



UNIVERSITÉ DE CAEN
U.F.R. DE SCIENCES
DÉPARTEMENT D'INFORMATIQUE

Vinh Thong TA

*Rapport de stage de D.E.A.
septembre 2004*

REPRÉSENTATION D'IMAGES PAR ARBRES BINAIRES DE RÉGIONS

Laboratoire d'accueil :

Laboratoire Universitaire des Sciences Appliquées de Cherbourg
groupe Vision et Analyse d'Images

Encadrants :

M. Olivier LÉZORAY
M. Abderrahim ELMOATAZ

*D.E.A. d'Intelligence Artificielle et Algorithmique
Option Image
Année 2003 - 2004*

REPRÉSENTATION D'IMAGES PAR ARBRES BINAIRES DE RÉGIONS

Vinh Thong TA - *vta@etu.info.unicaen.fr*
septembre 2004

Rapport de stage de D.E.A. réalisé au cours du
D.E.A. d'Intelligence Artificielle et Algorithmique option Image.

Année 2003 - 2004.

Résumé :

Dans le contexte de l'indexation et de la recherche des images dans de grandes bases de données ; une image doit avoir une représentation autre que l'image brute afin de pouvoir exploiter certaines de ses informations pertinentes.

Dans cette optique, le but de cette étude consiste à développer une représentation des images couleur basée sur le formalisme des arbres et à explorer des pistes permettant la génération et l'obtention d'une représentation arborescente optimale.

Deux approches se dégagent : l'une est dite «descendante» et l'autre «ascendante».

La première approche consiste à partir de la racine de l'arbre (l'image complète), pour arriver aux feuilles (l'image segmentée) grâce à une stratégie de «division/génération». La division étant déterminée à partir d'un critère d'uniformité sur les régions étudiées .

La seconde approche consiste à partir des feuilles pour arriver vers la racine de l'arbre. le processus démarre avec une image segmentée (par une méthode quelconque) et grâce à une stratégie de «fusion/génération» nous obtenons la représentation arborescente désirée. Un élagage peut être réalisé afin d'obtenir un arbre optimal.

Cette étude permet de mettre en évidence certains problèmes ainsi que des pistes et des solutions dans la mise en œuvre d'une telle représentation dans le domaine de l'imagerie.

Mots clés :

Images ; arbres binaires ; régions.

Je tiens à remercier M. Olivier LÉZORAY pour son suivi durant ce stage et ces nombreux conseils qui m'ont permis d'avancer et de réaliser ce travail.

Une pensée particulière pour ma chère et tendre ; merci de son soutien pendant cette période «estivale» sans qui peut être rien n'aurait été possible.

Une dernière pensée pour la fin d'une longue période enrichissante qui aura occupée une partie de ma vie ...

Table des matières

Remerciements	4
Introduction	7
1. Des problèmes aux solutions	8
1.1 Les représentations arborescentes	9
1.2 Méthodes de segmentation	10
1.3 Critères d'évaluation et optimisation	10
1.4 Stratégies retenues	11
2. Stratégie descendante	12
2.1 Segmentation par blocs	13
2.2 Critères de segmentation, d'uniformité	14
2.3 Détermination du seuil optimal	16
2.4 Conclusions, perspectives	22
3. Stratégie ascendante	24
3.1 Segmentation par LPE	25
3.2 Algorithme de «fusion/génération»	27
3.3 Élagage de l'arbre	30
3.4 Conclusions, perspectives	35
Conclusion générale	37
Bibliographie	38

Introduction

Au vue du nombre grandissant de documents visuels et audiovisuels échangés et l'explosion des bases de données d'images numériques, il semble indispensable de pouvoir mettre en œuvre des techniques d'indexation, de recherche et d'échange de ces informations.

Dans ce contexte, l'exploitation des données brutes (les images) ne semble pas un moyen judicieux pour obtenir des résultats performants et pertinents. L'utilisation d'une autre représentation semble nécessaire permettant ainsi une meilleure recherche et classification des images.

Dans cette optique, l'utilisation d'une structures arborescente semble être une des voies possibles aux vues dux applications spécifiques déjà existantes (compression, échange de données, recherche, ...). Ce type de représentation apparaît d'ailleurs dans le standard de description multimédia **MPEG-7**.

Le but de notre étude consiste à mettre en œuvre ce type de représentation d'un point de vue généraliste, de dégager les problématiques et les solutions possibles permettant de développer de telles structures : l'objectif étant de pouvoir construire une représentation arborescente à partir d'une image de tout type.

Notre étude nous a conduit à mettre en évidence deux approches possibles ; l'une «descendante» et l'autre «ascendante» dans la construction des arbres ; ainsi que les différentes problématiques que ces stratégies peuvent engendrées. Pour chacune de ces méthodes, des solutions et des pistes sont proposées.

Ce document sera structuré de la manière suivante : tout d'abord, nous ferons un point sur la littérature concernant les représentations arborescentes, les techniques de segmentation et d'évaluation de la qualité des résultats.

Ensuite, nous exposerons les deux approches mises en œuvre : d'abord descendante, puis ascendante. Pour ces techniques, nous exposerons nes choix ainsi qu'un certain nombre de résultats expérimentaux permettant de juger de la pertinence des stratégies.

Enfin, nous terminerons par exposer les conclusions et les perspectives permettant l'amélioration les résultats ainsi que des pistes qui pourront conduire à une poursuite de ce travail.

Des problèmes . . .

. . . aux solutions

Dans cette partie, nous exposerons les différentes techniques existantes dans la littérature concernant les représentations arborescentes. Elles nous emmèneront vers différentes stratégies et solutions possibles permettant d'atteindre nos objectifs.

Les thématiques seront les suivantes : les structures arborescentes, les techniques de mise en œuvre, les méthodes de segmentation ainsi que les moyens d'évaluation et d'optimisation des différentes approches.

1.1 Les représentations arborescentes

Depuis ces dernières années, les représentations arborescentes sont de plus en plus utilisées dans le domaine de l'image.

Ces dernières ont l'avantage d'offrir une structure de données utile et pratique dans la représentation des images. Elles sont utilisées et conviennent à de très nombreux domaines : indexation, compression, segmentation, reconnaissance de forme, ... [SALEMBIER 00].

Les représentations les plus couramment utilisées sont les **quadtrees** («arbres quaternaires») et les **arbres binaires** («binary partition tree»).

Les quadtrees

Les quadtrees sont généralement utilisés dans les domaines de la compression, du débruitage, du filtrage et de la segmentation ([FTÉRICH 04], [SHUKLA 02], [SHUKLA 04]).

En terme de structure de données, les quadtrees sont des arbres dont les nœuds contiennent soit quatre fils soit une ou plusieurs informations utiles (un ou plusieurs descripteurs) de la région.

Le principe de construction d'un quadtree réside dans un algorithme de division, stratégie que nous pouvons appeler **descendante** : démarrer de la racine (l'image complète) pour arriver jusqu'aux feuilles (les régions segmentées).

La racine de l'arbre contient soit une information relative à l'image entière, soit quatre fils qui correspondent à quatre sous images obtenues par une division horizontale puis une division verticale de l'image.

Chacun des fils peut alors à son tour soit contenir une information relative à sa zone de l'image, soit quatre fils obtenus de la même manière que précédemment.

La décision de division de la zone en quatre fils est obtenue soit par la non satisfaction d'un prédicat (critère d'uniformité quelconque), soit lorsque la zone à diviser atteint une taille donnée.

Les arbres binaires

Les arbres binaires constituent l'autre représentation arborescente communément utilisée dans de nombreux domaines [SALEMBIER 00] (indexation, recherche, segmentation, ...).

En terme de structure de données les arbres binaires sont comme les quadtrees à la nuance près, qu'au lieu de contenir quatre fils, un nœud de l'arbre n'en contient que deux.

Les arbres binaires sont construits à l'aide d'un algorithme de fusion. Nous l'avons appelé stratégie **ascendante** : en partant des feuilles (l'image segmentée), on remonte jusqu'à la racine (l'image complète ne formant plus qu'une seule région) par fusion successive des régions adjacentes grâce à un graphe d'adjacence des régions («Region Adjacency Graph») [BLAFFERT 01].

Afin de déterminer le graphe d'adjacence des régions, une pré segmentation sur l'image originale est faite. Cette segmentation peut être de n'importe quel type, mais la plus communément utilisée est la **LPE** («Lignes de Partage des Eaux»)[SALEMBIER 00]. Cette première segmentation va constituer les feuilles de l'arbre binaire. La fusion deux à deux des régions adjacentes (grâce au graphe) va permettre de générer l'arbre par un calcul de similarité entre les régions. Cette fusion engendre la création d'une nouvelle région revenant à créer un nouveau nœud dans l'arbre.

1.2 Méthodes de segmentation

Qu'il s'agisse de la stratégie ascendante ou descendante, les méthodes de segmentation sont nombreuses.

Nous avons vu que dans le cas de la stratégie descendante, un prédicat était mis en œuvre. Cela peut être un critère concernant le contraste ou la variance des régions. Il constitue donc un critère d'uniformité de la région permettant de décider de la division de celle-ci (constitution ou non de nouveaux fils dans l'arbre).

Dans le cas de la stratégie ascendante, nous avons vu qu'une pré segmentation de l'image était nécessaire afin de constituer le graphe d'adjacence et les feuilles de l'arbre binaire. La méthode de la «ligne de partage des eaux» est une des méthodes possibles.

La Ligne de Partage des Eaux (LPE)

Cette méthode est une méthode classique de segmentation par morphologie mathématique ([SERRA 82], [COSTER 90], [VINCENT 91]). Elle est basée sur une croissance de région à partir de germes obtenus en calculant les minima du gradient de l'image initiale.

Tout d'abord réalisée sur des images en niveaux de gris, elle est ensuite adaptée pour des images couleurs [MEYER 92]. Le problème de cette technique réside dans l'obtention de minima non pertinents qui peuvent être évités par filtrage ou par construction du graphe d'adjacence des régions [BLAFFERT 01].

Afin de comprendre cette méthode, nous pouvons voir l'image comme un «relief». L'altitude du relief est représentée par les niveaux colorimétriques de l'image. L'inondation progressive du relief (à l'aide des minima) permet d'obtenir les lignes de partage des eaux séparant les différents bassins du relief. La segmentation de l'image réside donc dans l'obtention de ces lignes de partage des eaux, c'est-à-dire trouver les lignes le long desquelles les niveaux colorimétriques varient rapidement.

Similarité entre région

La similarité entre régions permet dans la génération de l'arbre binaire, de savoir quelles seront les régions à fusionner en premier : «fusionner deux régions en une équivaut à créer un nouveau nœud ayant deux fils» [SALEMBIER 00], [SALEMBIER 97].

Cette similarité entre régions réside dans la détermination d'une métrique permettant de mesurer la distance entre deux régions. La distance euclidienne est la plus couramment utilisée. Ici la difficulté réside dans la prise en compte de la distribution colorimétrique de l'image : de nombreuses métriques particulières peuvent être utilisées [SALEMBIER 97], [BLAFFERT 01], [SCHETTINI 93]; toutes dépendantes de l'espace couleur et de la méthode de segmentation mis en place.

1.3 Critères d'évaluation et optimisation

Une fois la segmentation des images réalisée, des mesures d'évaluation doivent être faites afin de juger de la qualité des résultats. Les calculs les plus couramment utilisés sont le **PSNR** (Peak Signal to Noise Ratio) et le **MSE** (Mean Square Error) [BORSOTTI 98], permettant de juger des erreurs commises entre segmentation obtenue et image originale. Elles permettent également d'évaluer la qualité des critères d'uniformité mis en place lors d'une stratégie descendante (décision de division d'une région mère en deux régions filles).

