Thèse

Présentée par **Alain PUJOL**
Pour obtenir le grade de Docteur de L'Ecole Centrale de Lyon

Contributions à la Classification Sémantique d'Images

DIRECTEUR DE THESE: **Liming CHEN**

Ecole Doctorale Informatique et Information pour la Société (EDIIS)

Soutenue le 12 juin 2009

JURY

M. Patrick LAMBERT	Professeur des Universités	Polytech'Savoie	Rapporteur
M. Bernard MERIALDO	Professeur des Universités	Eurecom	Rapporteur
Mme. Gabriella CSURKA	Docteur	Xerox XRCE Grenoble	Examinateur
M. Georges QUENOT	Chargé de Recherche	Laboratoire d'Informatique de Grenoble	Examinateur
Mme. Michèle ROMBAUT	Professeur des Universités	Université Joseph Fourier	Examinateur
M. Liming CHEN	Professeur des Universités	Ecole Centrale de Lyon	Directeur de thèse

Remerciements

Je tiens à remercier en premier lieu le professeur Liming Chen pour avoir dirigé mes travaux et permis la réalisation de cette thèse. J'adresse également mes remerciements à Emmanuel Dellandrea et Mohsen Ardabilian pour leurs précieux conseils qui m'ont grandement aidé dans mes travaux, ainsi qu'à Christian Vial pour son aide, sa disponibilité et sa gentillesse en toute circonstance. Je veux aussi témoigner de ma gratitude à Colette Vial, Isabelle Dominique et Françoise Chatelin pour avoir partagé avec moi et dans la bonne humeur de nombreux problèmes, soucis administratifs et autres situations inextricables. Je remercie également toute l'équipe d'enseignement en informatique du département avec qui j'ai eu plaisir à travailler.

J'adresse évidemment un grand merci aux doctorants du laboratoire qui ont croisé ma route, qu'ils m'aient précédé ou suivi : Mohammed Hammami, Boulbaba Ben Amor, Peng Kun, Dzimitri Tishkou, Vyacheslav Parshin, Xiao Zhongzhe, Aliaksandr Paradzinets, Huanzhang Fu, Chu Duc Nguyen, Karima Ouji, Przemyslaw Szeptycki, Xi Chao, Kiryl Bletsko ainsi qu'à tous les étudiants ou stagiaires que j'ai côtoyés qui se sont tous révélé être des collègues très sympathiques.

J'ajouterai un remerciement spécial à l'équipe de rugby de Centrale qui m'a permis de me défouler et de vivre de très bon moments (ainsi que de visiter différents pavillons de l'hôpital Edouard Herriot).

J'adresse également des remerciements tout particuliers aux rapporteurs, les professeurs Patrick Lambert et Bernard Mérialdo qui ont pris le temps de lire et évaluer mes travaux ainsi que pour leurs remarques judicieuses qui m'ont permis d'améliorer ce manuscrit. Je remercie enfin l'ensemble des membres du jury pour l'intérêt qu'ils ont porté à mes travaux.

Je terminerai en remerciant mon épouse Sachi et ma famille qui m'ont toujours soutenu, et ce quoi que je fasse (à l'exception du rugby).

Table des Matières

Table des Illustrations

Résumé

La classification d'images par le contenu visuel est un domaine particulièrement actif et difficile de l'analyse d'images. En n'imposant aucune restriction sur les images traitées, on se retrouve en effet face à un contenu qui peut être composite, ambigu et qui plus est acquis dans de mauvaises conditions. Aussi difficile qu'elle puisse paraître, cette activité pose pourtant très rarement des problèmes à un être humain qui, quelle que soit la complexité de l'image d'origine, parvient toujours très rapidement à une décision.

Idéalement un système d'indexation automatique devrait permettre de rechercher des concepts dans une image hétérogène et de savoir détecter leur présence comme leur absence de manière non-mutuellement-exclusive. Notre objectif a d'abord été de nous inspirer de la performance de la classification humaine pour en tirer des procédés d'analyse nous mettant dans de bonnes conditions pour nous acquitter de cette tâche. Nous avons également déterminé des caractéristiques de forme pertinentes pour nous assister dans la tâche de classification. Enfin nous avons développé une classification efficace qui puisse s'adapter à ces conditions difficiles.

Les contributions de cette thèse portent sur les informations extraites de l'image, le procédé d'extraction ainsi que sur leur utilisation pour accéder à un verdict de classification.

Notre première contribution concerne les informations extraites de l'image. En nous inspirant des principes de la perception humaine, nous voulons exploiter le fait que les informations les plus importantes sont des informations singulières au sein d'une image. Ainsi nous basons notre descripteur de forme sur des segments, information visuellement bien plus significative que les gradients traditionnellement utilisés. Différentes formes de descripteurs sont explorées dans l'optique d'une classification globale (type d'image) comme d'une recherche d'objets visuels.

