Copie

Cette fonction ajoute un plus par rapport à la copie, réalisée dans les actions de seuillage ou de détourage.

Les couleurs de l'objet recopié sont modifiables à la demande par l'utilisateur, par la sélection d'une couleur sur un nuancier .

Au niveau de la rubrique, nous avons le mode de modification. Il permet à l'opérateur de conserver une combinaison des trois composantes TLS de l'objet, pendant cette recopie.

La palette de couleurs offre un large éventail de nuances mais leur nombre est de toute façon limité. Pour pallier à ce problème, il est prévu :

- .soit une interpolation entre deux couleurs du nuancier;

- .soit la saisie d'un niveau de gris (valeur inexistante dans le nuancier);

- .soit la saisie directe des valeurs numériques des composantes RVB d'une couleur.

4. Réalisation de l'interface homme-machine

4.1. Généralités

Les bibliothèques du système ATALIS sont des bibliothèques de haut niveau, orientées langage *objet*. La conception d'un système de dialogue est des plus simples. Pour chaque vue définie, nous pouvons :

- .lui associer un cadre de vue avec un titre;
- .l'attacher à une vue "mère" ou lui faire se rattacher d'autres vues;
- .lui associer une fonction d'affichage. L'ordre d'affichage exécute cette dernière;
- .lui associer une fonction de dialogue. Un dialogue sur cette vue, active cette fonction;
- .l'effacer ainsi que les vues dont elle est la racine, par une seule primitive.

Les vues particulières tels que les choix interactifs, les ascenseurs, les valeurs interactives ou les choix de nom de fichier, ont des fonctions de dialogue puissantes. Elles permettent de récupérer par exemple :

- .la valeur saisie ou le nom du fichier sélectionné;
- .l'entité du choix sélectionné (numéro)...

De même, d'autres primitives permettent de gérer le dispositif de dialogue. Nous pouvons lui associer :

- .une fonction de "clic" qui est activée lors d'un début de dialogue;
- .une fonction de gestion d'écho qui indique dans quelle phase se trouve le dialogue;
- .des fonctions d'affichage qui visualise la bonne icône "curseur" suivant le contexte.

4.2. Programme principal

En utilisant ces outils, le traitement principal devient :

.action <<G-EXTRA>>

.Initialisation de l'application

- initialisation des ressources
(cf paragraphe configuration en plans images)
- initialisation des variables de l'application
- définition des cadres, des images et des différentes vues
- initialisation de l'état des entités des choix
- initialisation des tables de conversions RVB <=>TLS
(cf paragraphe conversion RVB <=>TLS)

.Affichage du menu principal

.tantque (la fin du programme non demandée)

faire

.tantque (dispositif de dialogue non actif)

faire

**.examen des touches du clavier avec exécution de leur
fonction associée (cf paragraphe fonctions permanentes
et clavier)**

**.activation d'une fonction spécifique suivant le traitement
courant**

fin tantque

.tantque (dispositif de dialogue actif)

faire

**.<<appel par le système ATALIS des fonctions associées
aux vues >>**

fin tantque

fin tantque

.fin action <<G_EXTRA>>

L'exécution d'une fonction spécifique, déterminée par les fonctions de dialogue, dans le corps de la boucle principale, permet de ne pas interrompre le dialogue. Ceci est particulièrement intéressant pour des traitements longs, tel que l'élaboration du contour d'un objet.

4.3. Gestion du curseur

4.3.1. Affichage du curseur

A la position haute du curseur (dispositif de dialogue inactif), est associée une fonction appelée HAUT_MORPH. Elle gère l'icône du curseur suivant l'état courant de l'application (action en cours, appartenance ou non à la morphologie...).

.action <<HAUT-MORPH>>

.si (contexte courant est un contexte spécifique)

alors

.remplacement de l'icône curseur par la nouvelle icône

sinon

.garder l'icône

fin si

fin action

4.3.2. Saisie d'informations.

A la partie de l'écran autre que le menu, est associée une fonction de clic, appelée CLIC_DESSIN. A chaque action du menu principal est associée une fonction de clic spécifique (CLIC_"nom de l'action").

La fonction CLIC_DESSIN gère la translation des plans TRAIT et SOURCE. Lorsque cette action n'est pas en cours, elle active la fonction de clic spécifique.

.action <<CLIC_DESSIN>>

.si (déplacement demandé)

alors

tantque (dispositif de dialogue actif)

faire

.translation des plans suivant la position courante du curseur

fin tantque

sinon

.activation de la fonction spécifique de clic

fin si

fin action <<CLIC_DESSIN>>

La fonction spécifique effectue la saisie des morphologies. Pour d'autres traitements, comme la saisie de prototypes, une fonction d'écho est associée à l'action courante. La fonction de clic est alors inactive.

4.1. Affichage des morphologies

Ce paragraphe décrit les fonctions les plus importantes de l'interface homme-machine. Elles permettent :

- .le déplacement et la déformation des morphologies avec l'affichage du fantôme associé;
- .la gestion du mode MAJ** avec la sauvegarde de la zone dans le presse-papier, sa mise à jour, sa validation ou sa restauration .

La fonction spécifique à exécuter lors de la validation ou lors de la mise à jour est passée en paramètre d'appel.

La morphologie courante n'est qu'une vue particulière référencée par la variable MORPH_COURANTE, mise à jour par les différentes actions.

Cette gestion n'utilise que deux booléens :

- .MAJ = Etat du mode de gestion MAJ
Mise à jour par la fonction d'aides MISE-A-JOUR
_vrai si le mode MAJ est présent
- .PRESSE = Etat du presse-papier
Mise à jour par les fonctions d'affichage des morphologies
_vrai si une morphologie a été sauvegardée dans le presse-papier

Les fonctions qui assurent cette gestion sont au nombre de quatre. Il s'agit :

- .COPIER-MORPH
- .EFFACEMENT-MORPH
- .AFFICH-MORPH
- .VALID-MORPH

4.1.1. COPIER-MORPH

Cette fonction sauve la morphologie courante dans le presse-papier. Lorsque cette opération est possible, c'est à dire si la place est suffisante dans le presse-papier, la fonction met à jour le booléen PRESSE.

Cette fonction retourne le booléen _vrai si l'opération de sauvegarde a eu lieu (_faux sinon). La sauvegarde s'accompagne de l'ouverture d'une nouvelle feuille dans le presse-papier.

.action <<COPIER-MORPH>>

.val = _faux

.consultation des coordonnées de la morphologie courante

.si (la place est suffisante dans le presse-papier pour la sauvegarde)

alors

.val = _vrai

.ouvrir une nouvelle feuille dans le presse-papier

.effectuer la sauvegarde de la morphologie

.mémorisation de l'action de sauvegarde

PRESSE = _vrai

fin si

.retourner (val)

fin action <<COPIER-MORPH>>

4.4.2. EFFACEMENT-MORPH

Cette fonction efface le fantôme de la morphologie courante. Dans le cas où cette dernière a été mémorisée dans le presse-papier, la zone est restaurée à partir de la sauvegarde contenue dans le presse-papier. Le booléen PRESSE reprend la valeur _faux.

.action <<EFFACEMENT-MORPH>>

.effacement du fantôme de la morphologie à l'écran

.si (PRESSE)

alors

.la morphologie courante a été sauvegardée dans le presse-papier

.restauration de la zone à partir de la sauvegarde presse-papier

.suppression de la feuille de sauvegarde du presse-papier

.PRESSE = _faux

fin si

fin action <<EFFACEMENT-MORPH>>

4.4.3. AFFICH-MORPH

Cette fonction permet :

.soit de mémoriser la morphologie courante dans le presse-papier si le mode MAJ est présent. Elle exécute la fonction spécifique de traitement passée en paramètre, si la sauvegarde a été possible.