Une fois les arbres binaires générés, un élagage peut être réalisé afin d'obtenir un

résultat plus compact (dans le cas d'une compression) ou tout simplement pour améliorer la segmentation.

Généralement, les stratégies d'élagage de l'arbre résident dans un calcul d'énergie au niveau des feuilles de l'arbre : calcul prenant en compte la distortion de la région par rapport à l'image originale ainsi que l'erreur commise au niveau des données [SALEMBIER 00], [FTÉRICH 04], [SHUKLA 02], [SHUKLA 04]. La minimisation de cette énergie permet de prendre la décision d'élaguer l'arbre à un endroit précis. Ce processus est réalisé itérativement jusqu'à obtention d'un résultat jugé optimal pouvant faire intervenir les critères d'évaluation vus précédemment.

1.4 Stratégies retenues

Comme nous l'avons vu, de nombreuses techniques existent quant à l'utilisation et la mise en œuvre des représentations arborescentes dans l'image.

Nous avons choisi d'utiliser les arbres binaires et d'appliquer une stratégie descendante d'une part et ascendante d'autre part.

Pour la stratégie descendante, une méthode de «division/génération» sera mis en place. Pour la stratégie ascendante, une segmentation par LPE et l'utilisation du graphe d'adjacence des régions nous aiderons dans notre démarche de «fusion/génération» de l'arbre binaire. Un élagage de cet arbre sera ensuite réalisé afin d'obtenir une représentation optimale.

En ce qui concerne l'espace couleur utilisé (intervenant dans de nombreux calculs et évaluations), il est indépendant du problème des arbres binaires en tant que tel. Pour des raisons pratiques, nous utiliserons tout au long de nos expérimentations et de nos calculs l'espace couleur **RGB** (Red, Green, Blue).

Stratégie descendante

Dans cette partie, nous exposerons la première approche que nous avons mise en œuvre : la stratégie descendante. Elle consiste à générer l'arbre à partir de la racine. Cette méthode est la méthode de génération des quadrees, mais dans notre cas elle sera appliquée aux arbres binaires.

Tout d'abord, nous expliquerons notre méthode de segmentation et les critères d'uniformité utilisés afin d'assurer le processus de division des régions. Ensuite, nous présenterons les mesures de qualité utilisées afin de déterminer la segmentation «idéale». Enfin, des résultats expérimentaux illustreront nos propos.

2.1 Segmentation par blocs

Comme nous l'avons déjà exposé, la stratégie descendante est en fait un algorithme de division/génération divisant les régions de l'image et générant l'arbre binaire. Cette stratégie est la méthode de segmentation des quadrees [FTÉRICH 04] appliquée ici aux arbres binaires : au lieu de considérer une division horizontale puis verticale (cas des quadrees), nous réalisons une division horizontale ou verticale en fonction des dimensions du bloc étudié.

En terme de structure de données, nous considérons un bloc comme étant un objet à deux dimensions : une longueur et une largeur possédant des caractéristiques colorimétriques correspondant à l'espace couleur utilisé (rappelons que dans notre cas il s'agit de l'espace RGB).

D'un point de vue algorithmique, nous utilisons pour la génération de l'arbre binaire, une structure de données de type file **FIFO** (First In First Out) permettant d'emmagasiner les différents blocs créés au cours de la segmentation.

La décision de division d'un bloc en deux nouveaux blocs se fait par non satisfaction d'un prédicat portant sur les propriétés colorimétriques du bloc étudié à un instant donné. L'arbre binaire est généré simultanément à la division du bloc : la division entraîne la création de deux nouveaux nœuds fils rattachés à leur nœud père (le bloc qui vient d'être divisé).

Classiquement, l'algorithme de division des quadrees s'arrête soit par satisfaction du prédicat, soit lorsque la taille d'un bloc est de la taille d'un pixel. Afin de simplifier cet algorithme et d'éviter d'obtenir trop de petites régions non significatives (dans la segmentation finale), nous fixons une taille minimale à respecter (constante `taille` de l'algorithme). Cette taille varie bien évidemment en fonction du résultat désiré. Nous l'avons fixé à 3 pixels, ce qui correspond classiquement à la taille d'un élément structurant (3×3).

L'algorithme de division/génération peut se résumer de la manière suivante :

```

0  -----
1  pred   : un prédicat d'uniformité du bloc
2  long   : la longueur d'un bloc;
3  larg   : la largeur d'un bloc;
4  taille : taille minimale d'un cote de bloc;
5  liste  : fifo<bloc>;
6  ab     : l'arbre binaire
7  -----
8  début
9  {
10     initialiser liste avec un bloc contenant l'image initiale;
11     créer la racine de l'arbre avec ce bloc;
12     tant que(liste n'est pas vide)
13     {
14         bloc = liste.Pop();
15         si( bloc.long>=taille OU bloc.larg>=taille OU NON(pred) )
16         {
17
18             si( bloc.larg>bloc.long )
19                 diviser horizontalement le bloc;
20             sinon
21                 diviser verticalement le bloc;
22
23         créer deux nouveaux blocs avec les nouvelles dimensions;

```

```

24     créer deux nouveaux noeuds avec les deux nouveaux blocs;
25     mettre à jour ab avec les deux nouveaux noeuds;
26     ajouter les deux nouveaux blocs à liste;
27     }// fin si
28   }//fin tant que
29 }
30 fin

```

La figure 2.1 (de haut en bas et de gauche à droite) montre les premiers pas de l'algorithme de division/génération et la correspondance entre division en bloc de l'image et génération de l'arbre binaire. Nous pouvons remarquer que les feuilles de l'arbre binaire final correspondent aux régions segmentées de l'image.

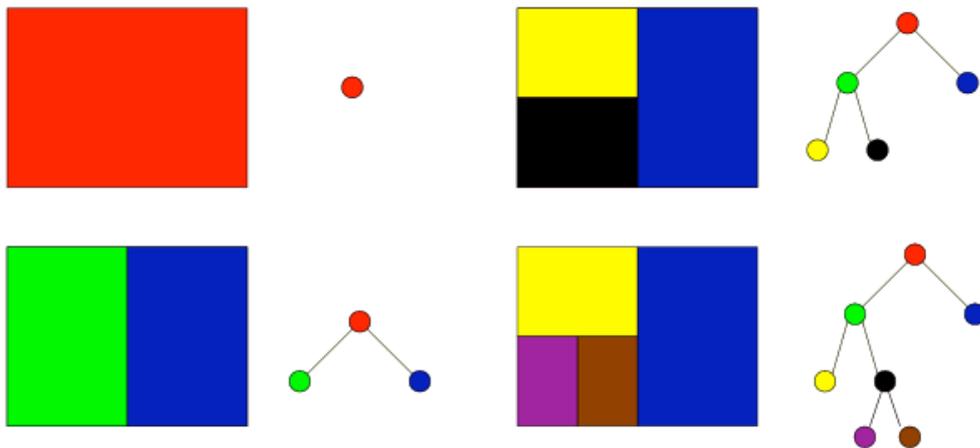


FIG. 2.1 – Quelques pas de l'algorithme division/génération

2.2 Critères de segmentation, d'uniformité

Une fois l'algorithme de division/génération opérationnel, le problème qui se pose désormais est le suivant : *quel prédicat utiliser ?*

Il est évident que la recherche d'un prédicat de segmentation n'est pas un problème facile. Il constitue à lui seul une des problématiques de tout système de segmentation. De ce fait, nous avons décidé de tester deux prédicats classiques et simples :

1. un critère portant sur le contraste.

Le calcul du contraste se fait de la manière suivante :

$$\text{Contraste}(B) = \frac{1}{3} \left(\sum_{i \in X, Y, Z} \left(\max(B_i) - \min(B_i) \right) \right) \quad (2.1)$$

avec :

- B , le bloc à diviser ;
- $\max(B_i)$ et $\min(B_i)$ respectivement, l'intensité la plus grande et la plus petite sur chaque composante X, Y, Z de l'espace couleur, ici RGB.

Le bloc est divisé si $\left(\text{Contraste}(B) > t \mid t \in [0..255] \right)$, t un seuil donné ;

2. un critère portant sur la variance ;

Le calcul de la variance se fait de la manière suivante :

$$Variance(B) = \frac{1}{3} \sum_{i \in X, Y, Z} \left(\frac{1}{n} \sum (B_i(x, y))^2 - \frac{1}{n} \left(\sum B_i(x, y) \right)^2 \right) \quad (2.2)$$

avec :

- B , le bloc à diviser ;
- $\sum B_i(x, y)$, correspond à la somme de toutes les intensités sur les composantes X, Y, Z de l'espace couleur (RGB) de tous les pixels du bloc B ;
- n , le nombre total de pixels du bloc B .

Le bloc est divisé si $(Variance(B) > t \mid t \in [0..255])$, t un seuil donné.

La question qui se pose à nous dorénavant est : *quel critère d'uniformité choisir pour notre algorithme de division/génération ?*

Les figures 2.3 et 2.4 montrent deux exemples d'images segmentées respectivement par le critère d'uniformité portant sur le contraste et par le critère portant sur la variance avec un seuil moyen de valeur égale à 125. Notons que ces cartes de régions représentent toutes les feuilles des arbres binaires générés.



FIG. 2.2 – Images originales

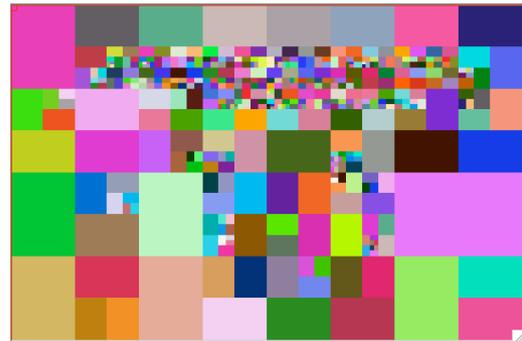
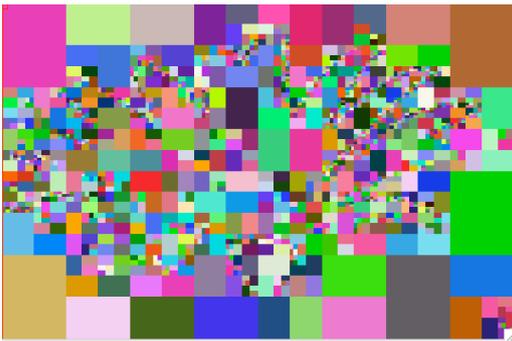


FIG. 2.3 – Cartes des régions segmentées par le critère de contraste

Les cartes des régions montrent bien que le critère utilisant la variance couleur conserve assez bien les formes originales et les détails des images. Les cartes utilisant le prédicat sur le contraste forment des blocs beaucoup plus gros et sont bien moins détaillées. Au vue des résultats, il semblerait légitime d'utiliser le critère sur la variance comme prédicat de segmentation, mais il ne faut pas oublier que le propos de cette étude est de travailler sur la représentation arborescente et non sur la carte des régions.