Notre seconde contribution porte sur la réduction du nombre de couleurs au sein d'une image le but est de simplifier sans l'endommager l'information de couleur afin de pouvoir l'exploiter ultérieurement de manière efficace. En particulier le but est ici de permettre une segmentation efficace de l'image par clustering qui, sans réduction préalable, porterait sur des dizaines de milliers de couleurs. Pour éviter d'endommager l'image, nous avons utilisé la théorie de l'information afin de quantifier l'information qu'apportait une couleur par rapport aux autres. Ceci nous permet ainsi de repérer et de sélectionner des couleurs perceptuellement importantes car singulières.

Notre troisième contribution est la continuation logique de la précédente : il s'agit d'un algorithme de segmentation d'image en régions de couleur homogène. Il s'agit d'un procédé en trois étapes. On effectue une réduction préalable du nombre de couleurs ainsi qu'un filtrage visant à améliorer sa robustesse. On effectue ensuite une détection rapide d'un nombre idéal de couleurs quantifiées suivi par une quantification par le procédé présenté dans le précédent paragraphe. Enfin nous opérons une séparation des régions spatialement disjointes et une fusion des régions trop petites ou trop proches de leurs voisines. Cet algorithme a pour principaux atouts sa robustesse et sa capacité à produire des régions de taille importantes à partir desquelles on peut envisager d'extraire des caractéristiques visuelles.

Notre quatrième contribution est une synthèse des précédentes : nous intégrons les éléments développés dans un système de classification automatique. Nous partons du constat que l'étude de la perception humaine suggère d'une part une domination de la perception globale sur la perception locale et d'autre part met en relief l'importance des relations entre différentes parties d'une image. Nous choisissons donc constituer une plateforme de classification se basant sur des informations extraites à partir de régions déterminées par notre

algorithme de segmentation. Nous utilisons nos caractéristiques basées sur des segments, complétées par des caractéristiques visuelles obtenues à partir de descripteurs existants (couleur, texture…) auxquelles nous ajoutons des informations provenant des régions voisines. Ces informations, collectées sur des images d'entrainement, sont fusionnées pour la constitution d'un "vocabulaire visuel". Toute image est ensuite exprimée à partir de ce vocabulaire et peut dès lors être classée en utilisant un procédé d'apprentissage supervisé.

Abstract

Image indexing based on visual content is an especially active and challenging field in image processing. Without any restriction on processed images, we indeed face contents which may be heterogeneous, ambiguous and also acquired in poor conditions. As difficult as it may appear, most of the time, this activity poses very few problems to human beings who always reach a quick classification decision, whichever the complexity of the original image.

An automated indexing system should, ideally, allow searching for concepts within a heterogeneous image and being able to detect their presence as well as their absence in a non-mutually exclusive way. Our first objective was to draw means of processing information from human perception which would put us into good conditions to make a successful classification. We also devised efficient shape features to help us in this classification task. Finally, we developed an efficient classification process that could adapt to these difficult conditions.

The contributions of this thesis are about the basic features extracted from the image, the extraction process itself as well as the classification process itself.

Our first contribution is related to the information extracted from the image. The principles of human perception lead us to look for singular information as it is deemed as perceptually more important. As a consequence, we based our shape descriptor on line segments which are visually more significant than the usually used gradient values. Several feature structures are proposed aiming both at global classification (type of the image) and visual object categorization.

Our second contribution is about image color reduction; its purpose is to simplify color information without damaging it in order to use it more efficiently. More specifically, our purpose here is to allow efficient color clustering which would otherwise involve tens of thousands of colors. To avoid damaging the original image, we used information theory in order to quantify the amount of information provided by a color compared to the others. This allows us to identify and select singular and therefore perceptually important colors.

Our third contribution is in direct continuation of the previous one: it consists in a color based image segmentation algorithm. It is a three-step process. We first reduce the number of colors and enhance robustness to noise through filtering. We then determine an ideal number of quantized colors followed by the quantization itself using the process introduced in the previous paragraph. We finally separate regions which are not spatially connected and merge regions that are either too small or too similar to their neighbors. This algorithm's assets are mainly its robustness as well as its ability to produce large coarse regions from which we can extract visual features.