.soit d'afficher le fantôme de la zone dans le cas contraire

```

.action <<AFFICH-MORPH(fct)>>
    .si (MAJ)
        alors
            .si ( COPIER-MORPH())
                alors
                    .la sauvegarde de la morphologie a été réalisée dans le
                    presse-papier
                    .exécution de la mise à jour de la zone par appel de la fonction
                    spécifique de traitement fct
                sinon
                    .la sauvegarde dans la presse-papier est impossible
                    .signal sonore "bip"
                fin si
            sinon
                .affichage du fantôme de la morphologie
                .PRESSE = _faux
            fin si
    fin action <<AFFICH-MORPH>>

```

4.4.4. VALID-MORPH

Cette fonction réalise la validation de la morphologie courante. Si le traitement de la zone a déjà été visualisé par la fonction précédente (mode MAJ présent), il suffit de libérer l'emplacement de la sauvegarde dans le presse-papier, par dépliement de la feuille de sauvegarde. Dans le cas contraire (mode MAJ absent), nous exécutons la fonction spécifique de traitement.

```

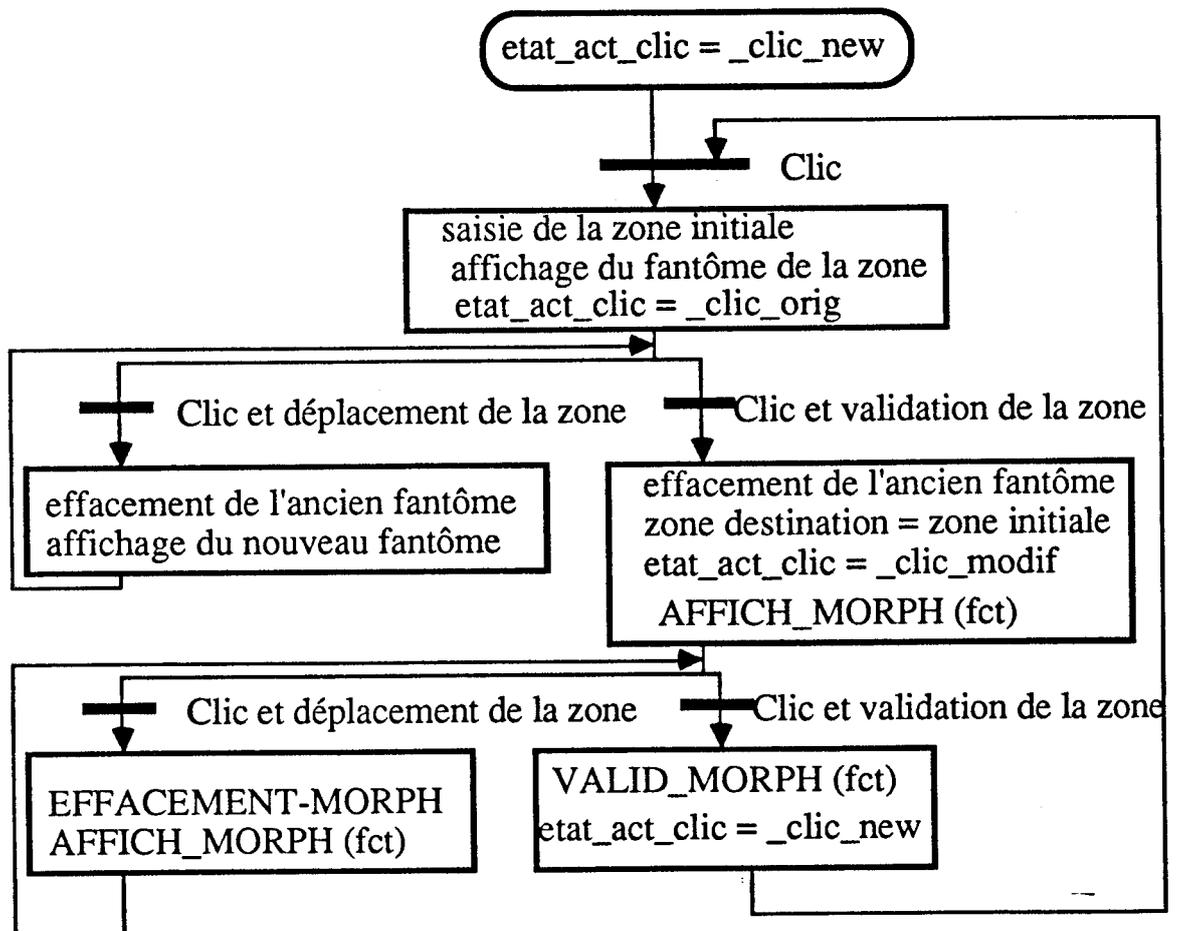
.action <<VALID-MORPH(fct)>>
    .effacement du fantôme de la morphologie courante
    .si ( PRESSE)
        alors
            .la fonction spécifique est déjà exécutée ( cf AFFICH-MORPH)
            .libérer l'emplacement de sauvegarde dans le presse-papier par
            dépliement de la feuille de sauvegarde
            .PRESSE = _faux
        sinon
            .exécution de la fonction spécifique de traitement fct
        fin si
    fin action <<VALID-MORPH>>

```

4.5. Saisie des morphologies

A chaque traitement est associée une fonction de clic appelée CLIC-"nom de l'action". Ces fonctions sont construites de façon similaire. Une variable qui a pour nom ETAT_ACT_CLIC, indique la phase courante de saisie :

- .-clic_new : saisie de la morphologie origine
- .-clic_orig : déplacement, déformation et validation de la zone origine
- .-clic_modif : déplacement et validation de la zone destination
- .-clic_attente : attente d'une action extérieur (exemple action de détournage avant saisie de la zone destination)



EXEMPLE DE SAISIE DE MORPHOLOGIE

Cette gestion utilise les fonctions d'affichage des morphologies décrites précédemment. Suivant le type de traitement en cours, la modification de ETAT_ACT_CLIC s'accompagne de l'actualisation des différentes entités des fonctions d'aides et du mode de copies. A l'inverse, la sélection d'une fonction d'aides influe sur cette saisie, en modifiant la valeur de la variable ETAT_ACT_CLIC (cf paragraphe "gestion de l'action courante").

4.6. Gestion de l'action courante

Il existe plusieurs possibilités nommées "étapes" pour activer une action. Il s'agit de :

- .étape_debut = début de l'action courante.
- .étape_fin = fin de l'action courante.
- .étape_affich = réaffichage de la zone origine après une annulation
- .étape_maj = génération du traitement correspondant à l'action courante avec sauvegarde de la zone destination dans le presse-papier, si le mode MAJ est présent.
- .étape_annul = annulation de l'opération courante avec restitution de la zone destination à partir de la sauvegarde dans le presse-papier.
- .étape_valid = validation de l'opération courante avec libération éventuelle de l'emplacement dans le presse-papier (fonction VALID-MORPH).
- .étape_destin = modification du plan mémoire contenant la vue destination (pour les traitements relatifs à la restitution ou à l'effacement).
- .étape_tout = sélection de tout le plan mémoire comme zone de traitement.

L'ordre d'activation est donné :

.par l'intermédiaire de la sélection d'un nouveau traitement à partir du menu principal (*étape_debut*).

.par le clic dans le titre de l'action courante (*étape_fin*).

.par l'intermédiaire d'une des fonctions d'aides ou mode de copies, décrites précédemment (*étape_affich, étape_maj, étape_annul, étape_tout*).

.par la modification des options des rubriques ou par une modification de contexte avec les fonctions exécutées dans le champ "Commandes ponctuelles" (comme par exemple la saisie de nouveaux prototypes) . **En mode MAJ, nous avons alors instantanément le résultat (*étape_maj*).**

La sélection d'une action dans le menu principal initialise la variable ACT_COURANTE avec le numéro d'entité de l'action.

Nous avons déclaré un tableau nommé TEX_ACTIONS, qui contient l'adresse de la fonction principale de l'action courante (EX_"mnémonique de l'action courante").