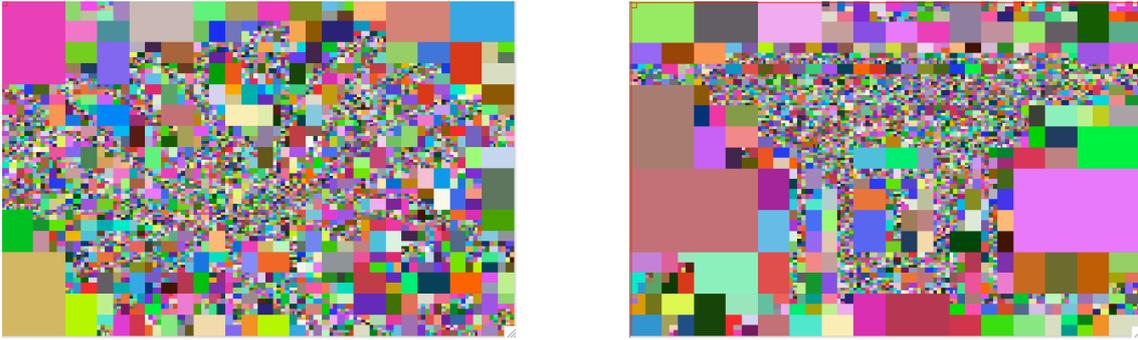


FIG. 2.4 – Cartes des régions segmentées par le critère de variance

De ce fait, il semble judicieux de prendre en compte, en plus du résultat de la segmentation, la complexité de l'arbre généré ainsi que les temps de calculs pour prendre une décision.

image		contraste			variance		
nom	taille (en pixels)	profondeur	nœuds	temps	profondeur	nœuds	temps
fleur	396 × 600	15	2808	6s	15	9960	31s
porte	396 × 600	14	868	3s	15	8024	20s

FIG. 2.5 – Tableau de comparaison des arbres générés par contraste et par variance

Le tableau de la figure 2.5 récapitule certaines données concernant les arbres binaires générés et le temps de calcul. Nous pouvons constater que le nombre de nœuds entre le contraste et la variance varie d'un facteur 3 à un facteur 9 influençant le temps de calcul (temps indicatif dépendant d'autres facteurs).

Afin de simplifier nos recherches et d'après les résultats montrés, nous avons opté pour le critère le plus simple, c'est-à-dire l'utilisation du contraste comme prédicat de segmentation. Les différents résultats qui vont suivre dans les prochaines parties restent valables pour n'importe quel critère.

N.B. : dans la suite de cette partie, tous les exemples que nous montrerons seront des exemples issus d'une segmentation utilisant le prédicat portant sur le contraste.

2.3 Détermination du seuil optimal

Un autre problème non trivial de toute segmentation est la détermination du seuil. Dans notre cas, il s'agit de déterminer le seuil de satisfaction du prédicat de division. Trouver un seuil optimal est le compromis entre avoir une bonne segmentation et une structure d'arbre plus ou moins compacte. En ayant choisi le critère d'uniformité du contraste, la compacité de l'arbre est déjà plus ou moins garantie.

Le Mean Square Error

Un moyen classique de déterminer un seuil optimal est de calculer le MSE entre l'image originale et l'image recolorée. La figure 2.6 montre deux exemples d'images recolorées à l'aide des cartes des régions issues de la segmentation de la figure 2.3.

La formulation du MSE est la suivante :

$$MSE(I, C) = \frac{1}{n} \sum_{(x,y) \in I} \left(\frac{1}{3} \sum_{i \in X,Y,Z} \left(I_i(x,y) - C_i(x,y) \right)^2 \right) \quad (2.3)$$

avec :

- I et C , respectivement l'image originale et l'image recolorée ;
- $I_i(x, y)$ et $C_i(x, y)$, correspondent à l'intensité sur les composantes X, Y, Z de l'espace couleur (RGB) de l'image originale et de l'image recolorée ;
- n , le nombre de pixels de l'image I ou C .



FIG. 2.6 – Images recolorées à l'aide de la carte des régions et l'image originale

La stratégie à mettre en œuvre consiste à calculer le MSE en fonction d'un seuil, seuil variant de 0 à 255. Une fois la courbe obtenue nous espérons trouver soit des plateaux, soit des minima locaux ou un minimum global correspondant au seuil optimal.

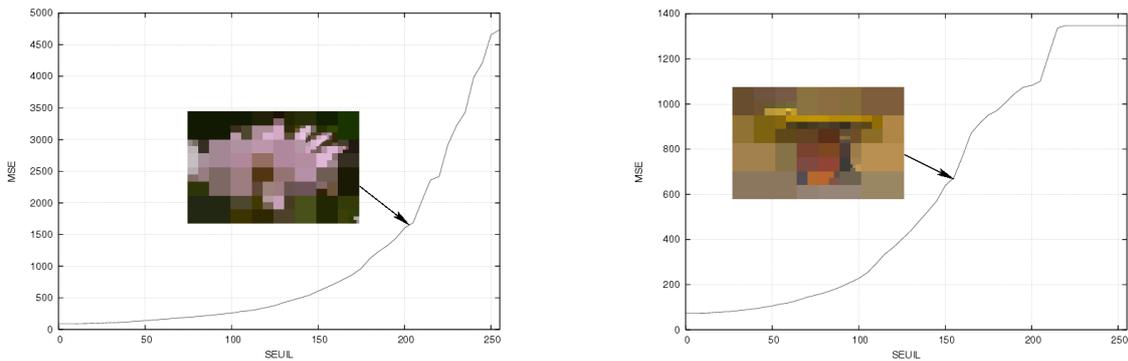


FIG. 2.7 – Courbes du MSE pour l'exemple de la fleur et de la porte

La figure 2.7 montre le calcul du MSE pour l'image de la porte et de la fleur. Comme nous pouvons le voir, la courbe est croissante et ne présente que peu de fluctuations. Les premières fluctuations (légers plateaux) apparaissent pour un seuil à 200 en ce qui concerne la fleur et à 155 pour la porte. Les images reconstituées à partir des cartes de régions (feuilles de l'arbre binaire) ne présentent pas une bonne segmentation des images originales, les blocs sont très gros et de nombreux détails ont été perdus. De plus, l'arbre généré est trop compact. Le nombre de nœuds est très faible : respectivement 263 nœuds pour la fleur et 153 nœuds pour la porte, alors que la figure 2.6 présentait 2808 nœuds pour la fleur et 868 nœuds pour la porte.

Il est donc nécessaire de pouvoir affiner ces résultats. Une solution est de ne pas considérer uniquement le calcul du MSE, mais de mettre en place une **fonction d'énergie** [FTÉRICH 04]. Cette fonction d'énergie doit permettre de prendre en compte, au niveau des régions, l'erreur commise par rapport à l'image originale, mais également l'interaction qu'une région peut avoir avec ses voisins, c'est-à-dire la similarité inter régions. La fonction d'énergie pour une région R peut donc se formuler de la manière suivante :

$$E(R) = \text{intra}(R) + \text{inter}(R) \quad (2.4)$$

avec :

- R , la région étudiée ;
- $intra(R)$, le MSE de l'équation 2.3 adaptée pour une région ;
- $inter(R)$, une distance représentative de la similarité entre la région R et ses voisins (en 8 connexité) ;

L'énergie globale peut donc se formuler de la manière suivante :

$$E(I) = \sum_{r \in I} intra(r) + \sum_{r \in I} inter(r) \quad (2.5)$$

avec I , l'image étudiée et $r \in I$ toutes les régions de l'image I .

Cette fonction d'énergie doit nous permettre de trouver soit un minimum global, soit des minima locaux censés représenter le seuil le plus adapté à appliquer. Le but est donc de minimiser $E(I)$.

La problématique de la fonction d'énergie consiste donc à trouver une distance appropriée afin de quantifier la distance entre deux régions dans le calcul de $inter(R)$. Ce problème n'est pas trivial. Il existe une multitude de métriques possibles chacune d'entre elles plus ou moins adaptée à un problème donné.

La distance euclidienne

Notre première approche fût d'appliquer la **distance euclidienne** pour la fonction $inter(r)$. Cette distance est classique et très utilisée pour déterminer la distance entre deux régions (similarité entre deux régions).

La distance euclidienne entre deux régions R_1 et R_2 se formule de la manière suivante :

$$euclidian(R_1, R_2) = \left(\sum_{i \in X, Y, Z} \left(\frac{R_1(i)}{n_1} - \frac{R_2(i)}{n_2} \right)^2 \right)^{\frac{1}{2}} \quad (2.6)$$

avec :

- R_1 et R_2 , les deux régions étudiées ;
- n_1 et n_2 , respectivement le nombre de pixels de la régions R_1 et R_2 ;
- $R_1(i)$ et $R_2(i)$, respectivement la somme de toutes les intensités de R_1 et R_2 des composantes X, Y, Z de l'espace de couleur, dans notre cas RGB.

La fonction $inter(r)$ devient donc :

$$inter(r) = \frac{1}{l} \sum_{i \in l} euclidian(r, r_i) \quad (2.7)$$

avec l , le nombre total de voisins de la région r et r_i région voisine de r en 8 connexité.

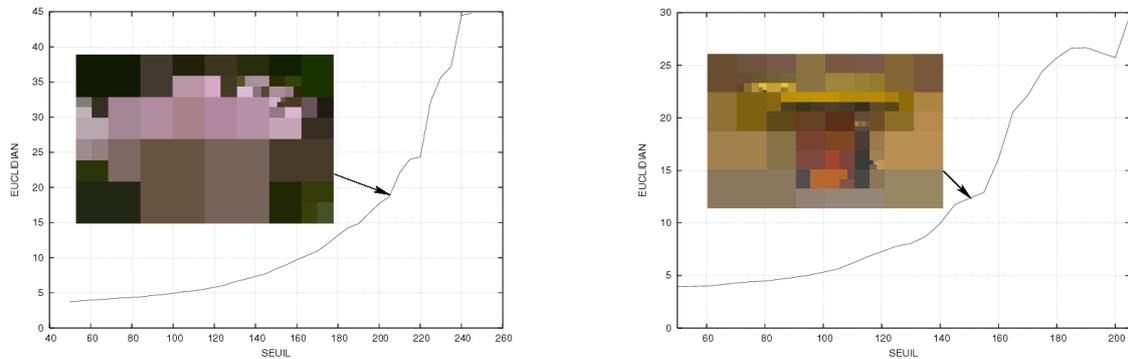


FIG. 2.8 – Courbes de la fonction d'énergie calculée avec la distance euclidienne pour la fleur et la porte

La figure 2.8 montre les courbes du calcul de la fonction d'énergie avec la distance euclidienne. Nous pouvons remarquer que ces courbes sont très proches de la courbe du MSE de la figure 2.7, avec néanmoins quelques fluctuations supplémentaires. Comme précédemment, ces fluctuations apparaissent trop tard au niveau de la valeur du seuil. Les seuils sont relativement similaires aux courbes du MSE : un seuil de 215 pour l'image de la fleur et un seuil de 150 pour l'image de la porte (au lieu de 200 et 155 pour le MSE). Les images de la figure 2.8 montrent les images reconstituées à l'aide de la carte des régions segmentées, comme précédemment, la segmentation est trop grossière : le nombre de nœuds traduit bien ce manque de détails dans les images ; respectivement 105 nœuds pour la fleur et 189 nœuds pour la porte (valeurs quasi identiques à celles obtenues précédemment).

Ces résultats, malgré la fonction d'énergie, ne sont pas encore satisfaisants. Pour essayer de les améliorer, nous allons modifier le terme $inter(r)$ de la fonction d'énergie c'est-à-dire essayer une autre métrique que la distance euclidienne. Cette nouvelle métrique sera la **distance de Student**.

La distance de Student

Afin d'affiner les résultats précédents, nous avons adopté une seconde approche : l'utilisation de la **distance de Student** qui prend en compte de manière plus robuste les propriétés colorimétriques des régions et utilise la distance euclidienne.