Our fourth contribution is a synthesis of the previous ones: we integrate the elements we devised into an automated classification system. Basing on the study of human perception, we observe the precedence of global perception over local perception as well as the importance of the relationship between neighboring parts of an image. We therefore chose to build a classification platform built from information extracted from coarse regions provided by our segmentation algorithm. We use our line segment features combined to other visual characteristics provided by existing features (such as color, texture…) to which we add information from neighboring regions. This information, collected on a training set, is gathered to build a "visual vocabulary". All images are then described through this vocabulary and can be therefore classified using a supervised learning process.

Chapitre 1: Introduction

1. Contexte et objectif

La production d'images numériques a connu un essor considérable ces dernières années, tant chez les professionnels de l'image que chez les amateurs avec la démocratisation des appareils de photo numériques. Les capacités de stockage sur les divers supports (disques durs, cartes SD, cartes CF, DVD…) repoussent de plus en plus loin les limites de quantité possible de prises de vues. Devant un tel contexte d'explosion de contenu numérique se pose inévitablement le problème de l'organisation et de la recherche. A l'heure actuelle les bases d'images sont encore majoritairement indexées manuellement, via l'association de métadonnées sous la forme de mots clés et autres informations circonstancielles sur la prise de vue. Ces annotations, quoique fiables, ne sont pas exemptes de défaut. Le principal défaut étant précisément le volume de données lui-même, qui rend la tâche particulièrement longue et fastidieuse. Ce problème décuple l'impact de tous les autres : les informations ne sont à priori annotées qu'en un nombre limité de langues, elles doivent aussi se conformer à un format dont toute modification entraîne une nouvelle tâche d'annotation. Il y a aussi la notion de subjectivité qui rend la tâche d'annotation pas forcément reproductible car deux annotateurs différents ne produiront pas systématiquement la même annotation pour une même image.

C'est donc tout naturellement qu'a émergé la recherche pour une solution informatique au problème. C'est ainsi qu'ont rapidement fleuri, d'un côté des systèmes de recherche par le contenu (dits systèmes Content Based Image Retrieval, ou CBIR) et d'un autre des systèmes d'indexation automatique. Si ces deux mécanismes diffèrent par leur approche, les techniques mises en œuvre sont très similaires. Nous aborderons pour notre part le problème sous l'aspect "indexation automatique" mais nous étudierons bien évidemment les systèmes CBIR avec intérêt au nom de ces nombreuses similarités.

L'objectif général de cette thèse vise donc à développer des solutions afin de réduire le "fossé sémantique" qu'évoquent beaucoup de travaux dans la littérature. Il s'agit du fossé qui sépare la "sémantique" de l'acquisition de données à partir des pixels de l'image. Le terme de sémantique est utilisé un peu abusivement : on entendra ici simplement "caractérisation de l'image ou de son contenu par des concepts de haut niveau". Intuitivement, il s'agit d'étiqueter les images automatiquement par des concepts que l'on perçoit dans celles-ci : personne, train, montagne, paysage, etc. Nous souhaitons donc aller au-delà des travaux sur la recherche par similarité visuelle d'images tels qu'ils ont été développés dans les années 1990 [1], [2].

2. Problématique, notre approche

Le problème est donc le suivant : on sait extraire des éléments qui caractérisent, par exemple, la couleur, la texture, la forme générale, mais il ne s'agit que d'informations mathématiques. A partir de celles-ci, il apparaît particulièrement difficile de remonter à des éléments qui aient un sens et surtout un intérêt pour une personne qui souhaiterait que la machine lui décrive tout ou partie de l'image. On cherche donc, étant donnée une image numérique, à pouvoir donner des informations comme "Cette image contient des personnes et un avion" ou un peu plus générales comme "Il s'agit d'une photo de montagne". Il s'agit donc d'un problème typiquement de vision par ordinateur sous sa forme probablement la plus difficile. En effet, les images qu'il faut analyser ne subissent aucune contrainte, et les

concepts, comme par exemple montagne ou paysage, véhiculés dans une image, peuvent avoir des formes, des angles de vue ainsi que des conditions d'éclairage totalement libres. Les meilleurs résultats dans la compétition internationale Pascal [3] ne reportent qu'un taux de précision de 50% environ sur 20 classes alors qu'il existe plus de dizaine de milliers de concepts répertoriés dans le site "wordnet" [4].

Dans ce travail, nous nous sommes intéressés à deux aspects du fossé sémantique : la rive mathématique où l'on va formaliser des données pertinentes et porteuses de sens ainsi que son franchissement par la traduction desdites données en un ou plusieurs concepts présents dans l'image. Alors que la majorité des travaux de la littérature tente d'appliquer une approche statistique du type "bag-of-features" ([5], [6], [7], …), une approche qui a été empruntée de celle de "bag-of-keywords" pour la caractérisation du contenu textuel, nous proposons de suivre ici des mécanismes de perception humaine dans l'interprétation automatique de l'image, en occurrence des principes simples issue de la théorie "Gestalt". Notre approche est donc fondée d'une part sur la théorie de la perception humaine et d'autre part les techniques existantes dans le domaine du traitement d'images en général.