L'activation correspond à l'exécution d'une fonction générale EX_ACTIONS qui regroupe les traitements communs à toutes les actions. Il s'agit en particulier des étapes de début et de fin de traitement. EX_ACTIONS appelle à son tour la fonction spécifique courante donnée par TEX_ACTIONS [ACT_COURANTE]

.action <<EX_ACTIONS (étape)>>

.choix (étape)

quand étape_début :

.ETAT_ACT_CLIC = _clic_new (cf saisie des morphologies)
 .effacement du menu principal
 .appel de TEX_ACTIONS [ACT_COURANTE] (étape_début)

quand étape_fin:

.appel de TEX_ACTIONS [ACT_COURANTE] (étape_fin)
 .ETAT_ACT_CLIC = _clic_rien
 .affichage du menu principal
 .ACT_COURANTE = Absent

défaut :

.appel de TEX_ACTIONS [ACT_COURANTE] (étape)

fin choix

fin action <<EX_ACTIONS>>

L'étape "étape_début" des fonctions spécifiques de traitement initialise les différentes vues relatives à l'action courante ainsi que ses propres variables. L'étape "étape_fin" efface ces vues. Le traitement des autres étapes est similaire à celui de la saisie des morphologies, c'est à dire qu'il dépend de la variable ETAT_ACT_CLIC et qu'il utilise les fonctions définies dans le paragraphe "affichage des morphologies". L'exemple suivant correspond à l'étape "étape_maj" d'un traitement spécifique :

.si (ETAT_ACT_CLIC eq _clic_modif)

alors

.la zone origine a été saisie => exécuter l'action de MAJ
 EFFACEMENT_MORPH(fct)
 AFFICH_MORPH(fct)

fin si

4.7. Gestion des rubriques

Dans la phase "étape_début", les fonctions spécifiques initialisent le tableau TAB_COMPL_RUB avec les références des vues des différents complémentaires qui composent le champ "rubrique". La variable RUBRIQUE_COURANTE contient le numéro de l'entité sélectionnée dans ce champ. A la rubrique sont associées une fonction de traitement TRT_"RUBRIQUE" et une fonction d'affichage AFF_"RUBRIQUE".

La fonction de traitement du complémentaire TRT_"COMPL" met à jour une variable VAR_"COMPL", qui permet à AFF_"RUBRIQUE" d'afficher la bonne icône dans la rubrique avec la bonne police de caractères.

La *modification* d'une valeur d'un complémentaire *active* l'action courante avec l'étape "étape_maj".

.action <<TRT_"RUBRIQUE">>

.si (sélection d'un nouveau complémentaire de numéro 'num')

alors

.effacement du complémentaire courant dont la vue est
TAB_COMPL_RUB [RUBRIQUE_COURANTE].

.rendre valide l'entité de numéro RUBRIQUE_COURANTE dans
le champ rubrique.

.RUBRIQUE_COURANTE = num

.rendre sélectionnée l'entité de numéro RUBRIQUE_COURANTE dans
le champ rubrique.

.affichage du complémentaire courant dont la vue est
TAB_COMPL_RUB [RUBRIQUE_COURANTE].

fin si

fin action <<TRT_"RUBRIQUE">>

.action <<TRT_"COMPL">>

.si (sélection de l'entité 'num' dans le complémentaire)

alors

.rendre valide l'entité VAR_"COMPL" dans le complémentaire

.VAR_"COMPL"= num

.rendre sélectionnée l'entité VAR_"COMPL" dans le complémentaire

.affichage de l'entité RUBRIQUE_COURANTE dans le champ
rubrique avec la nouvelle sélection VAR_"COMPL"

.EX_ACTIONS (_étape_maj)

fin si

fin action <<TRT_"COMPL">>

5. Fonctions spécifiques du traitement d'images

Les fonctions spécifiques du traitement d'images sont appelées dans les étapes, "étape_maj" et "étape_valid" de l'interface homme-machine, ou lors d'actions de clic comme la saisie de prototypes.

5.1. Fonctions relatives à l'extraction

5.1.1. Saisie de prototypes

Le tableau TAB_PROTO, dimensionné à _nb_max_proto (1024) éléments, mémorise les prototypes en suivant le dispositif de dialogue, après que l'opérateur est cliqué sur l'icône "saisie de prototypes". La saisie n'est réalisée qu'à l'intérieur de la zone origine de traitement. Elle s'effectue d'après le modèle de couleurs RVB.

La variable NB_PROTO indique le nombre de prototypes actuellement saisis. L'effacement des prototypes n'est autre que la remise à zéro de cette variable.

Pour éviter un nombre trop important de prototypes, le prototype courant n'est enregistré que s'il n'est pas déjà mémorisé dans TAB_PROTO.

.action <<SAISIE_PROTO>>

 .récupération des coordonnées courantes du dispositif de dialogue

.si (nous sommes à l'intérieur de la zone origine)

alors

 .récupération des composantes du pixel

.si (le pixel n'est pas déjà mémorisé et NB_PROTO < _nb_max_proto)

alors

 .mémorisation du prototype dans TAB_PROTO[NB_PROTO]

 NB_PROTO = NB_PROTO + 1

fin si

sinon

 .signal sonore "bip"

fin si

fin action <<SAISIE_PROTO>>

5.1.2. Exécution du seuillage

Nous décomposons cette opération en deux phases :

 .une phase d'initialisation :INIT_SEUILLAGE;

 .une phase de seuillage proprement dite :EXEC_SEUILLAGE.

La première est spécifique, elle dépend de l'algorithme utilisé, tandis que la seconde est commune. Cette dernière teste la validité de chaque pixel de la zone origine, ligne après ligne. Elle réalise les combinaisons entre la zone origine et la zone destination, en fonction de la sélection du mode de copies souhaité et de la validité du pixel traité. Elle appelle pour chaque pixel, une routine SEUILLAGE qui dépend du type de seuillage. Cette fonction retourne soit ce pixel s'il est valide, soit le pixel noir (couleur invisible) s'il est invalide.

5.1.2.1. Seuillage RVB 12 bits ou seuillage TLS

Ces deux algorithmes de seuillage sont similaires, la seule différence est que l'un travaille dans le modèle de couleurs RVB et que l'autre dans celui TLS.

Quelque soit le modèle de couleurs, nous avons 4096 couleurs différentes. Le tableau PROTO_VALID est un tableau de booléens de 4096 éléments, un pour chaque couleur. Il indique la validité de cette couleur en fonction des prototypes saisis ainsi que l'écart tolérable sur chaque composante (e).

.action <<INIT_SEUILLAGE>>

.initialisation du tableau PROTO_VALID à _faux

.pour (i =0; (i ne NB_PROTO); i = i +1)

.récupération des composantes (c_1, c_2, c_3) du prototype

TAB_PROTO[i] avec conversion dans le modèle de couleurs TLS
si algorithme de seuillage TLS

.pour chaque couleur j comprise entre ($(c_1 - e)$ et $(c_1 + e)$)

et ($(c_2 - e)$ et $(c_2 + e)$) et ($(c_3 - e)$ et $(c_3 + e)$)

PROTO_VALID[j] = _vrai

fin pour

fin action <<INIT_SEUILLAGE>>

.action <<TEST (pixel)>>

.indice = pixel RVB 12 bits ou pixel TLS suivant le modèle de
couleurs (type d'algorithme)

.si (PROTO_VALID[indice])

alors

.retourner(pixel) << pixel valide >>

sinon

.retourner (pixel noir) << pixel invisible>>

fin si

fin action <<TEST>>

5.1.2.2. Seuillage par rapport aux composantes RVB 24 bits

Les fonctions sont similaires à celles expliquées précédemment, à ceci-près que nous ne travaillons plus sur les couleurs mais sur chacune des composantes. Nous avons 3 tableaux TAB_VALID_ROUGE, TAB_VALID_VERT et TAB_VALID_BLEU de 256 éléments chacun qui jouent le rôle du tableau PROTO_VALID, dans INIT_SEUILLAGE et TEST.

```
.action <<TEST (pixel(r,v,b))>>

    .si ((TAB_VALID_ROUGE[r] ) et
        (TAB_VALID_VERT[v] ) et
        (TAB_VALID_BLEU[v] )
        alors
            .retourner(pixel) << pixel valide >>

        sinon
            .retourner (pixel noir) << pixel invisible>>

    fin si

fin action <<TEST>>
```