La distance de Student entre deux régions R_1 et R_2 se formule de la manière suivante :

$$student(R_1, R_2) = \left(\frac{euclidian(R_1, R_2)}{\left(\frac{n_1 v_1 + n_2 v_2}{n_1 + n_2 - 2} \right)^{\frac{1}{2}} \times \left(\frac{1}{n_1} + \frac{1}{n_2} \right)^{\frac{1}{2}}} \right)^2 \quad (2.8)$$

avec :

$$v_1 = \frac{1}{3} \left| \sum_{i \in X, Y, Z} \frac{1}{n_1} \left(R_1^2(i) - \left(\frac{R_1(i)}{n_1} \right)^2 \right) \right| \text{ et } v_2 = \frac{1}{3} \left| \sum_{i \in X, Y, Z} \frac{1}{n_2} \left(R_2^2(i) - \left(\frac{R_2(i)}{n_2} \right)^2 \right) \right| \quad (2.9)$$

- R_1 et R_2 , les deux régions étudiées ;
- n_1 et n_2 , respectivement le nombre de pixels de la régions R_1 et R_2 ;
- v_1 et v_2 , la variance calculée au niveau de R_1 et R_2 ;
- $R_1(i)$ et $R_2(i)$ respectivement la somme de toutes les intensités de R_1 et R_2 des composantes X, Y, Z de l'espace de couleur, dans notre cas RGB.
- $R_1^2(i)$ et $R_2^2(i)$ respectivement la somme de toutes les intensités au carré de R_1 et R_2 des composantes X, Y, Z de l'espace de couleur, dans notre cas RGB.

Les figures 2.9 et 2.10 montrent deux exemples de courbes de la fonction d'énergie calculée avec la distance de student pour les exemples de la fleur et de la porte .

Nous pouvons remarquer que les courbes varient nettement plus que dans les exemples précédents (figures 2.7 et 2.8) et présentent des minima exploitables. La courbe de la fleur en présente deux et celle de la porte en présente quatre.

En ce qui concerne la courbe de l'énergie de l'image de la porte, nous pouvons remarquer que le minimum global (le point (4)) correspond aux fluctuations que nous avons remarquées sur les courbes précédentes (MSE et énergie avec la distance euclidienne) : aux alentours de 150 de valeur de seuil.

Les figures 2.11 et 2.12 montrent les résultats de la segmentation pour l'image de la fleur pour les seuils correspondant au point (1) et (2) de la courbe de la figure 2.9. Ces points correspondent à un seuil de 85 pour le point (1) et de 120 pour le point (2).

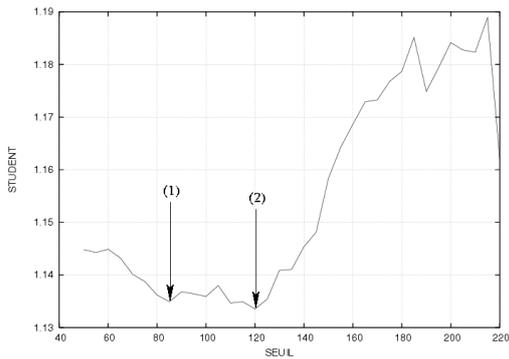


FIG. 2.9 – Courbe de la fonction d'énergie calculer avec la distance de student pour l'exemple de la fleur

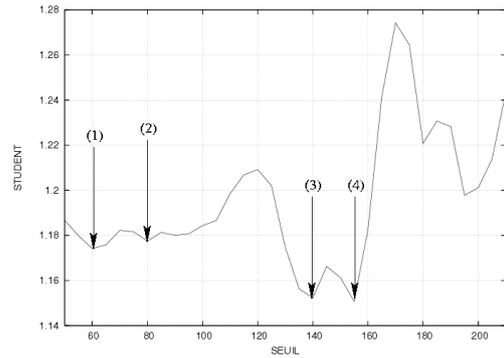


FIG. 2.10 – Courbe de la fonction d'énergie calculer avec la distance de student pour l'exemple de la porte



FIG. 2.11 – Résultat de la segmentation pour le point (1) correspondant à un seuil de 85



FIG. 2.12 – Résultat de la segmentation pour le point (2) correspondant à un seuil de 120

Nous pouvons constater que les deux résultats sont tout à fait satisfaisants, la fleur a bien été extraite et certains détails ont été préservés. La différence notable entre ces deux résultats vient de l'arbre binaire généré. En effet, l'arbre correspondant à la figure 2.11 (seuil de 85) possède un total de 5813 nœuds et celui correspondant à la figure 2.12 (seuil de 120) possède 3079 nœuds, ce qui correspond à un gain d'environ 47% au niveau de la représentation arborescente, gain non négligeable par rapport au résultat.

Les figures 2.13, 2.14 et 2.15 montrent les résultats de la segmentation, respectivement pour les points (1), (2) et (3) de la figure 2.10 correspondant aux seuils de 60, 80 et 140 (minima de la courbe d'énergie). Nous n'avons pas illustré le point (4) car il correspond aux résultats des figures 2.7 et 2.8 en ce qui concerne l'exemple de la porte.



FIG. 2.13 – Résultat de la segmentation pour le point (1) correspondant à un seuil de 60



FIG. 2.14 – Résultat de la segmentation pour le point (2) correspondant à un seuil de 80



FIG. 2.15 – Résultat de la segmentation pour le point (3) correspondant à un seuil de 140

Comme pour l'exemple de la fleur, ces résultats sont satisfaisants. Les éléments principaux ont été extraits. Pour la figure 2.15, la segmentation est trop grossière et de nombreux détails ont été perdus.

Au niveau des arbres binaires, les nombres de nœuds sont de l'ordre de 6019, 3801 et 305 respectivement pour les points (1), (2) et (3).

Les deux points intéressants de la courbe 2.10 sont bien entendu les points (1) et (2). Comme pour le cas de la fleur, nous avons un gain de l'ordre de 37% au niveau de la complexité des arbres tout en conservant une segmentation correcte.

La question que nous pouvons nous poser maintenant est : *Quel minimum sélectionner ?* Plusieurs solutions sont possibles :

1. soit sélectionner le minimum global, ce qui entraîne le calcul de toute la courbe d'énergie (gourmand en temps de calcul). Nous avons vu que dans l'exemple de la figure 2.10, ce minimum global ne donnait pas de bons résultats.
2. soit sélectionner le premier minimum local rencontré, ce qui donne de bons résultats comme le montrent les figures 2.11 et 2.13, mais donnent également des structures d'arbres binaires moins compactes ;
3. une autre possibilité serait peut être de lisser la courbe, et de sélectionner le premier minimum local ou encore de sélectionner un seuil moyen compris entre les deux premiers minima locaux.

Cette question reste ouverte et de nombreuses autres possibilités peuvent être testées et envisagées.

Afin d'achever cette partie sur la stratégie descendante et avant de conclure, nous allons illustrer avec un exemple complet, la segmentation, la détermination d'un seuil optimal avec l'arbre binaire correspondant à ce seuil.



FIG. 2.16 – Image originale

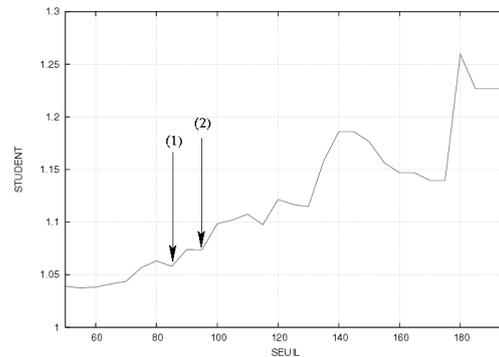


FIG. 2.17 – Courbe de la fonction d'énergie calculer avec la distance de Student pour l'exemple de l'avion

La figure 2.16 montre l'image originale de l'exemple proposé. Il s'agit d'une image couleur de taille 394×600 , codée en **Bitmap** et d'espace couleur RGB. La figure 2.17 montre la courbe de la fonction d'énergie calculée avec la distance de Student. Les deux minima locaux sont représentés avec les points (1) et (2). Ils correspondent aux valeurs de seuil 85 et 95.

Les figures 2.18, 2.19, 2.21 et 2.22 montrent les cartes de segmentation et les images recolorées respectivement pour les points (1) et (2) de la figure 2.17.

Les figures 2.20 et 2.23 montrent les arbres binaires générés respectivement pour un seuil de 85 et pour un seuil de 95. Le nombre de nœuds est de 430 pour la figure 2.20 et de 240 pour la figure 2.23 soit quasiment moitié moins.

Les points noirs représentés sur les figures correspondent aux feuilles des arbres mais également aux régions représentées par les cartes de segmentation : les feuilles de l'arbre de la figure 2.20 aux régions de la figure 2.18 et les feuilles de l'arbre de la figure 2.23 aux régions de la figure 2.21.

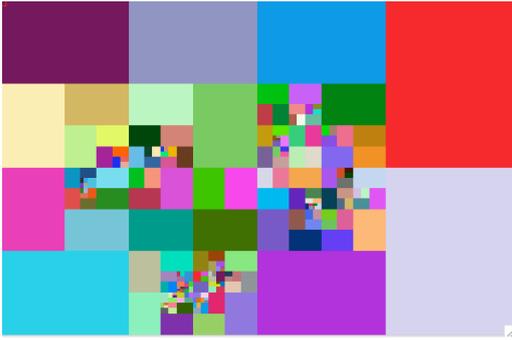


FIG. 2.18 – Carte des régions segmentées avec un seuil de 85 (point (1))

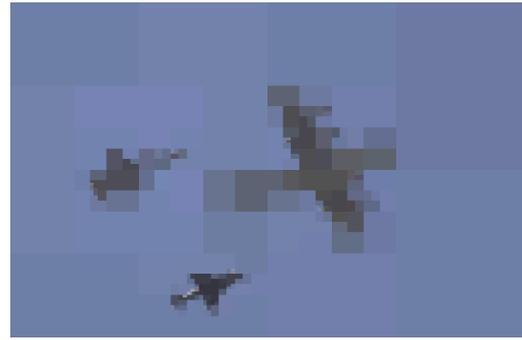


FIG. 2.19 – Image recolourée à partir de l'image originale et de la carte de segmentation de la figure 2.18

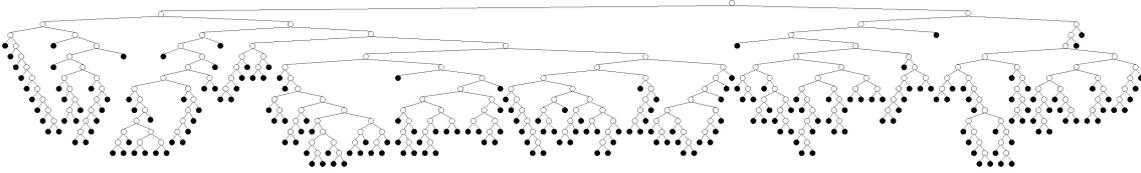


FIG. 2.20 – Arbre binaire généré au point (1)

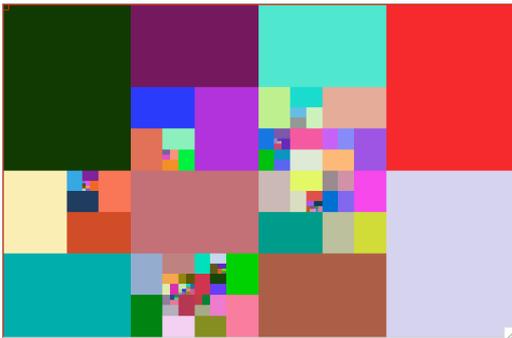


FIG. 2.21 – Carte des régions segmentées avec un seuil de 95 (point (2))

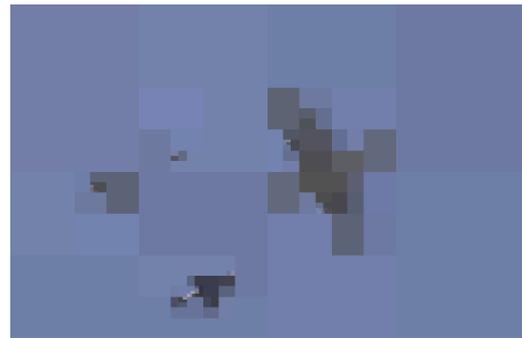


FIG. 2.22 – Image recolourée à partir de l'image originale et de la carte de segmentation de la figure 2.21

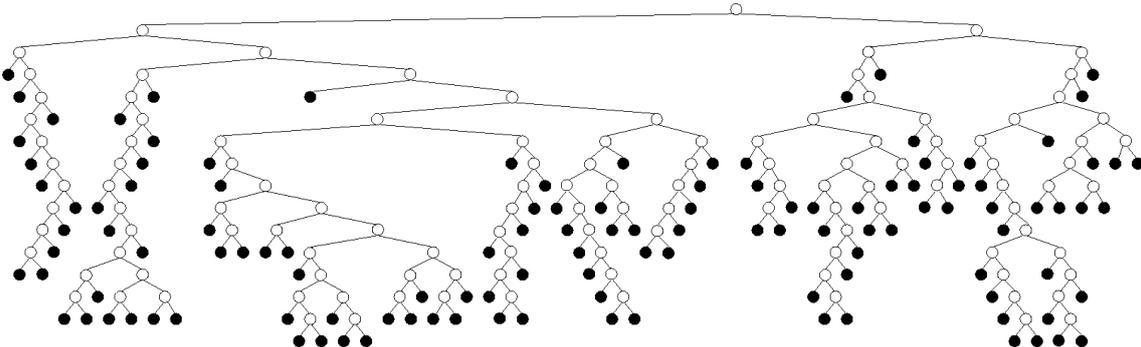


FIG. 2.23 – Arbre binaire généré au point (2)

2.4 Conclusions, perspectives

Dans cette partie, nous avons présenté la stratégie descendante mise en œuvre. Cette stratégie consiste à générer l'arbre depuis la racine jusqu'aux feuilles correspondant aux

régions segmentées finales.