Après une courte introduction sur la perception humaine de l'image, nous décrivons rapidement dans la suite les trois problèmes fondamentaux d'une vision par ordinateur, la représentation, l'apprentissage et la reconnaissance, afin de mieux situer nos contributions.

2.1. L'inspiration de la perception humaine

Si on se pose le problème de la perception de l'information visuelle par l'homme afin d'y trouver d'éventuelles sources d'inspiration pour la vision informatique, on se retrouve confrontés d'une part à la médecine pour l'aspect "mécanique" et d'autre part aux sciences cognitives pour l'aspect analyse de l'information et raisonnement.

L'aspect "mécanique" et en particulier le fonctionnement de l'œil a été assez largement couvert pour le développement de certaines caractéristiques à extraire de l'image. On peut principalement citer la présence de récepteurs sensibles à l'orientation qui sont à la base de beaucoup de travaux sur des filtres sélectifs, par exemple la transformation en ondelettes de Gabor qui modélise le comportement des cellules du cortex visuel primaire [8]. Il faut cependant noter que la plupart des caractéristiques décrivant la forme ont tendance à privilégier la fiabilité d'extraction. Ainsi celles-ci sont plutôt établies pour correspondre à un formalisme mathématique robuste plutôt que ressembler à un mécanisme biologique plus délicat à modéliser. Enfin dans ce type d'approches, on peut également noter les travaux sur des descripteurs et des traitements inspirés du système visuel humain dans le domaine de la vidéo ainsi que de la détection et du suivi temporel [9].

L'aspect de l'interprétation est moins fréquemment abordé. Il faut dire que les mécanismes de l'analyse sont beaucoup plus délicats à observer. Il existe toutefois quelques travaux qui font référence aux théories de Gestalt ([10], [11], …), et c'est dans cette direction que nous nous orienterons pour développer nos travaux en cherchant à nous inspirer plus globalement des différents aspects de la perception humaine lorsque cela est possible. Nous aborderons dans le chapitre 2 quelques aspects de la vision humaine et les apports que nous envisageons dans nos travaux.

2.2. Représentation des caractéristiques

La représentation consiste à passer de la seule chose dont on dispose (une matrice de pixels de couleur) à une information plus pertinente pour une analyse ultérieure du contenu de

l'image. Construire ces descripteurs de "bas niveau" constitue la première étape et la base du travail, le défi ici étant d'extraire des caractéristiques de taille raisonnable, discriminantes et efficaces. Examinons une simple photographie (Figure 1) que l'on chercherait à catégoriser de manière globale. Un humain va immédiatement percevoir les informations essentielles de cette photo que sont la route et les deux rangées d'arbres.

Figure 1: Exemple d'image à caractériser

La première question que l'on se pose est de savoir d'où on va extraire une information plus ou moins pertinente pour la caractérisation du contenu visuel. En effet on trouve 4 types d'approches dans la littérature :

- Approche globale : des données provenant de toute l'image sont analysées
- Approche par régions homogènes : l'image est décomposée en régions selon un critère d'homogénéité et les données sont extraites au sein de chaque région
- Approche par régions fixes : l'image est décomposée en régions de taille fixe (blocs)
- Approche locale par points d'intérêt : des points dits "points d'intérêt" sont produits sur l'image. Les données sont extraites dans un petit voisinage autour de ces points.

L'approche par image complète se base sur l'exploitation statistique d'éléments extraits pixel par pixel. Elle suppose implicitement que la totalité de l'image est reliée à l'objet de la recherche ou de l'index. Tout objet incohérent introduirait du bruit dans les caractéristiques. Cette limitation incite de fait à se tourner vers des méthodes plus locales. On notera que l'intérêt pour la localisation spatiale des caractéristiques est présent depuis le début. Ainsi les travaux de Swain et Ballard en 1991 [12], qui peuvent être considérés comme les pionniers de l'indexation et la recherche d'image par le contenu, utilisent déjà des caractéristiques localisées avec leur descripteur de couleur "histogram backprojection".