5.1.2.3. Seuillage RVB 24 bits

Nous travaillons directement sur les prototypes pour tester la validité du pixel. La phase d'initialisation n'existe pas pour ce traitement.

```

.action <<TEST (pixel(r,v,b))>>

.val = _faux
.i = 0

.tantque (i ne NB_PROTO)
  faire
    .récupération des composantes (rp,vp,bp) du prototype
    TAB_PROTO[i]

    .si ( (( r >= (rp - e) ) et ( r <= (rp + e) )
          et (( v >= (vp - e) ) et ( v <= (vp + e) )
          et (( b >= (bp - e) ) et ( b <= (bp + e) ) )
      alors
        .val = _vrai << pixel valide >>
        .i = NB_PROTO

      sinon
        .i = i + 1 << passage au prochain prototype>>

    fin si

  fin tantque

.si (val )
  alors
    .retourner(pixel) << pixel valide >>
  sinon
    .retourner (pixel noir) << pixel invisible>>
  fin si

fin action <<TEST>>

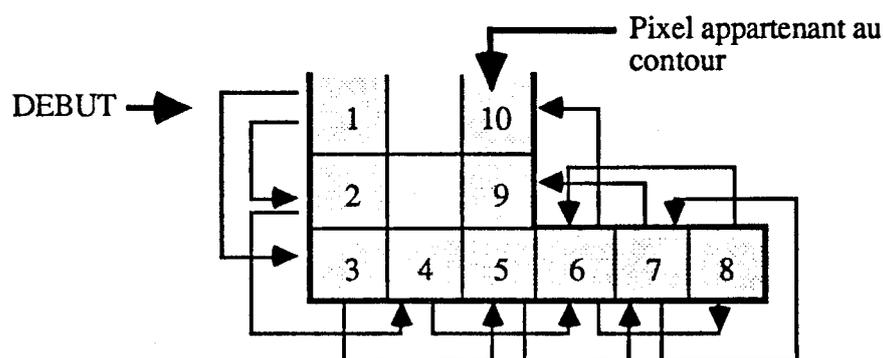
```

5.2. Fonctions relatives au détournage

5.2.1. Principe de l'algorithme de ROSENFELD et KAK

Nous avons retenu l'algorithme de ROSENFELD et KAK (ICA87) pour l'extraction des contours (cf autres techniques du suivi de contours). Nous tournons dans le sens trigonométrique, autour du point courant, à partir du précédent point du contour.

Le schéma suivant montre son fonctionnement. Un pixel est dit "appartenant à l'objet", si sa couleur est différente de la couleur invisible (noire). Le contour est marqué par l'écriture d'un pixel blanc.



L'ALGORITHME DE ROSENFELD ET KAK

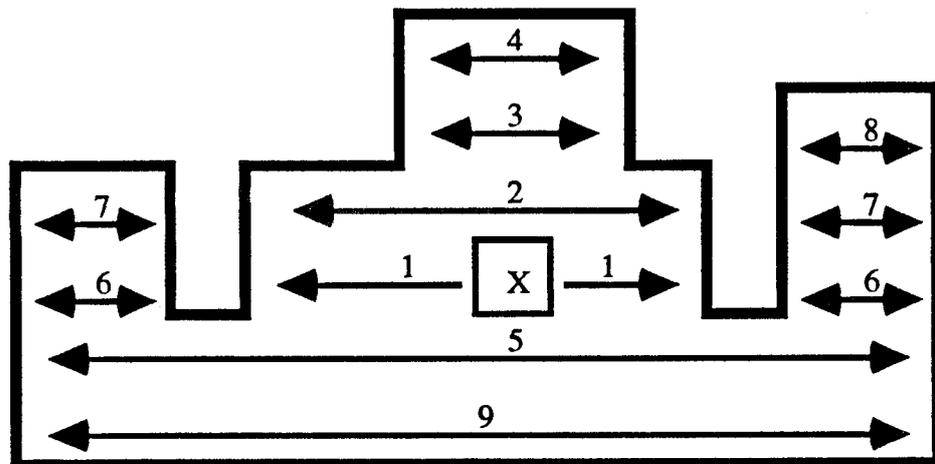
5.2.2. Algorithme de tâche

A partir du contour délimité de cette façon, il ne reste plus qu'à récupérer le contenu de l'objet. Nous utilisons la fonction d'analyse de tâche programmée dans le système ATALIS.

Cette fonction est très performante, car elle minimise les niveaux de récursivité et elle optimise les lectures. Elle s'exécute à partir d'un pixel appartenant à l'objet. Sans entrer dans les détails, le schéma suivant explique son déroulement.

Nous examinons chaque ligne jusqu'à la rencontre des contours de l'objet. L'examen se fait toujours vers le haut si cela est possible (procédure séquentielle), dans le cas contraire il reprend vers le bas, à partir du plus grand numéro de ligne mémorisée non traitée (récursivité).

Pour chaque pixel traité, l'algorithme de tâche appelle une fonction utilisateur TEST, qui teste si le pixel appartient ou non au contour de l'objet. Dans l'affirmative, l'examen de la ligne dans ce sens est arrêté.



ALGORITHME D'ANALYSE DE TACHE

```

.action <<TEST (pixel)>>
    .si (pixel est un pixel blanc)
    alors
        .objet = _faux <<le pixel traité appartient au contour>>
    sinon
        .objet = _vrai <<le pixel appartient à l'objet>>
    fin si

    .retourner (objet)
fin action <<TEST>>

```

L'algorithme de tâche élabore un tampon qui mémorise les différentes coordonnées de l'objet de la façon suivante :

```

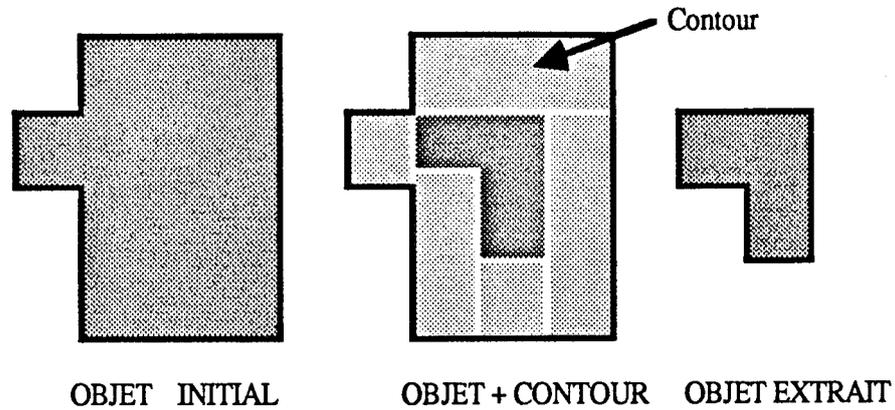
.coordonnées X,Y du premier pixel de la ligne;
.nombre de pixels sur cette ligne

```

L'exploitation de ce tampon combinée à la lecture de l'image initiale, permet d'afficher soit l'objet, soit son complémentaire.

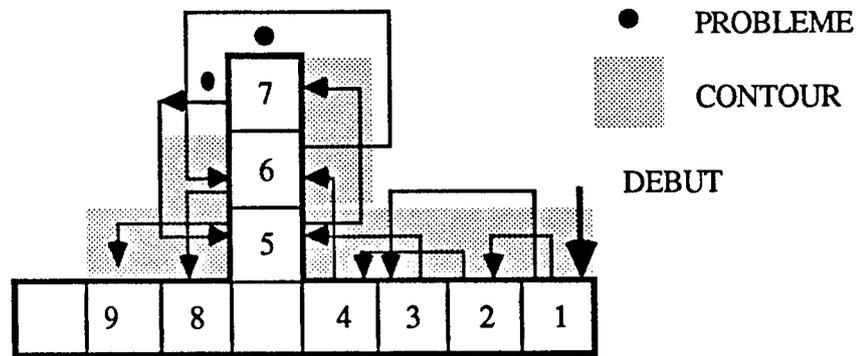
5.2.3. Adaptation de l'algorithme de ROSENFELD et KAK

L'inconvénient est que l'algorithme de contour ne convient pas tout à fait avec celui d'analyse de tâche. Nous obtenons bien l'objet mais privé de ses contours.