Nous avons également soulevé un certain nombre de problèmes auxquels nous avons fourni des réponses. Ces réponses ne sont pas exhaustives et peuvent encore être améliorées.

Les problèmes soulevés sont les suivants :

1. le prédicat d'uniformité du bloc à diviser ;
2. la détermination d'un seuil optimal conduisant à approfondir la fonction d'énergie.

Bien entendu, des améliorations peuvent être apportées à l'étude que nous avons entreprise.

Tout d'abord, en ce qui concerne le prédicat d'uniformité, d'autres critères peuvent être testés. Par exemple :

1. un calcul d'uniformité de la distribution des couleurs du bloc étudié. Ce calcul peut également entraîner l'étude des espaces couleurs (YUV pour des images multimédias et audiovisuels, HSL, ...) plus adaptés à certains types d'images ;
2. un critère d'uniformité plus spécifique et complexe faisant intervenir plusieurs termes : par exemple, pondérer le contraste et la variance ou encore utiliser les histogrammes couleurs, ...

Concernant la détermination du seuil optimal de segmentation, de nombreuses possibilités s'offrent à nous. Nous pouvons par exemple penser à étudier d'autres métriques ou encore améliorer la fonction d'énergie en pondérant chaque terme par un facteur.

Une autre possibilité non exploitée serait d'élaguer l'arbre une fois la segmentation terminée afin d'essayer d'obtenir la meilleure représentation possible. Des techniques d'élagage de l'arbre seront exposées dans la partie concernant la stratégie ascendante. Nous pouvons également les utiliser pour la stratégie descendante.

En ce qui concerne la méthode de segmentation en elle-même, nous avons vu qu'elle donnait des résultats plus ou moins fidèles à l'image originale. Du fait de l'algorithme de division par bloc, l'image reconstituée possède des «effets de blocs (artefacts de compression)» comme dans la norme JPEG. Cette représentation est communément utilisée, du fait de la compacité de la structure, pour des échanges d'informations.

Une autre possibilité à étudier serait de modifier la segmentation par blocs par une segmentation par morphologie mathématique de type LPE tout en conservant les bénéfices des arbres binaires : une sorte de «LPE binaire». L'algorithme resterait descendant.

Partant de l'image originale, nous pouvons déposer deux germes ; soit de manière aléatoire, soit en sélectionnant les deux plus petits minima parmi tous les minima locaux de l'image, soit à l'aide d'une autre technique de «semence».

Une fois les germes placés, nous les faisons croître jusqu'à l'obtention de deux régions. Nous remplaçons alors deux nouveaux germes sur chaque région obtenue et nous recommençons. Nous itérons de cette manière tant que le prédicat d'uniformité n'est pas satisfait ou lorsqu'une région arrive à une taille donnée.

Cette méthode semble intéressante car nous pensons qu'elle permettrait de conserver les contours de l'image, de minimiser le nombre de régions et de permettre de faire une étude comparative avec la stratégie ascendante qui sera exposée dans la partie suivante.

Stratégie ascendante

Dans cette partie, nous exposerons l'étude de la stratégie ascendante. Cette méthode consiste à générer l'arbre binaire à partir d'une image déjà segmentée (création de l'arbre à partir des feuilles).

Tout d'abord, nous exposerons la technique de segmentation utilisée (LPE) ainsi que la chaîne de traitements nécessaire à sa réalisation.

Ensuite, nous exposerons les techniques pour générer l'arbre binaire et les méthodes d'élagage que nous avons étudiées.

Afin d'illustrer nos propos, des exemples ponctueront cette partie.

Comme nous l'avons déjà dit, la stratégie ascendante de génération de l'arbre binaire consiste à démarrer des feuilles de l'arbre (les régions segmentées de l'image) pour arriver à la racine (l'image complète). La création de l'arbre se fait à l'aide d'un algorithme de «fusion/génération».

3.1 Segmentation par LPE

La première étape de la stratégie mise en place consiste à obtenir une image pré segmentée. Dans la littérature, la méthode la plus communément utilisée est la LPE [SALEMBIER 00], mais la méthode de génération de l'arbre binaire reste valable pour n'importe quelle méthode de segmentation.

Comme nous l'avons déjà évoqué au début de ce rapport, la LPE est issue de la morphologie mathématique [MEYER 92], [SERRA 82] et consiste à voir l'image comme un relief. La segmentation est réalisée grâce à une croissance de régions issue de germes. La détermination des germes se fait par calcul des minima locaux à partir du gradient de l'image. Une fois la croissance des régions achevée, nous obtenons l'image segmentée.

Obtention de l'image segmentée : chaîne de traitements

Afin d'obtenir notre image segmentée par LPE, nous devons mettre en place œuvre une chaîne de traitements. La figure 3.24 résume cette chaîne.

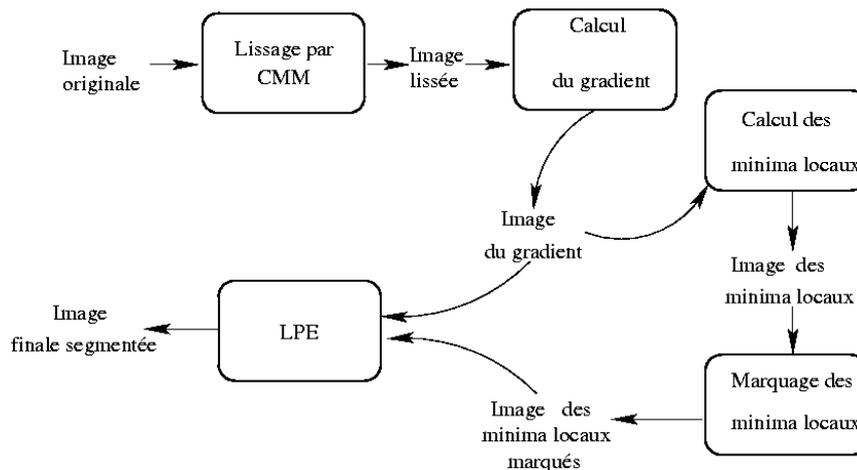


FIG. 3.24 – Chaîne de traitements pour obtenir l'image segmentée par la LPE

Avant de pouvoir réaliser la LPE, une succession d'étapes est nécessaire :

1. le lissage par **CMM** (lissage par diffusion non linéaire par courbure). Cette étape permet de minimiser le nombre de germes pour la croissance des régions ;
2. le calcul du gradient de l'image lissée ;
3. la détection des minima locaux du gradient ;
4. le marquage de ces minima (il s'agit d'une simple numérotation) ;
5. enfin la LPE en utilisant l'image du gradient et les germes correspondant aux minima marqués du gradient.

Les figures 3.25 à 3.29 montrent les images obtenues après les étapes de la chaîne de traitements de la figure 3.24. Nous obtenons un total de 2042 régions à l'issue de la LPE (figure 3.29). Afin de réduire le nombre de ces régions, nous fusionnons toutes les régions ayant une surface inférieure à 50 pixels afin de minimiser le nombre de feuilles de l'arbre binaire initial.



FIG. 3.25 – Image originale



FIG. 3.26 – Image lissée



FIG. 3.27 – Image du gradient

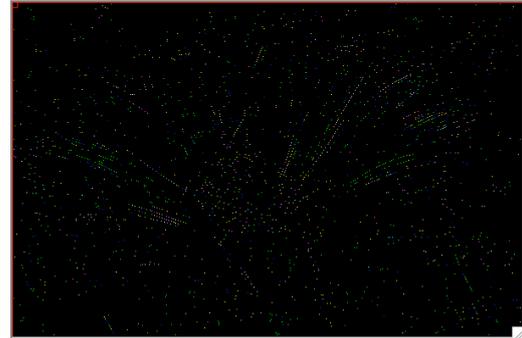


FIG. 3.28 – Image des minima locaux



FIG. 3.29 – Image finale segmentée par LPE (2042 régions)



FIG. 3.30 – Fusion des petites régions de surface inférieure à 50 pixels (888 régions)

Fusion des petites régions

Pour fusionner les petites régions, nous utilisons la graphe d'adjacence des régions. Ce graphe permet de connaître les voisins de chaque région de l'image segmentée

Le principe de la fusion est le suivant :

pour chaque région

1. nous parcourons tous ses voisins ;
2. si le voisin courant possède une surface inférieure à 50 pixels (taille fixée empiriquement) alors nous l'agglomérons à la région étudiée ;
3. nous mettons à jour le graphe d'adjacence des régions.

Le résultat de cette fusion des petites régions est illustré à la figure 3.30. Grâce à cette fusion, nous passons de 2042 régions à 888 régions.

L'image des régions fusionnées va nous servir de point de départ pour l'algorithme de fusion/génération.

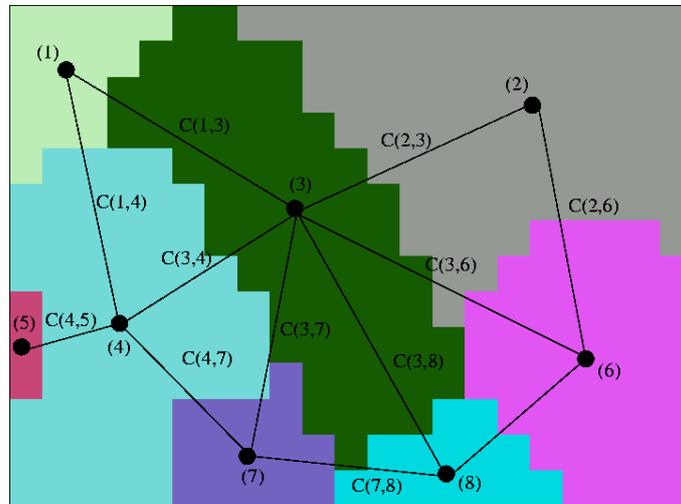


FIG. 3.31 – Exemple de graphe d'adjacence des régions

La figure 3.31 montre un exemple de graphe d'adjacence des régions. Chaque sommet est numéroté et représente une région de l'image. Chaque région est reliée à ses voisins par un lien (ou arête) possédant une valeur ($C(1,4)$ par exemple). Cette valeur (ou coût) est calculée grâce à une fonction (fonction de coût) qui traduit la similarité entre les deux sommets (régions) que le lien relie.