Les approches par points d'intérêt et par régions fixes s'opposent aux approches globales et par régions homogènes au sens où elles utilisent un voisinage de taille et de forme prédéfinie. De ce fait la localisation se fait de manière simple et prévisible. Les régions ont toutes la même taille et de ce fait les caractéristiques sont homogènes entre elles. De plus, s'intéresser à de petites régions revient à décomposer l'image en de petits éléments qui seront moins susceptibles d'être partiellement masqués. En effet un objet qui serait composé d'une seule région verrait cette région changer de manière importante en cas d'occlusion partielle alors qu'un objet composé d'un grand nombre de régions en verrait certaines disparaitre mais d'autre demeurer telles-quelles et donc encore simples à apparier. On peut également objecter que la taille et la forme des régions ne s'adapte pas au contenu de l'image. A ce titre l'observation de l'image à plusieurs échelles et en particulier la notion de "scale-space" [13] a été plus récemment introduite pour permettre de définir une échelle appropriée pour un point donné et ainsi adapter un peu la région au contenu. La méthode d'extraction par des points d'intérêt est actuellement "à la mode" mais le fait que les points ne soient pas reliés entre eux nous prive potentiellement des relations de voisinage si importantes à l'analyse humaine.

Dans nos travaux, Nous suivons une approche par régions qui trouve une justification perceptuelle et apporte une solution à cette nécessité de localisation. De plus le critère d'homogénéité garantit que les pixels d'une région constituée auront un minimum de cohérence entre eux sur le plan perceptuel. L'extraction de caractéristiques peut se faire de manière similaire à celle de l'image globale en la pratiquant simplement sur les pixels de la région. Nous aborderons la problématique de la segmentation en régions de manière plus détaillée dans le chapitre 4. Néanmoins, la production de régions est toutefois un exercice non seulement délicat mais aussi couteux en temps et en ressources machine. L'apport d'une segmentation dans la catégorisation d'images par rapport à l'approche populaire "bag-of-features" de la littérature est aussi étudié dans cette au chapitre 6.

Une fois la région d'extraction déterminée, restent à déterminer les caractéristiques que l'on va extraire. On va essentiellement en trouver de trois sortes : *forme, texture et couleur*. Si on retourne à la Figure 1, on peut essayer de dégager des indices dans toutes ces catégories. On a tout d'abord des couleurs qui sont assez représentatives si on veut représenter la route, les troncs ou les feuilles. On a ensuite les descripteurs de forme qui, dans le cas présent, seraient particulièrement informatifs. En effet les lignes verticales présentes dans les arbres tout comme les lignes de fuite de la route sont assez informatives. Enfin les textures de feuilles et de tronc sont intéressantes pour identifier les rangées d'arbres. Ces caractéristiques peuvent être représentées de nombreuses façons : le choix de représentation de l'image se fait non seulement par un choix de caractéristiques à capturer mais également par l'application d'un formalisme mathématique qui permettra d'exprimer au mieux l'information. Ces deux opérations (extraction puis modélisation) induisent une perte de données, que ce soit par simple omission (on n'a pas extrait suffisamment d'éléments pour décrire exhaustivement) ou par approximation, les caractéristiques extraites devant être relativement compactes (comme on le verra dans la section suivante). Lorsqu'on construira un descripteur nous devrons contrôler cette perte d'information afin de présenter les meilleures performances possibles. Différentes techniques de représentation seront présentées en détail dans le chapitre 3, et nous présenterons notre contribution dans ce domaine dans le chapitre 5.

2.3. L'apprentissage et la classification

La catégorisation d'images par la machine étant un problème de vision par ordinateur, elle a besoin d'un apprentissage. Les difficultés ici sont bien connues. D'abord, la rareté de données d'apprentissage, c'est-à-dire les images annotées, par rapport à l'extrême variabilité d'apparences d'objets ou concepts qu'il faut identifier. Cette rareté de données d'apprentissage est encore compliquée par un déséquilibre criant entre les données d'apprentissage. En effet, pour une catégorie donnée, par exemple voiture, le nombre d'exemples négatifs l'emporte largement sur les exemples positifs, ce qui peut complètement fausser la frontière de séparation de classe.

Ensuite, on se retrouve vite confronté problème bien connu de la "curse of dimensionality" [14]. En effet, on pourrait être tenté de regrouper un maximum d'indices visuels (couleur, texture, contours…) décrits d'une variété de façons et de manière aussi précise que possible. On disposerait ainsi de toutes les informations nécessaires pour traiter le problème efficacement. Une caractéristique étant représentée par un ensemble de n valeurs numériques, elle est un point dans un espace de n-dimensions. Or, la croissance de n entraine une croissance exponentielle du nombre de points nécessaires pour décrire l'espace. On

démultiplie donc le nombre d'exemples nécessaires pour pouvoir bien décrire une catégorie ou un objet. Ceci nous contraint donc à utiliser des caractéristiques compactes.