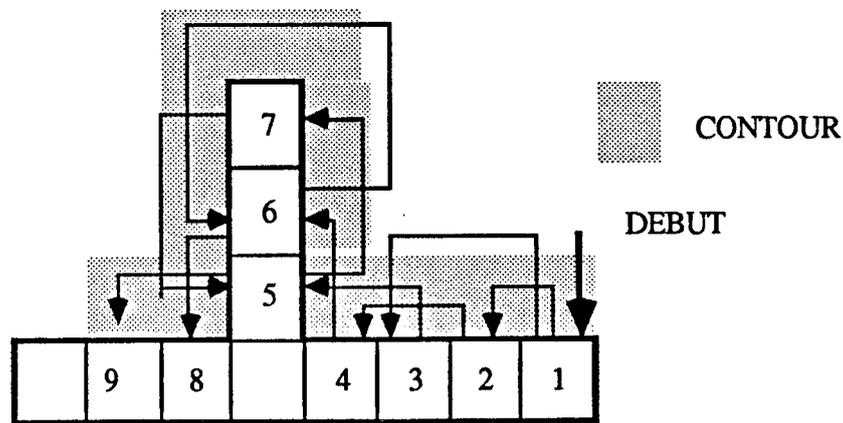


EXEMPLE DE PROBLEMES D'EXTRACTION

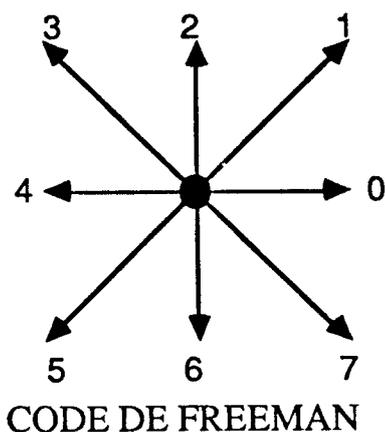
Nous gardons le même algorithme, à la différence près, que nous marquons de la couleur blanche, non pas le pixel contour mais le pixel qui précède sa détection, dans l'algorithme de recherche. Nous obtenons les contours externes. Hélas, certains objets posent encore quelques problèmes !



Il faut pendant la recherche du prochain pixel appartenant au contour, marquer tous les points rencontrés n'appartenant pas à l'objet, comme faisant partie du contour.



Nous lisons le voisinage du pixel courant du contour et nous rangeons ces pixels dans le tableau TAB_PIXEL en fonction du codage de FREEMAN. Dans l'algorithme de recherche, le passage au prochain pixel testé, correspond à l'incréméntation modulo 8 de l'indice du tableau TAB_PIXEL. Pour inverser le sens du détournage, il suffit de remplacer l'incréméntation par une décréméntation.



Si le prochain pixel P du contour a le numéro x dans le codage de FREEMAN par rapport au pixel courant, l'algorithme de recherche commence en prenant comme premier point traité le pixel P' du voisinage P ayant comme numéro $(x + 4) \text{ modulo } 8$.

Le pixel courant du contour P a comme coordonnées X_CONTOUR et Y_CONTOUR, PENTE_CONTOUR mémorise son numéro.

.action <<POURSUITE_CONTOUR >>

.lecture du voisinage de P avec rangement suivant le code de FREEMAN
dans tab_pixel

.inc = 1

.si (sens de détournage = sens des aiguilles d'une montre)

alors

 inc = -1

fin si

.<<détermination de la pente i du premier point testé>>

i = (PENTE_CONTOUR + inc) modulo 8

trouve = _faux

tantque (non trouve)

faire

 .détermination du pixel P' testé en fonction de la pente i

 .si (pixel P' est un pixel noir)

alors

 .<< le pixel appartient au contour>>

 écriture d'un pixel blanc à ce point

 i = i + 1

sinon

 .<< le pixel est interne à l'objet>>

 trouve = _vrai

fin si

fin tantque

.mémorisation des coordonnées du nouvel pixel courant en fonction de la
pente i dans X_CONTOUR et Y_CONTOUR

.mémorisation de la nouvelle pente

PENTE_CONTOUR = (i + 4) modulo 8

fin action <<POURSUITE_CONTOUR>>

5.3. Conversion RVB <==> TLS

5.3.1. Formules de conversions

Les formules de passage du modèle RVB en TLS sont les suivantes (MAR84):

$$.T = \arcsin(\sqrt{3}/2 * (V - R) / S)$$

$$.L = (R + V + B) / 3$$

$$.S = \sqrt{R*R + V*V + B*B - B*V - B*R - V*R}$$

Les formules de passage du modèle TLS en RVB donnent :

$$.R = L - 1/3*S*\cos(T) - 1/\sqrt{3}*S*\sin(T)$$

$$.V = L - 1/3*S*\cos(T) + 1/\sqrt{3}*S*\sin(T)$$

$$.B = L + 2/3*S*\cos(T)$$

Ces formules de conversion sont pénalisantes. Il ne reste plus qu'à les optimiser!

5.3.1. Optimisation RVB => TLS

La solution consiste à "précalculer" ces conversions. Un tableau de structures donne pour chaque couleur RVB sa correspondance en TLS. Comme il est évidemment impensable d'avoir un tableau à 16 millions d'éléments, nous nous limitons à 4096 couleurs, c'est à dire à une image en 12 bits.

L'indice de l'élément est calculé de la façon suivante en considérant les composantes RVB comprises entre 0 et 15 (poids forts) :

$$.indice = ((R * 16) + (V * 16)) + B$$

Ce tableau est initialisé au début de l'application.

5.3.2. Optimisation TLS => RVB

La solution est à peu près identique. Nous calculons pour les 255 valeurs de teintes possibles, à l'initialisation des formules suivantes et nous les rangeons respectivement dans les tableaux COST et SINT :

$$.1/3*\cos(T)$$

$$.1/\sqrt{3}*\sin(T)$$

Il ne reste plus lors d'une demande de conversion, qu'à réaliser les opérations élémentaires suivantes qui sont rapides :

$$\begin{aligned} .\text{COS} &= S * \text{COST}(T) \\ .\text{SIN} &= S * \text{SIN}(T) \\ .R &= L - \text{COS} - \text{SIN} \\ .V &= L - \text{COS} + \text{SIN} \\ .B &= L + 2 \text{COS} \end{aligned}$$

5.3.3. Gain en temps de réponse

Les opérations trigonométriques sont des opérations coûteuses par rapport à une recherche d'éléments dans un tableau.

Sur le compatible PC défini en annexe, le calcul d'une fonction cosinus demande environ 760 μs alors que l'extraction d'un élément du tableau ne nécessite que 20 μs .

5.4. Autres fonctions

Il est impossible dans ce document de parler de toutes les fonctions conçues pour la réalisation du logiciel G_EXTRA (plus de 172 !) et encore moins de celles qui ont été implémentées pour l'étude théorique (variance, différence d'images, suivi de contours par opérateur gradient ...).

Nous avons essayé de montrer dans ce chapitre les fonctions les plus importantes de l'interface homme-machine, car elles constituent une part primordiale de la réalisation et de l'efficacité de ce logiciel.

Nous avons parlé des fonctions spécifiques du traitement d'images qui sont le seuillage et la détection de contours et bien sûr l'objet de ce mémoire !

Malheureusement, d'autres fonctions bien qu'intéressantes sont passées sous silence :

- .fonction de retouche avec l'utilisation d'un algorithme de BRESENHAM;
- .fonction d'effacement de contours

CONCLUSION

Cette étude s'est déroulée en plusieurs phases aussi difficiles et inconnues les unes que les autres, mais toutes aussi intéressantes.

Dans un premier temps, nous avons été amenés à nous initier au traitement d'images. S'il est vrai qu'il existe une importante bibliographie consacrée à ce sujet, il n'est pas du tout évident, loin s'en faut, d'en faire une synthèse et d'en retirer des techniques simples mises en œuvre dans le cadre de notre étude. Du reste, il est souvent nécessaire de les expérimenter pour connaître leurs performances et ainsi leurs limites.

Dans le traitement d'images, nous avons retenu trois grandes étapes :

- .prétraitement avec l'égalisation d'histogramme;
- .saisie de prototypes avec les algorithmes de seuillage de couleurs;
- .détection de contours.

Les autres techniques, décrites dans ce rapport, n'ont pas été implémentées dans ce logiciel (temps de réalisation). Bien qu'intéressantes, (nous pensons en particulier à la segmentation combinant le seuillage relatif aux couleurs et celui relatif à l'aspect), il est possible, à moindre coût, de les introduire dans les versions futures de cette application.

Il est apparu que l'analyse de texture est très importante. Cette dernière technique, sauf quelques courts essais (variance, gradient, différence d'images), n'a été abordée que sur le plan théorique. C'est dans cet axe, mais avec des moyens plus importants que ceux d'un mémoire C.N.A.M., que se situent les principales améliorations de ce logiciel. C'est un domaine en plein essor, mais pour l'instant très difficilement exploitable par la synthèse d'images, vu sa complexité.

L'algorithme du seuillage RVB en 24 bits, excellent par ses résultats, est étroitement lié au nombre de prototypes saisis (cf annexe). Il est souhaitable dans une future version d'améliorer sa vitesse d'exécution, par exemple :

- .en effectuant un tri sur le tableau de prototypes saisis;
- .en comparant le pixel courant avec le pixel précédemment traité...