3.2 Algorithme de «fusion/génération»

D'un point de vue algorithmique, nous utilisons pour la fusion, une structure de données de type **tas hiérarchique** permettant d'emmagasiner toutes les régions initiales et celles créées par fusion.

Le tas contient toutes les arêtes du graphe d'adjacence et elles sont classées en fonction de leur valeur de coût (de manière croissante).

L'algorithme d'arrête lorsqu'il ne reste plus qu'une seule région (la racine de l'arbre) c'est-à-dire quand le tas est vide.

Calcul de la similarité entre régions en vue d'une fusion

Afin de savoir comment fusionner deux régions et dans quel ordre, nous utilisons une métrique représentative de la similarité entre deux régions.

La première approche est d'utiliser une métrique représentative des caractéristiques colorimétriques. Ce sera la **distance de Student** comme pour la stratégie descendante dont la formulation se trouve à l'équation 2.8. Dans le cas de la segmentation par blocs cette métrique suffisait. Dans le cas actuel, il serait judicieux d'exploiter les informations spatiales issues de la segmentation par LPE (informations de contours, de périmètre et de surface).

Ce sont pour ces raisons que nous utilisons pour caractériser ces informations, la **distance de Schettini** [SCHETTINI 93] dont la formulation est la suivante :

$$schettini(R_1, R_2) = \frac{4 \times f_{12}^2}{\min(p_1, p_2) \times \max(R_{l_1}, R_{l_2})} \quad (3.10)$$

avec :

- R_1 et R_2 , les deux régions voisines à étudier ;
- p_1 et p_2 , respectivement le périmètre de R_1 et de R_2 ;
- f_{12} la frontière commune entre R_1 et R_2 ;
- l_1 et l_2 , respectivement les voisins de R_1 et de R_2 ,
- $max(R_{l_1}, R_{l_2})$, la frontière commune la plus grande parmi les voisins de R_1 et de R_2 .

Notre fonction de coût C entre deux régions voisines R_1 et R_2 peut donc se formuler de la manière suivante :

$$C(R_1, R_2) = student(R_1, R_2) \times schettini(R_1, R_2) \quad (3.11)$$

Cette fonction de coût va nous permettre de connaître l'ordre prioritaire de fusion : l'arête ayant le coût le plus faible parmi toutes les arêtes du graphe sera fusionnée en priorité.

Algorithme de fusion/génération

L'algorithme de fusion/génération peut se résumer de la manière suivante :

```

0 -----
1 cout() : une fonction de calcul du coût d'une arête
2 graphe : le graphe d'adjacence des régions
3 arête : une arête
4 tas    : le tas hiérarchique
5 ab     : l'arbre binaire
6 -----
7
8 début
9 {
10 //INITIALISATION
11 initialiser graphe avec toutes les régions de l'image initiale;
12 initialiser tas avec toutes les arêtes de graphe, classer le tas
13   en fonction de cout(arête);
14
15 //FUSION GENERATION
16 Tant que ( NON( tas.vide() ) )
17 {
18   arête = tas.Pop();
19   créer deux noeuds avec les sommets de arête;
20   fusionner les deux sommets, créer un nouveau sommet:
21   mettre à jour les données du nouveau sommet;
22   mettre à jour graphe (les arêtes, les sommets);
23   mettre à jour ab, avec le nouveau sommet;
24   mettre à jour tas avec les nouvelle arêtes,
25     en calculant cout(arête);
26 } //fin Tant que
27
28 }
29 fin

```

La figure 3.32 montre (de haut en bas et de gauche à droite) le déroulement de l'algorithme de fusion des régions de l'image originale et la création de l'arbre binaire. L'algorithme commence avec toutes les feuilles de l'arbre (les régions segmentées) et fini lorsque la racine est atteinte.

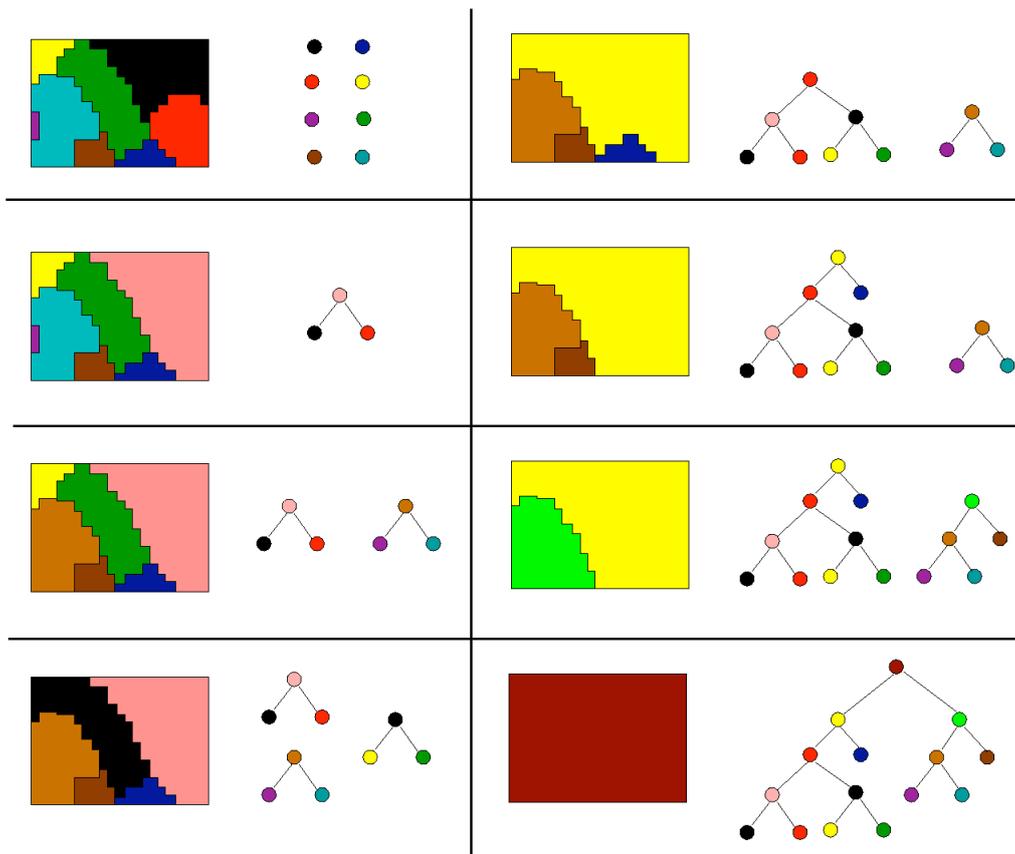


FIG. 3.32 – Déroulement de l'algorithme «fusion/génération»



FIG. 3.33 – Image originale



FIG. 3.34 – Carte des régions après segmentation par LPE et fusion des petites surfaces

Algorithme de fusion/génération sur un exemple

Les figures 3.33 à 3.35 montrent le résultat de l'algorithme de fusion/génération sur un exemple : l'image de l'avion. L'image de la figure 3.33 montre l'image originale et celle de la figure 3.34 montre le résultat de la segmentation par LPE et de la fusion des petites régions. La figure 3.35 montre la structure de l'arbre binaire généré à l'issue de l'algorithme.

Nous pouvons remarquer que comparativement à la méthode descendante la structure de l'arbre est plus compacte et le nombre de nœuds est moins important (190 nœuds) par rapport aux figures 2.20 (460 nœuds) et 2.23 (240 nœuds). La structure en elle même est

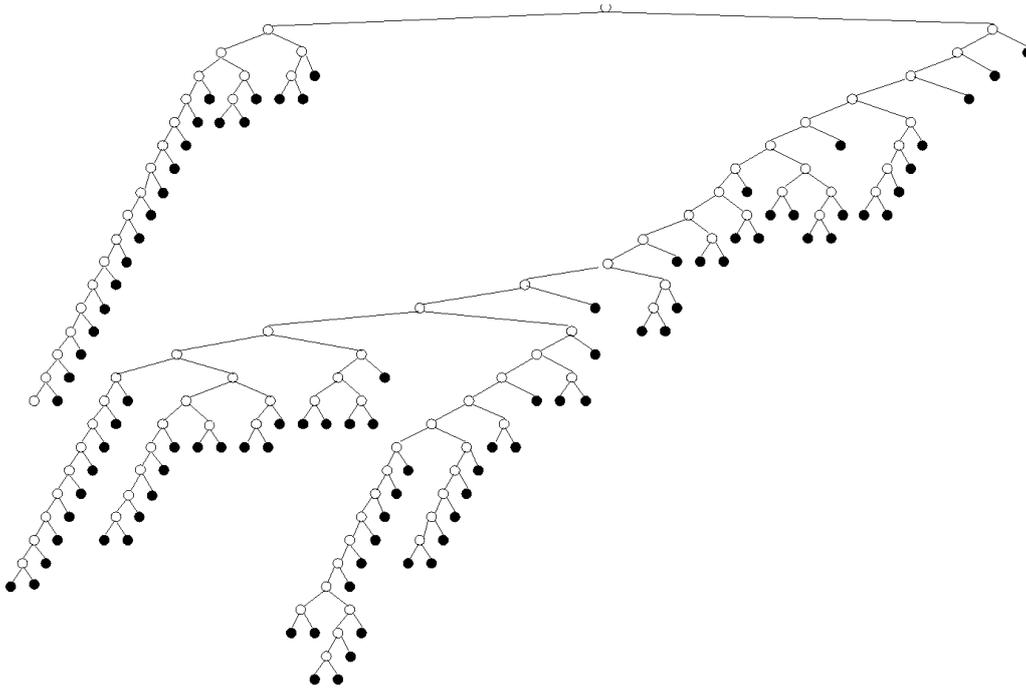


FIG. 3.35 – Arbre binaire généré à partir de la carte des régions de la figure 3.34

différente ; ceci résultant de la différence de stratégie développée. Nous pouvons également voir que le résultat de la segmentation est lui aussi différent : l'utilisation de la LPE permet de conserver les informations de contours.

Une fois la représentation arborescente générée, nous allons essayer de l'élaguer (couper des nœuds) afin d'optimiser la structure et d'améliorer la segmentation.

3.3 Élagage de l'arbre

Élaguer l'arbre consiste à le couper à certains endroits. La question qui se pose est : *comment définir les sites judicieux ?*

Pour cela, nous allons utiliser une fonction d'énergie.

Fonction d'énergie

Comme pour la stratégie descendante, nous allons utiliser une fonction d'énergie qui doit rendre compte, au niveau des régions, des erreurs commises par rapport à l'image originale mais aussi des interactions entre leurs voisins (similarité). C'est pour cela que notre fonction d'énergie aura la même forme que l'équation 2.4 c'est-à-dire $E(R) = intra(R) + inter(R)$.

La différence avec l'équation que nous avons précédemment utilisée et la modification du terme $inter(R)$. En effet, comme pour la valuation des arêtes du graphe d'adjacence des régions, nous allons profiter des informations spatiales données par la segmentation. Par conséquent, le terme $inter(R)$ aura la forme suivante :

$$inter(R) = \frac{1}{l} \sum_{l \in V_R} student(R, r_l) \times schettini(R, r_l) \quad (3.12)$$

avec :

- R , la région, le nœud étudié ;
- $student$, la distance de student de l'équation 2.8 ;
- $schettini$, la distance de schettini de l'équation 3.10 ;
- l , le nombre total de voisin de R ;
- V_R , l'ensemble des voisins de R en 8 connexité ;

Notons que le terme $intra(R)$ ne change pas, nous utilisons toujours le Mean Square Error (MSE de l'équation 2.3).