Enfin, un autre problème majeur à éviter est celui de l'overfitting. L'apprentissage doit être fait pour ne pas trop "coller" aux données d'apprentissage afin d'avoir une bonne capacité de généralisation sur des données non vues. Ceci se fait différemment selon le type d'algorithme utilisé (ex : taille de la couche cachée pour un perceptron multicouches) mais c'est quoiqu'il arrive un problème que l'on se devra de garder à l'esprit. Notre modèle de classification sera présenté dans le chapitre 6.

Une particularité du problème de catégorisation d'images est la caractérisation du contenu visuel à travers les descripteurs. En effet, alors que le nombre de descripteurs de bas niveau, couleur, forme ou la texture, est variable d'une image à une autre, les machines d'apprentissage doivent travailler avec des vecteurs de caractéristiques de taille fixe. Dans nos travaux, nous utilisons une approche assez classique qui consiste à apprendre un " vocabulaire visuel" à partir des données d'apprentissage. Nos contributions ici sont la fusion de plusieurs types de vocabulaires visuels pour l'amélioration de performances de classification.

3. **Nos contributions**

Nous avons évoqué globalement les différentes parties du procédé d'analyse d'images que nous allons mettre en œuvre ainsi que les principales problématiques auxquelles nous nous confrontons. Sur un point de vue pratique, le cadre que nous nous sommes fixés pour notre approche a deux applications : l'indexation automatique d'images dans des catégories assez larges (photo de montagne, de forêt, etc.) et la détection de la présence d'objets au sein d'images. Cette dernière se fera par l'utilisation de nos algorithmes sur la base d'images du "challenge pascal" [3].

Notre approche est avant tout une approche motivée par la perception humaine, nous en évoquerons les principaux principes qui nous serviront de ligne directrice dans notre travail au chapitre 2.

Notre premier axe de travail a concerné la couleur et en particulier la décomposition de l'image en régions de couleur homogène. A ce titre nous avons apporté deux contributions : la première est un algorithme de réduction du nombre de couleurs dans l'image mettant l'accent sur l'optimisation de la conservation de l'information de couleur telle qu'elle est perçue. Cet algorithme produit une erreur quadratique de quantification (MSE) inférieure aux algorithmes de l'état de l'art. La seconde est un algorithme de segmentation en régions se basant sur cette réduction ; il suit également les principes Gestalt et apporte, par rapport aux algorithmes de l'état de l'art une robustesse nécessaire pour traiter automatiquement une grande quantité d'images non contraintes.

Notre troisième contribution se situe au niveau des descripteurs utilisés. Toujours en relation avec les théories de la perception, nous cherchons à extraire de manière fiable des données de forme plus informatives que des données à l'échelle du pixel actuellement utilisées. En se basant sur la détection de segments pour l'analyse des lignes de fuite développée par Mohsen Ardabilian [15], nous avons développé une série de descripteurs de forme qui se sont avérés efficaces pour la caractérisation de contenu visuel. Les tests que nous avons réalisés avec différents mécanismes de classification que nous avons développés montrent des performances de classification supérieures aux populaires descripteurs SIFT.

Notre dernière contribution se trouve dans la classification. A l'aide des composants que nous avons développé et en suivant toujours notre ligne directrice constituée par les théories de la perception humaine, nous mettons en place une classification basée sur la notion de "vocabulaire visuel". Nous introduisons un histogramme construit sur la notion d'appartenance floue d'un vecteur de caractéristiques aux "mots" d'un vocabulaire construit pour chaque type de caractéristiques. Cette méthode de classification produit des résultats consistants sur la difficile base Pascal VOC 2007 [3].

4. **Organisation de la thèse**

Dans le chapitre 2 nous évoquerons brièvement les théories de la perception humaine, en nous attardant plus particulièrement sur les aspects interprétation d'images et perception de la couleur qui ont influencé nos travaux.

Nous poursuivrons ce travail sur l'existant dans le chapitre 3 par un état de l'art sur les techniques de base d'extraction de caractéristiques ainsi que sur les algorithmes de clustering et les métriques utilisées qui sont des problèmes récurrents dans le contexte de l'analyse d'images.

La suite de ce manuscrit évoquera nos contributions : dans le chapitre 4 nous détaillerons nos travaux sur la décomposition de l'image en régions. Nous présenterons nos deux contributions dans ce domaine, respectivement sur la réduction du nombre de couleurs et sur la segmentation. Le chapitre 5 introduira nos contributions en matière de descripteurs basés sur la détection de segments. Nos dernières contributions, dans le domaine de la classification, seront abordées dans le chapitre 6.

Finalement nous conclurons à partir des résultats de nos travaux et discuterons des perspectives ultérieures de travail dans le chapitre 7.