Nous avons vu aussi que les résultats du traitement d'images sont fonction de l'image initiale. Malheureusement avec une image de qualité médiocre, il s'avère souvent difficile d'extraire correctement un objet. Il n'y a pas de miracle en infographie!

Dans la seconde phase, nous avons été confrontés aux réalités d'une interface homme-machine, digne de ce nom. Sa conception montre qu'il s'agit d'un **métier à part entière**. Il faut beaucoup de réflexions, de concertations et de remises en question pour en dresser les bases. Nous pensons que le logiciel G-EXTRA répond aux critères d'une bonne interface, malgré quelques imperfections dues à sa jeunesse. D'autres fonctions, comme par exemple l'acquisition vidéo d'une image ou des fonctions "clavier" supplémentaires, peuvent être amenées à étoffer *facilement* ce logiciel. Il est également souhaitable d'autoriser les traitements sur des zones autre que des rectangles (*rapidité et souplesse*).

La dernière phase a été la réalisation proprement dite. Bien que la plus facile des étapes, il a fallu néanmoins apprendre à utiliser au mieux les bibliothèques graphiques du système ATALIS.

En résumé, nous pouvons dire que le logiciel G-EXTRA, sans être un logiciel parfait, a répondu au sujet demandé, c'est à dire à l'extraction d'objets dans un contexte audiovisuel. Il est vrai qu'il n'a pas les mêmes prétentions que des logiciels tel que MULTISCOPE, qui a demandé pas moins de 10 hommes-année d'efforts, ou d'autres comme SATIS qui a bénéficié de la coopération du C.N.E.T.

La réalisation de ce mémoire a eu d'autres effets que la création du logiciel G-EXTRA. Il a permis de nous remettre en question sur un sujet complètement inconnu, il y a à peine un an. La distance qui sépare la commande d'un disjoncteur à l'affichage d'un pixel sur un moniteur vidéo, est grande.

De même, venant d'une société qui comporte plusieurs centaines de personnes, il est très enrichissant de comparer les méthodes de travail à celles d'une société plus modeste, mais tout aussi dynamique.

ANNEXES

Annexe 1 : Caractéristiques du PC Compaq 386S

- .Processeur principal 80386SX INTEL à 16 MHz
- .Coprocesseur 80387SX INTEL à 16 MHz
- .Indice NORTON relatif à IBM/XT 15,6

Annexe 2 : Caractéristiques de l'application G-EXTRA

- .Taille de l'exécutable : 287 Koctets
- .Fonctions spécifiques :
 - .taille = 61 Koctets de code
32 Koctets de RAM
 - . 192 fonctions dans 23 fichiers C (378 Koctets de source)
 - . 71 vues et 51 icônes définies pour l'interface homme-machine

Annexe 3 . Comparaison des vitesses de traitement des algorithmes d'extraction

Les essais ont été réalisés à partir d'une même image (580 x 570 pixels), en faisant varier le nombre de prototypes saisis et l'algorithme d'extraction. Le résultat indiqué correspond au temps de traitement, exprimé en secondes.

a) 1 prototype saisi

.seuillage TLS	= 35 s	
.seuillage RVB en 12 bits	= 32 s	
.seuillage en fonction des composantes RVB en 24 bits	= 33 s	
.seuillage RVB 24 bits	= 33 s	

b) modification du nombre de prototypes saisis

.les <i>temps</i> de traitement des seuillages TLS, RVB 12 bits et COM 24 bits sont <i>inchangés quelque soit ce nombre</i>		
.seuillage RVB 24 bits avec 17 prototypes	= 81 s	
.seuillage RVB 24 bits avec 394 prototypes	= 474 s !!!	

Annexe 4 : Photographies d'écran

Cette annexe présente différentes photographies d'écrans réalisées à partir du logiciel G-EXTRA :

.la photographie numéro 1 correspond aux principaux menus de l'application G-EXTRA:

- .menu principal ;
- .menu relatif à l'opération de copie ;
- .menu relatif à l'opération d'extraction par seuillage ;
- .menu relatif à l'opération d'extraction par détournage (suivi de contours).

.la photographie numéro 2 est un exemple d'extraction par seuillage. Cette opération suffit pour extraire la forêt en avant-plan. Nous remarquons l'icône caractéristique (un poisson) pour la saisie des prototypes.

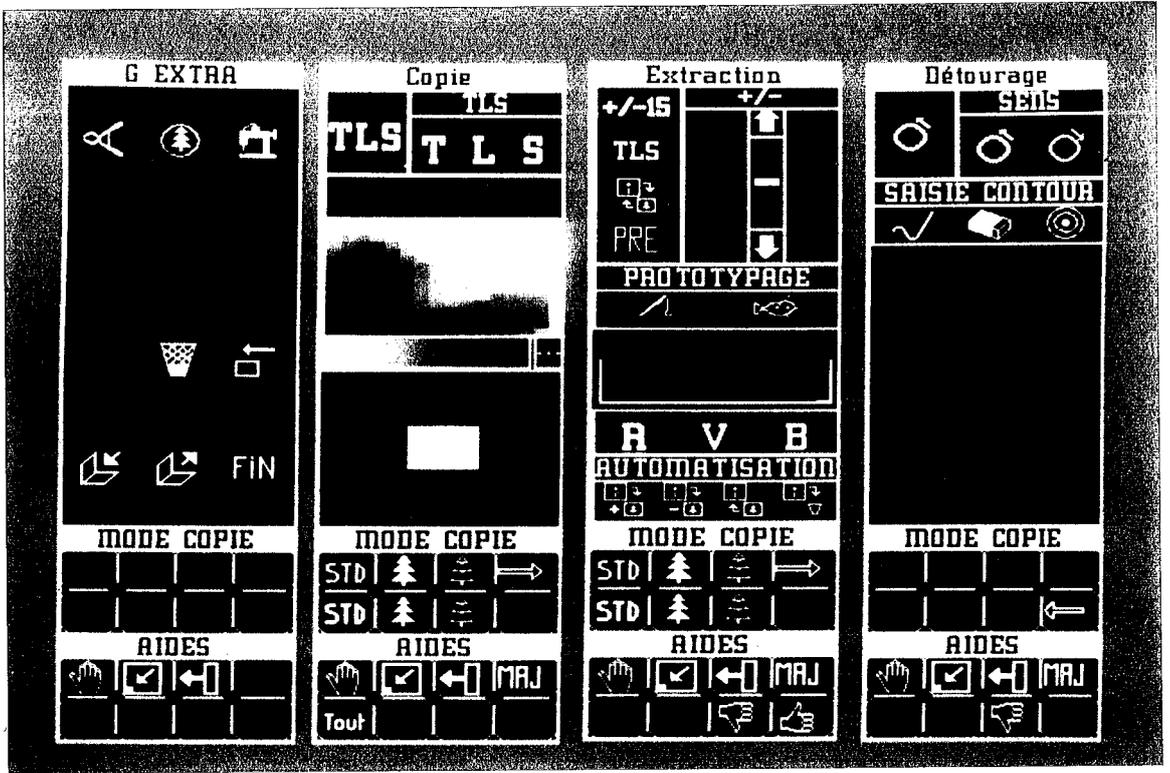
.la photographie numéro 3 présente l'état de l'écran lors d'une opération de détournage. Les contours de la montagne ont été extraits par une opération de seuillage. Le tracé de la montagne est matérialisé par une ligne blanche.