Au fur et à mesure de l'élagage, en faisant la somme des énergies au niveau des feuilles de l'arbre, nous devons pouvoir obtenir des fluctuations de la valeur de l'énergie. Cela devrait nous permettre de définir un élagage et une représentation optimale.

Comment élaguer l'arbre ?

Dans un premier temps, nous avons réalisé un élagage grossier c'est-à-dire par une «droite» puis pour affiner les résultats nous avons utilisé un élagage par une «courbe».

Élagage par une «droite»

Le principe de l'élagage par une droite est simple. Il consiste à couper l'arbre en fonction de sa profondeur en partant des feuilles les plus profondes pour arriver jusqu'à la racine. La stratégie est la suivante : en élaguant l'arbre en fonction de la profondeur, nous calculons la somme des énergies au niveau des feuilles.

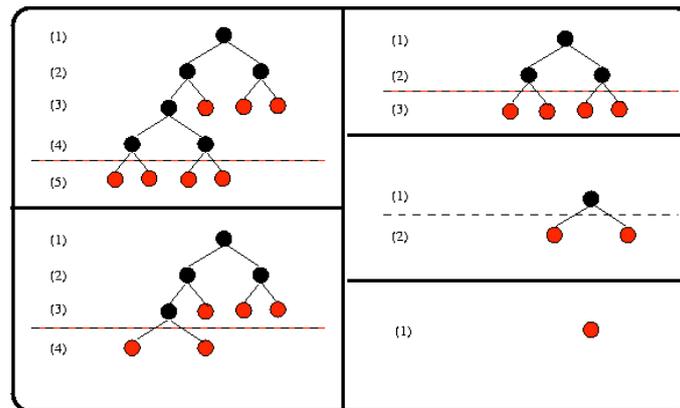


FIG. 3.36 – Schéma de principe de l'élagage par une droite

La figure 3.36 illustre (de haut en bas et de gauche à droite) le principe de l'élagage par une droite. A chaque niveau les feuilles sont mises à jour. Le trait en pointillé représente l'endroit de la coupure.

Les figures 3.37 et 3.38 montrent la courbe de l'énergie en fonction du nombre de régions (le nombre de feuilles de l'arbre) lors d'un élagage par une droite. La courbe de la figure 3.37 représente la somme des énergies des feuilles et la courbe de la figure 3.38 représente la même somme mais normalisée par le nombre de feuilles.

Nous pouvons constater que malgré les fortes fluctuations des deux courbes non normalisée, les deux résultats concordent. Les deux minima locaux (points 1 et 2 de la courbe 3.37) se retrouvent quasiment au même endroit ce qui semble normal étant donné que l'élagage est réalisé de la même manière.

Les fluctuations des courbes proviennent de la méthode d'élagage en elle-même : couper par une droite modifie énormément la structure de l'arbre.

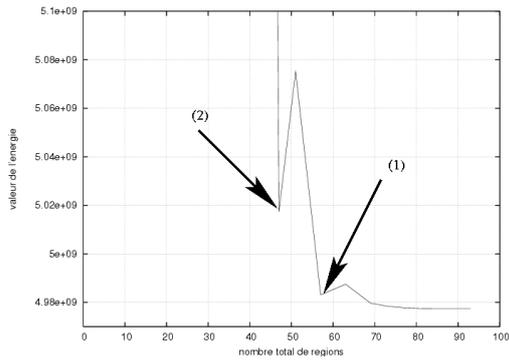


FIG. 3.37 – Courbe de l'énergie en fonction du nombre de régions pour l'exemple de l'avion : élagage par une droite

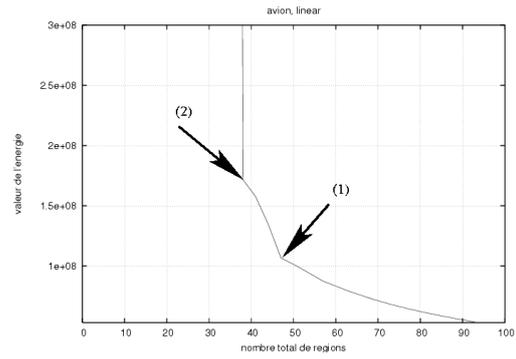


FIG. 3.38 – Courbe de l'énergie normalisée en fonction du nombre de régions pour l'exemple de l'avion : élagage par une droite



FIG. 3.39 – Image recolorée



FIG. 3.40 – Carte de segmentation après élagage par une droite (36 régions)

Les figures 3.39 et 3.40 montrent l'image recolorée à partir de la carte de segmentation et de l'image originale ainsi que la carte des régions obtenue après élagage par une droite. Nous pouvons constater que le résultat est satisfaisant même si des détails ont été supprimés lors de l'élagage.

La figure 3.41 montre le résultat sur l'arbre binaire.

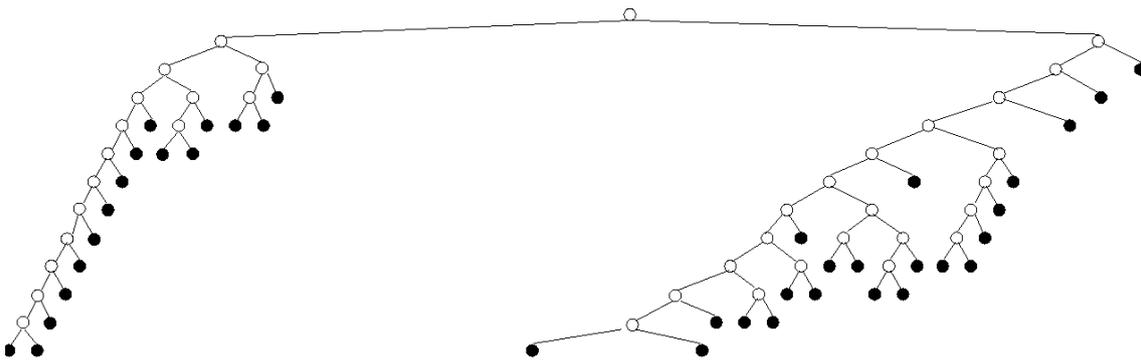


FIG. 3.41 – Arbre binaire après l'élagage par une droite

Élagage par une «courbe»

La seconde approche d'élagage consiste à couper l'arbre à l'aide d'une courbe. Le principe est la suivante : choisir à un niveau local le nœud idéal à enlever à un instant donné.

Comment choisir ce nœud ?

Nous allons utiliser la fonction d'énergie. Deux approches ont été essayées :

1. la première consiste à déterminer, à chaque pas de l'élagage, parmi tous les nœuds pères des feuilles, quel est le père qui possède l'énergie la plus petite parmi tous ces pères. L'élagage s'effectue au niveau de ce candidat. Une fois l'arbre coupé à cet endroit, nous recalculons la fonction d'énergie au niveau des feuilles. Nous itérons jusqu'à l'obtention de la racine ;
2. la seconde approche est similaire à la première. La différence provient du fait qu'au lieu de se référer à l'énergie (de chaque nœud père des feuilles) nous analysons la différence d'énergie entre le père et ses deux fils. Le père qui possède la plus petite de ces différences parmi tous les pères est le candidat à l'élagage. De même que précédemment, une fois l'arbre coupé à l'endroit du candidat, nous recalculons l'énergie au niveau des feuilles. Ce principe est itéré jusqu'à l'obtention de la racine.

Pour résumer, ces approches consistent à sélectionner parmi tous les pères possédant au moins une feuille ; le père qui est le plus similaire par rapport à l'image originale pour la première approche et celui qui est le plus similaire par rapport à ces fils en ce qui concerne la seconde approche.

L'élagage par une courbe doit permettre de couper l'arbre d'abord à des endroits où les régions sont très similaires pour arriver enfin aux endroits qui le sont le moins.

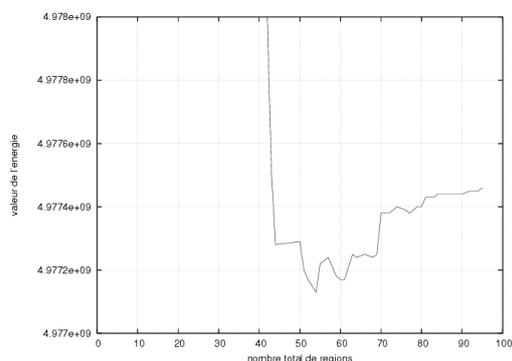


FIG. 3.42 – Courbe de l'énergie en fonction du nombre de régions pour l'exemple de l'avion : élagage par une courbe (première approche)

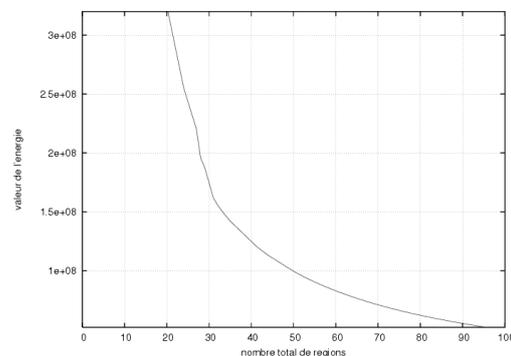


FIG. 3.43 – Courbe de l'énergie normalisée en fonction du nombre de régions pour l'exemple de l'avion : élagage par une courbe (première approche)

Les figures 3.42 et 3.43 montrent la fonction d'énergie pour la première approche d'élagage par une courbe : l'une normalisée l'autre non. Les figures 3.44 et 3.45 montrent la fonction d'énergie pour la seconde approche.

Nous pouvons constater que les deux approches donnent des résultats similaires et nous laisseraient penser qu'elles sont équivalentes et que l'élagage se fait de la même manière. En ce qui concerne les courbes non normalisées (figures 3.42 et 3.44), nous pouvons voir qu'elles possèdent beaucoup de fluctuations ce qui est normal au vue de la méthode de calculs : une grande valeur de l'énergie risque de rendre négligeable les autres valeurs. Malgré tout, les fluctuations peuvent rendre exploitables ces courbes en cherchant par exemple un minimum global. Ces deux courbes se corrèlent néanmoins dans la zone comprise 50 et 70

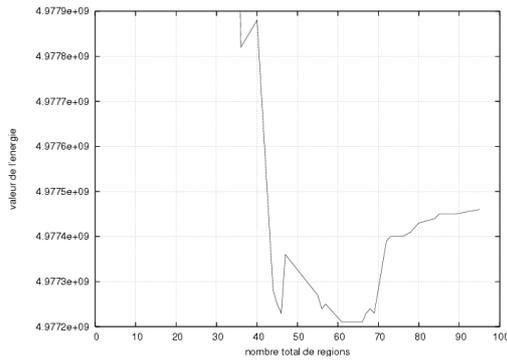


FIG. 3.44 – Courbe de l'énergie en fonction du nombre de régions pour l'exemple de l'avion : élagage par une courbe (seconde approche)

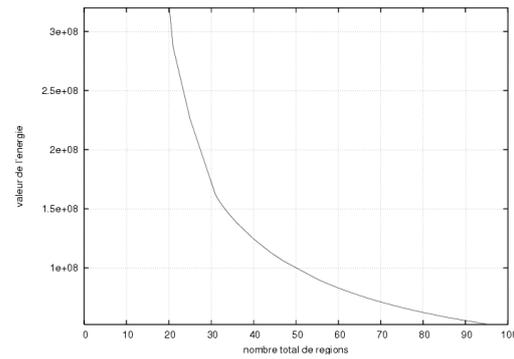


FIG. 3.45 – Courbe de l'énergie normalisée en fonction du nombre de régions pour l'exemple de l'avion : élagage par une courbe (seconde approche)

régions.