Chapitre 2: L'interprétation humaine

Comme nous l'avons évoqué dans l'introduction, nous avons voulu que nos travaux sur la classification d'images soient guidés par des aspirations de perception humaine. Aussi, nous décrivons dans ce chapitre les principes fondamentaux d'une part sur la perception humaine d'après la théorie de Gestalt dans la section 1, et d'autre part sur la perception et la représentation de la couleur dans la section 2 qui joueront le rôle de repères d'orientation dans le développement de notre approche.

1. Théories sur la perception humaine

1.1. Principes de la théorie "Gestalt"

Nous avons mentionné dans l'introduction qu'un certain nombre de papiers faisaient référence aux principes "Gestalt". Cette théorie fournit en effet une inspiration intéressante pour l'analyse d'image qui pourrait potentiellement renforcer notre système classification. Elle a pour principe qu'un ensemble perçu est différent de la somme de ses parties et fournit diverses lois pour relier des éléments adjacents de l'image. C'est un point important car beaucoup d'algorithmes actuels ont tendance à extraire des éléments isolés, nous y reviendrons lors de notre discussion sur la classification au chapitre 6. Une série de principes de base de gestalt sont énoncés sous forme de "lois" décrites dans [11] :

- La loi de la bonne forme : un ensemble de parties quelconques tend à être perçu d'abord comme une forme, cette forme se veut simple, symétrique
- La loi de bonne continuation : un ensemble de parties ont tendance à être perçues dans une continuité, comme des prolongements les uns par rapport aux autres.
- La loi de la proximité : nous regroupons d'abord les parties les plus proches les unes des autres.
- La loi de continuité : nous regroupons ensuite les plus similaires (couleur, forme, …) entre elles pour percevoir une forme.
- La loi de destin commun : des parties en mouvement ayant la même trajectoire sont perçues comme faisant partie de la même forme.
- La loi de clôture : une forme fermée est plus facilement identifiée comme une figure (ou comme une forme) qu'une forme ouverte.

Plus exemple on peut évoquer les principes de similarité qui provoquent le regroupement récursif d'éléments (appelés "Gestalts partiels") pour former un élément complet ("Gestalt global"). Ce phénomène est illustré par la Figure 2 : on y voit plusieurs éléments que la perception va apparier récursivement : on va regrouper d'abord les formes similaires en 4 gestalts partiels avant d'associer ces gestalts partiels en fonction de la couleur (et de leur alignement pour finalement reconnaitre les deux branches d'un X.

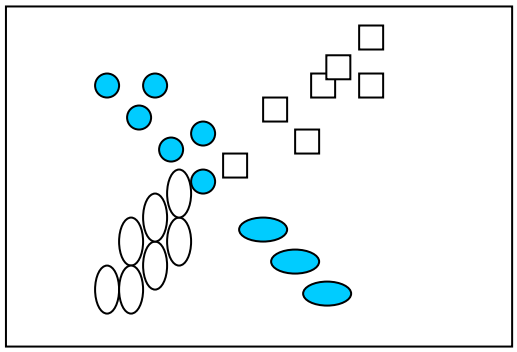


Figure 2: Exemples de gestalts partiels et de gestalt global

Ceci n'est qu'un exemple d'utilisation successive de quelques lois Gestalt (ici proximité, continuité de couleur et de forme et "bonne continuation"). On peut aussi remarquer un petit conflit qui montre que ces simples lois ne sont pas triviales à interpréter. Le carré au centre du "X" pose problème : en effet, il est aligné avec deux ronds bleus et plus proches d'eux que des autres carrés. Les lois de continuité de couleur et de forme rentrent en conflit avec la loi de proximité. Dans notre cas, ceci ne change toutefois rien au Gestalt global. Mais on peu trouver des cas d'images bi-stables où on a deux perceptions possibles mais qui ne peuvent coexister.

1.2. Prépondérance du global sur le local

Un autre cas intéressant est celui de l'interférence entre la perception locale et la perception globale. L'exemple illustré Figure 3 montre clairement la prépondérance du global sur le local : l'identification des petites lettres est perturbée par les grandes lettres mais pas l'inverse.

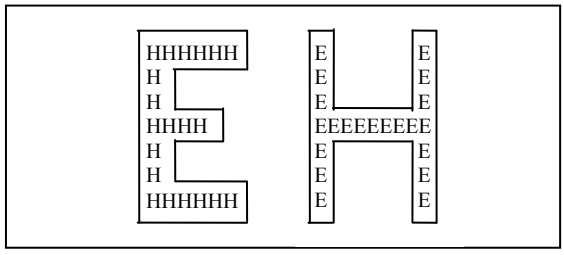


Figure 3: Prépondérance du global sur le local

Ce phénomène est observé par D. Navon dans ses travaux [16], ceux-ci insistent également sur le caractère inévitable de l'interprétation globale : on ne peut se forcer à ignorer l'aspect global au profit du local.