.les photographies suivantes sont des exemples (non exhaustifs) d'utilisation du logiciel G-EXTRA :

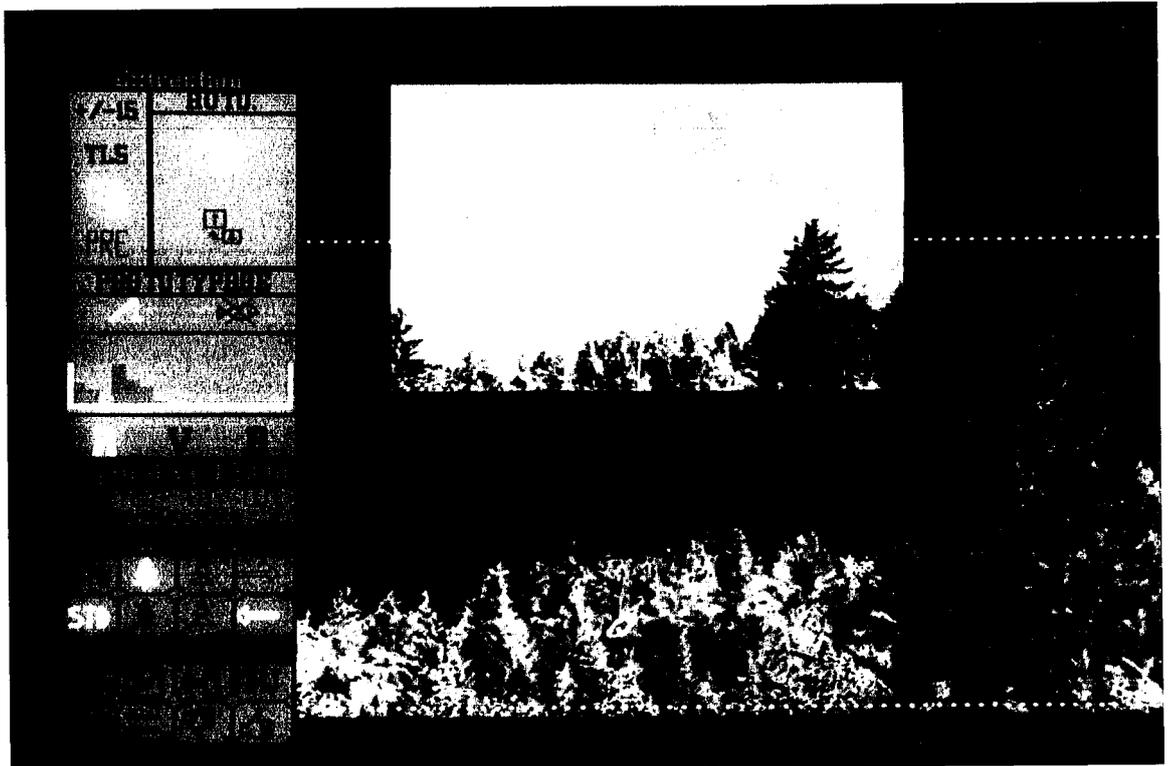
.la photographie numéro 4 est un exemple de création de décor à partir de deux images. La forêt et la montagne appartiennent à la première image tandis que les autres éléments du décor sont issus de la seconde. Nous notons la modification de teinte de la forêt (la luminance et la saturation sont conservées).

.la photographie numéro 5 est l'image initiale que l'on retrouve modifiée sur la photographie numéro 6. Avec la photographie numéro 7, elles peuvent servir à l'élaboration d'un catalogue vestimentaire (adaptation de couleurs).

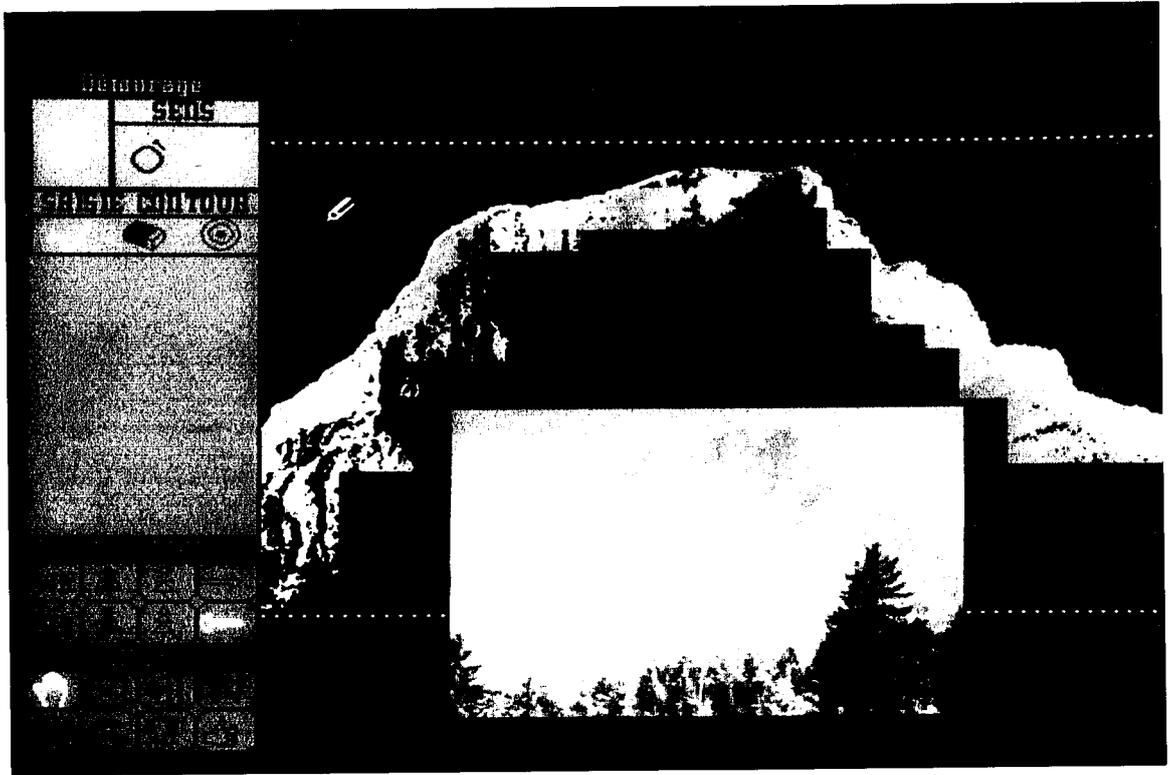
.la photographie numéro 8 montre la constitution d'une banque de données ("imageur") relative aux animaux.



Photographie numéro 1 : principaux menus du logiciel G-EXTRA



Photographie numéro 2 : Exemple d'extraction par opération de seuillage



Photographie numéro 3 : Exemple d'extraction par suivi de contours



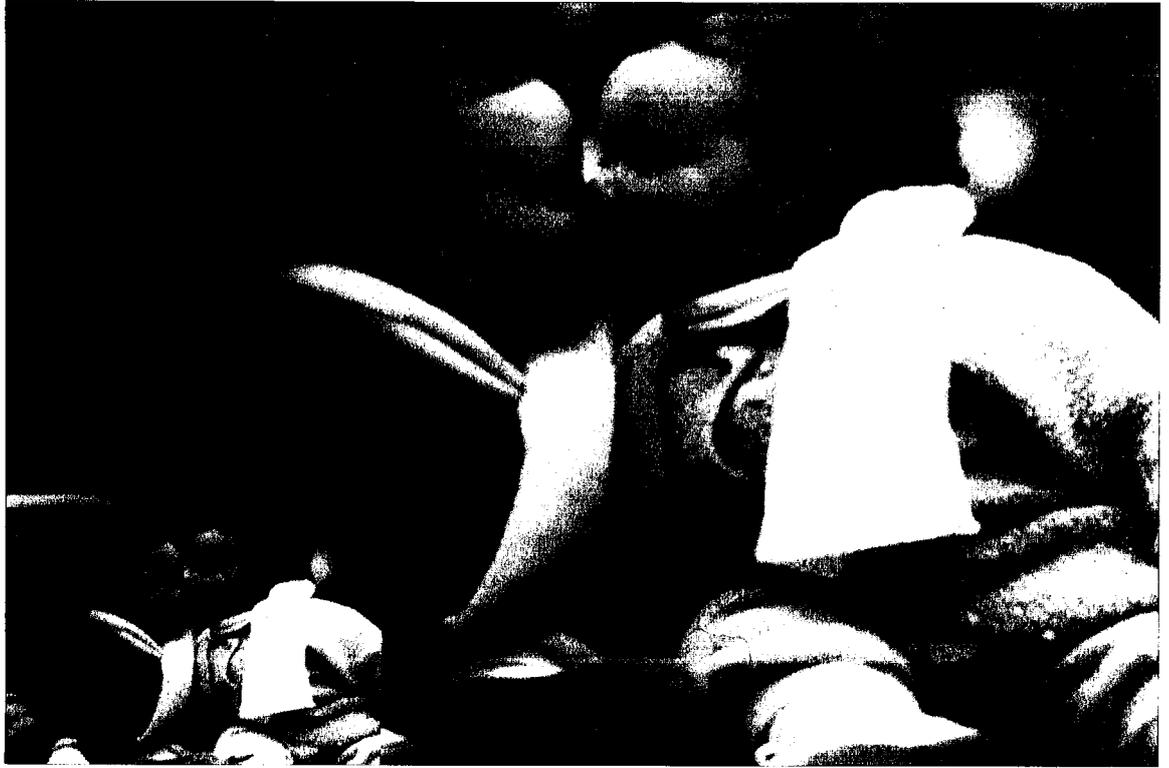
Photographie numéro 4 : Exemple de création de décor