Sur les courbes normalisées (figures 3.43 et 3.45), nous pouvons tout d'abord constater qu'elles n'ont pas la même allure que les deux précédentes. Elles sont toutes les deux croissantes et ne présentent que peu de fluctuations ce qui semble normal vu la normalisation du calcul. Ces deux courbes nous montrent que l'élagage suit la structure de l'arbre d'où le manque de fluctuations et la croissance constatée.

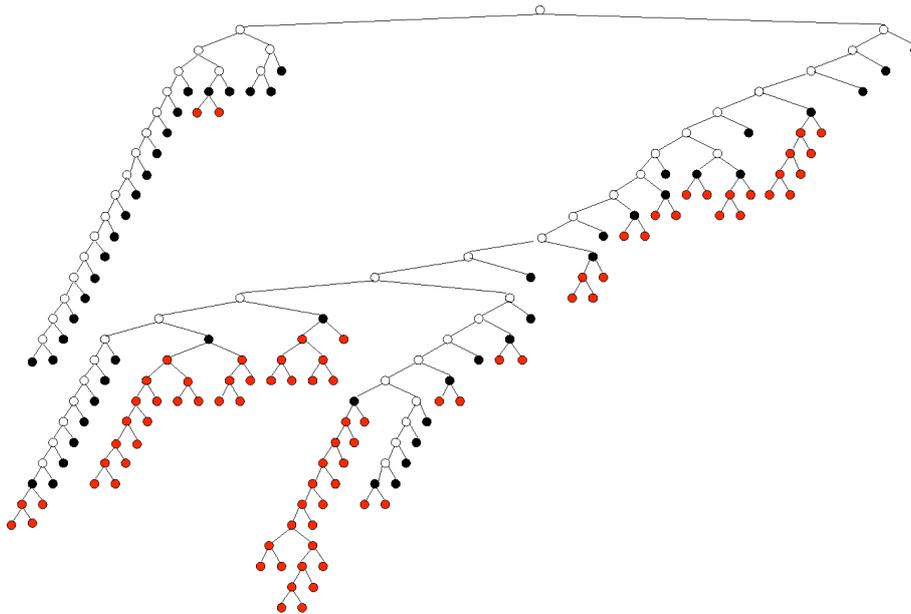


FIG. 3.46 – Arbre binaire élagué une courbe (première approche)

La figure 3.46 montre l'arbre binaire après élagage par une courbe selon la première approche avec un total de 54 feuilles : les nœuds rouges correspondent aux nœuds détruits. Pour obtenir l'arbre, l'algorithme a élagué 37 nœuds. Avec la seconde approche, pour obtenir le même résultat : un arbre de 55 nœuds, l'algorithme a sélectionné 24 nœuds mais les coupures se produisent aux mêmes endroits pour les deux méthodes.

Les figures 3.47 et 3.48 montrent l'image reconstituée à partir de la carte des régions de la figure 3.48. Cette carte correspond aux feuilles de l'arbre binaire élagué de la figure 3.46.

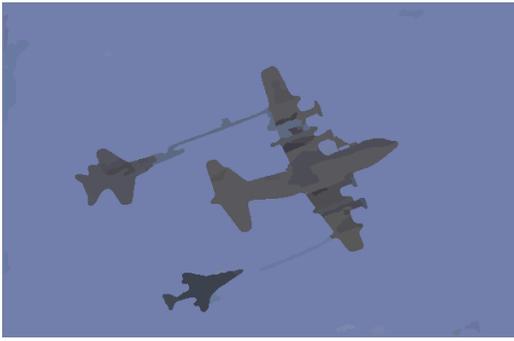


FIG. 3.47 – Image recolorée obtenue à partir de la figure 3.48

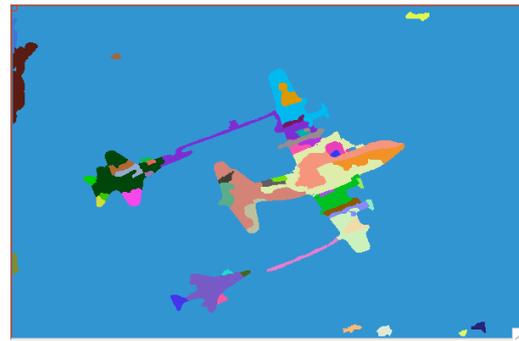


FIG. 3.48 – Carte des régions obtenue après élagage selon une courbe avec la première approche (54 régions)

3.4 Conclusions, perspectives

Dans cette partie, nous avons présenté la stratégie ascendante développée dans le cadre des arbres binaires ainsi que notre algorithme de fusion/génération permettant à partir d'une image segmentée de générer l'arbre binaire. La segmentation de l'image est indépendante de notre méthode de génération de la structure arborescente. Dans notre cas, nous avons utilisé une LPE permettant ainsi d'obtenir des informations spatiales sur l'image (informations de contours, de périmètre, de surface).

Afin de fusionner les régions, nous avons utilisé un critère de fusion prenant en compte les informations colorimétriques et spatiales des régions étudiées.

Enfin, nous avons présenté des méthodes d'élagage de l'arbre binaire permettant d'obtenir une représentation compacte à l'aide d'une fonction d'énergie.

Des améliorations peuvent être apportées dans tous les points que nous avons abordés dans cette partie.

La première des modifications qui pourraient intervenir dans notre étude est l'utilisation d'images non compressés afin d'améliorer la reconstruction des images à partir de l'arbre binaire et donc d'éviter des effets résultants des artefacts de compression.

Le deuxième point serait la modification de la fonction d'énergie et/ou du calcul de la valuation des arêtes lors de la décision de fusion. En effet, dans notre étude, nous avons pris la même méthode pour la fusion des régions et pour l'évaluation de la similarité entre régions. Il est bien évidemment possible d'utiliser d'autres métriques dans ce calcul de similarité. Ceci permettrait d'avoir une indépendance entre le critère de fusion et la fonction d'énergie permettant la détermination de l'élagage optimale.

En ce qui concerne la fonction d'énergie, il est envisageable de la modifier. Par exemple pondérer chaque terme avec un facteur permettant d'influencer les résultats de l'élagage : «forcer» l'algorithme à élaguer d'abord les nœuds les plus profonds, ...

Le dernier point de cette conclusion concerne l'élagage, comme nous l'avons vu, à l'heure actuelle, la détermination d'un «seuil» d'élagage n'est pas encore optimale. Une modification possible du processus peut être envisagée : dans le cas de l'élagage par une courbe, il n'y a pas de restriction dans la sélection des candidats. Notre algorithme permet, par exemple, qu'un nœud n'ayant qu'une seule feuille puisse être supprimé. La conséquence est la possibilité de couper en une seule fois une grande partie de l'arbre ce qui peut entraîner la suppression de détails importants dans l'image. Nous pourrions envisager, dans la sélection des candidats à l'élagage, à ajouter la contrainte suivante : un nœud ne peut être candidat à la suppression que si ses deux fils sont des feuilles.

Concernant les courbes de la fonction d'énergie ; nous avons vu que dans le cas de la normalisation du calcul, elles étaient peu fluctuantes, mais possédaient une courbure. Une solution possible dans la détermination d'un seuil d'élagage optimale consisterait à calculer les tangentes à la courbe ou directement de calculer la courbure permettant ainsi de déterminer le moment le plus optimal d'arrêter l'élagage de l'arbre obtenant ainsi une structure arborescente idéale.

Conclusion générale

Dans ce document, nous avons présenté l'étude réalisée sur les représentations arborescentes et plus particulièrement les arbres binaires dans le domaine de l'image. Cette structure de données est de plus en plus utilisée en imagerie du fait de sa compacité, de ses possibilités à contenir des informations pertinentes, de sa représentation «multi-échelle» de l'image . . . Tout ceci permet d'envisager son application à de nombreux domaines (compression, indexation, reconnaissance de formes, segmentation . . .).

Le propos de ce document était l'étude de ces arbres binaires et notamment leur mise en œuvre. De cette étude, nous avons pu dégager deux approches dans les techniques de création de cette représentation arborescente.

La première de ces approches, que nous avons appelé la «stratégie descendante», permet de partir d'une image quelconque et d'aboutir à une segmentation par blocs et à la création de l'arbre binaire. Durant l'étude de cette méthode, nous avons dégagé plusieurs problématiques et donné certaines solutions. Parmi elles, des pistes concernant les prédicats d'uniformité servant au processus de division par blocs. Ensuite, nous avons mis en œuvre un certain nombre de métriques permettant de calculer la similarité entre régions donnant ainsi une solution dans la détermination d'un seuil idéal de segmentation.

La seconde approche que nous avons étudiée est la «stratégie ascendante». Cette méthode est également une méthode permettant de construire l'arbre binaire représentant une image. Contrairement à la technique descendante, cette approche ne permet pas de réaliser une segmentation de l'image. Elle a besoin, pour être utilisée, de partir d'une image déjà segmentée. La méthode de segmentation est indépendante de la création de la structure arborescente. En ce qui nous concerne, nous avons utilisé une LPE permettant d'obtenir des informations de contours sur les images. Ces informations nous ont servi dans la détermination d'un critère de fusion permettant de générer l'arbre binaire. Dans cette partie, nous avons étudié des techniques d'élagage de l'arbre pouvant également servir pour la méthode descendante. Nous avons vu que les techniques étudiées permettaient d'obtenir une représentation plus compacte et optimale de la structure arborescente.

Cette étude a permis de valider et tester des connaissances et des solutions sur les arbres binaires par rapport à la littérature. Elle donne certaines solutions et ouvre des pistes pour une poursuite des recherches.

Bibliographie

- [FTÉRICH 04] S. Ftérich, C. Ben Amar. CryptoCompression d'Images Fixes Par la méthode de Quadtree optimisée et AES. *CORESA*, 2004.
- [SHUKLA 02] R. Shukla, P. L. Dragotti, M. Do, M. Vetterli. Improved quadtree algorithm based on joint coding for piecewise smooth image compression. *Proc. IEEE International Conference on Multimedia*, août 2002.
- [SHUKLA 04] R. Shukla, M. Vetterli. Geometrical Image Denoising Using Quadtree Segmentation. *International Conference on Image Processing*, 2004.
- [SALEMBIER 00] P. Salembier, L. Garrido. Binary Partition Tree as an Efficient Representation for Image Processing, segmentation, and Information Retrieval. *IEEE Transactions on Image Processing*, 9(4), avril 2000.
- [SALEMBIER 97] P. Salembier, L. Garrido and D. Garcia. Image sequence analysis and Merging algorithm. In *International Workshop on Very Low Bit-rate Video, VLBV'97, (invited paper)*, 1-8, juillet 1997.
- [GARRIDO 99] L. Garrido, P. Salembier, J.R. Casas, Representing and Retrieving Regions using Binary Partition Trees. *IEEE Int. Conference on Image Processing, ICIP'99*, octobre 1999.
- [BLAFFERT 01] T. Blaffert. Recognition of anatomically relevant object with binary partition trees. *IEEE*, 2001.
- [SERRA 82] J. Serra. Image Analysis and Mathematical Morphology. *Academic Press*, 1982.
- [MEYER 92] F. Meyer. Color Image Segmentation. *Proceedings of the 4th Conference Image Processing and its Applications*, 303-306, 1992.
- [VINCENT 91] L. Vincent, P. Soille. Watersheds in Digital Spaces : An efficient Algorithm Based on Immersions Simulations. *IEEE transactions on PAMI*, 13(16), 583-598, 1991.
- [COSTER 90] M. Coster, J.L. Chermant. Précis d'analyse d'images. *Presses du CNRS, 2^{ime} édition*, 1990.
- [BORSOTTI 98] M. Borsotti, P. Campadelli, R. Schettini. Quantitative evaluation of color image segmentation results. *Pattern recognition letters*, 19, 741-747, 1998.
- [SCHETTINI 93] R. Schettini. A segmentation algorithm for color image. *Pattern Recognition Letters*, 14(6), 499, 1993.