Photographie numéro 5 : Image initiale non traitée



Photographie numéro 6 : Adaptation de l'image numéro 5



Photographie numéro 7 : Exemple de traitement d'images



Photographie numéro 8 : Constitution d'une banque d'images animales

- [GAG85] GAGALOWICZ A., MA S.D. et TOURNIER-LASSERVE.
"Nouveaux modèles pour des textures homogènes et inhomogènes"
Troisième congrès de l'AFCT de NANCY, septembre 1981,
pp 667-678.
- [GAN88] *Manuel d'utilisation G.ANIM*, logiciel d'animation en temps réel
MEYLAN, version 2.1 , GETRIS IMAGES SA, août 1988.
- [GDI89] *Manuel d'utilisation G.DIAL*, bibliothèque graphique de dialogue
MEYLAN, version 1.2. , GETRIS IMAGES SA, janvier 1989.
- [GLI89] *Manuel d'utilisation G.LIB*, bibliothèque graphique de dialogue
MEYLAN, version 1.2. , GETRIS IMAGES SA, janvier 1989.
- [HAR85] HARALICK R.M et SHAPINO L.G.
"Survey Image Segmentation Techniques".
Computer Vision, Graphics and Image Processing, volume 29,
1, janvier 1985, pp 100-132.
- [HOR81] HORAUD P.R. et COULON P.Y. "Un algorithme pour détecter
des objets à forme complexe dans une image", *troisième congrès de
l'AFCT de Nancy*, septembre 1981, pp 213-225.
- [ICA87] *Cours sur les systèmes de vision par ordinateur de ICAP*
(Instrumentation et capteur), GRENOBLE, société anonyme située
sur l'aire de transfert de technologie (ASTEC) du polygone
scientifique de GRENOBLE, 1987.
- [MAR82] MARTINEZ F. "*Vers une approche systématique de la synthèse
d'image. Aspects logiciels et matériels*", GRENOBLE, thèse d'état,
novembre 1982.
- [MAR84] MARTINEZ F. "*La synthèse d'image , concepts, matériels et
logiciels*", EDITESTS, 1984.
- [MOL88] MOLLIE G., "*Etude des problèmes de représentation et de
composition des couleurs appliquée au domaine du dessin par
ordinateur*", Rapport de DEA d'informatique laboratoire
ARTEMIS GRENOBLE, juin 1988.
- [MUL88] MULTISCOPE version 1.1 et 1.2, logiciel de traitement d'images
satellites, "*Description des fonctionnalités*", TOULOUSE, CAP
SOGETI SYSTEMES TOULOUSE
- [PER88] PEROCHE B, ARGENCE J., GHAZANFARPOUR D. et
MICHELUCCI D., "*La synthèse d'images*", HERMES, 1988.

Jean-Yves DOUCET. Utilisation d'images réelles pour la production d'images synthétiques dans un contexte audiovisuel.

Mémoire d'ingénieur C.N.A.M. Grenoble 1989

Résumé : Le travail consiste à concevoir et à réaliser un système permettant d'utiliser des éléments extraits d'images réelles numérisées en couleur, pour faciliter et améliorer la production d'images synthétiques statiques ou dynamiques.

La réalisation de ce travail débouche sur une application entièrement interactive, à partir d'une station de synthèse ATALIS de la société GETRIS IMAGES.

L'infographie est divisée en deux parties :

- .la synthèse d'images;
- .le traitement d'images.

L'idée de cette étude est d'utiliser les principes de base du traitement d'images pour compléter les possibilités graphiques de la première.

Nous décrivons dans la partie étude théorique, les différents modèles de couleur (RVB, TLS...). Nous montrons les applications du traitement d'images et les différences de contextes par rapport à notre problème. Nous explicitons les différentes techniques qui invariablement se décomposent en trois phases :

- .prétraitement;
- .segmentation;
- .post-traitement.

Nous définissons entre autres, les termes d'histogramme, d'égalisation, d'apprentissage, de prototypes, de variance, de gradient, de filtres et de texture.

A la fin de cette étude, nous esquissons notre solution. Elle utilise la technique du suivi de contours sur une image préalablement segmentée. La segmentation s'effectue par rapport à un ensemble de prototypes qui admet une tolérance.

Dans la partie réalisation, nous montrons comment les contraintes de l'interface homme-machine ont été résolues. Nous y détaillons les algorithmes de segmentation et de suivi de contours.

Mots clés : Traitement d'images - Segmentation - Détourage - Suivi de contours - Seuil - Texture - Interface - Image couleur

Keywords : Image Processing - Segmentation - Cut-out - Contour analysis - Threshold - Texture - Interface - Picture colour

BIBLIOGRAPHIE

- [ART87] Laboratoire ARTEMIS, "*Etude et développement d'un système de production et de consultation interactive d'images*", rapport final du contrat ADI numéro 84/819, GRENOBLE, Institut National Polytechnique de GRENOBLE, 1987.
- [ATA88] *Manuel d'installation ATALIS version 2.2.*, MEYLAN, GETRIS IMAGES SA, septembre 1988.
- [BOU85] BOUBEKRAOUI A. et ZAVIDOVIQUE B. "Une méthode d'analyse des régions texturées par des mesures locales "contradictoires"", *cinquième congrès de l'AFCEP sur la reconnaissance des formes et intelligence artificielle*, Grenoble novembre 1985, pp 689-701.
- [BRA88] BRANCIER C., TRIQUET M., "SIGGRAPH' 88. De l'art à l'industrie", *DECISION*, 196, 5 septembre 1988, pp 27-36.
- [CAN81] CANNON T. et HUNT B.
"Le traitement des images par ordinateur".
Pour la Science, 50, décembre 1981, pp 88-98.
- [CHA84] CHARROUF KHALID, "Projet PICASSO, Peinture interactive en couleur assistée par ordinateur", *Premier colloque IMAGE Traitement, Synthèse, Technologie et Applications*, BIARRITZ, mai 1984, pp 177-181.
- [CHA86] CHAIX J.M. "*Une introduction au traitement et à l'analyse d'images*", GRENOBLE, Laboratoire de Thermodynamique et physico-chimie métallurgiques Institut National polytechnique de GRENOBLE, septembre 1986.
- [CHE85] CHEKROUM D. "La vision artificielle, Algorithmes de traitement d'image et architectures". *LE NOUVEL AUTOMATISME*, janvier-février 1985, pp 45-51.
- [DAL87] DALLOZ X. "Le marché et les applications de la synthèse d'images", *Minis et micros*, 273, février 1987, pp 15-19.
- [DUD73] DUDA R.O. et HART P.E. "*Pattern classification and scene analysis*", NeW-YORK, J.WILEY, 1973.
- [FOU88] FOUCHARD G. "Animation temps réel : la solution GETRIS", *MICRO-SYSTEME*, 89, Septembre 1988, pp 183-186.

- [POS87] POSTAIRE J.G. "*Analyse des images numériques et théorie de la décision*", PARIS, DUNOD Informatique, 1987.
- [PRE82] PREVOT C. "Reconnaissance artificielle des formes sur des méthodes de détection des contours", *Le Nouvel Automatismes* mars 1982, pp 47-53.
- [REI86] REINHARD H. "*Cours de mathématiques du signal*", PARIS, DUNOD Université, 1986.
- [ROC75] ROCK I. "*Introduction to perception*", MACMILLAN, NEW-YORK, 1975.
- [SEG88] SEGURA J. "Image de synthèse : du professionnel sur PC", *PC Informatique*, 46, 1 septembre 1988.
- [SEY87] SEYDEN E. "Infographie : de la CFAO au graphique de gestion", *Temps réel*, 2 février 1984, pp 29-33.
- [THO88] THORPE C., HEBERT M.H, KANADE T. et SHAFER S.A. "Vision and navigation for the Carnegie-Mellon NAVLAB", *IEEE Transaction on pattern analysis and machine intelligence*, volume 10, 3, mai 1988, pp 362-373.