En se basant sur ces études, dans notre travail nous cherchons à imiter ce procédé en décomposant l'image en grosses régions homogènes construites par agglomérations successives, puis en étudiant leurs caractéristiques internes tout en intégrant des informations de voisinage.

1.3. Associations de concepts

Par ailleurs, on peut aussi se référer au modèle de "Schémas" introduit par Piaget, la connaissance est définie comme un ensemble de "schémas" de propriétés définies soit par d'autres schémas soit par des propriétés élémentaires. La phase de reconnaissance est ainsi l'appariement de l'objet détecté à un schéma (le terme objet est ici à prendre au sens large : il peut aussi bien désigner un élément visuel concret et identifiable qu'un ensemble abstrait de propriétés). Le quiproquo, la mauvaise interprétation, naissent d'un mauvais appariement dû à des propriétés communes mais aussi a une prépondérance d'un schéma par rapport à un autre : le schéma le plus couramment utilisé sera en effet appliqué préférentiellement.

On formule alors l'idée de tracer le parallèle entre les propriétés que l'on vient d'évoquer et des valeurs caractéristiques qu'une machine extrairait de ce que l'on souhaite caractériser. Ceci nous conduit à construire un ensemble de valeurs singulières, chaque objet à caractériser étant représenté par un vecteur de caractéristiques considéré comme une combinaison de valeurs singulières issues de cet ensemble. C'est un principe que nous nous proposons de suivre dans notre travail : d'une manière imagée l'ensemble de valeurs singulières constituerait un vocabulaire qui serait utilisé pour décrire des images et qui représenterait les concepts que l'on peut associer à un objet. Un objet serait ainsi reconnu grâce aux éléments (mots du vocabulaire) dont il est proche. On peut enfin noter, dans le cadre de cette analogie, que l'on ne se limite pas à des éléments sémantiques. Une combinaison non identifiée d'éléments de texture peut tout à fait constituer un élément caractéristique, même si aucun langage naturel ne prévoit pas de mot pour définir cet élément. Ce principe sera développé dans le chapitre 6 sur la classification.

1.4. Expériences

Nous nous sommes également interrogés sur la décision de classification elle même. On se pose la question des facteurs qui interviennent sur la décision de classer une image dans une catégorie plutôt qu'une autre en cas d'ambigüité. Ici, l'analyse du comportement de l'humain pour nous guider dans notre approche peut apporter un plus intéressant sous la forme d'heuristiques ou d'informations à priori par rapport à des méthodes génériques de classification utilisées seules. Nous avons eu la chance de participer à une étude dans le cadre du stage d'un étudiant en sciences cognitives sur le procédé de décision de classification ainsi que sur l'importance de quelques éléments dans cette décision [17]. Cette étude, ainsi que les discussions qui l'ont accompagnée, ont révélé un certain nombre d'aspects intéressants. Un premier élément est venu du principe d'analyse fréquentielle : il ressort que la première étape de l'interprétation d'une image se fait en basses fréquences spatiales, ce qui signifie qu'avant d'analyser précisément le contenu d'une scène on observe un ensemble de formes grossières. Ceci motive donc particulièrement une approche basée sur une décomposition préalable de l'image en régions. Par ailleurs cette étude a aussi démontré plus précisément l'importance de la position des objets au sein de l'image et permis de quantifier l'impact de la proportion d'un objet dans une image. Tout ceci renforce l'importance que l'on peut accorder à l'analyse spatiale ainsi que l'importance de la position absolue d'une région. On retrouve dans ces expériences la relation privilégiée entre un objet et son entourage ainsi que l'énoncé qu'un ensemble perçu est différent de la somme de ses parties.

1.5. Discussion

Transposer les principes de la vision humaine à l'informatique est un problème délicat et dont on peut même s'interroger de la pertinence. Certes la modélisation de ce qui existe et fonctionne est un bon moyen de parvenir à ses fins mais encore faut il que l'on comprenne

parfaitement les mécanismes que l'on souhaite imiter (ce qui est loin d'être le cas en ce qui concerne la vision et l'interprétation des images perçues) et que l'on sache les reproduire par des opérations prenant un temps fini. On peut, dans ce sens, fort justement objecter qu'il n'existe pas plus d'avions qui battent des ailes que d'oiseaux à hélice et qu'il est ainsi nécessaire de savoir s'affranchir de l'imitation du vivant. Une conclusion des travaux menés dans [10] est spécifiquement que l'inspiration par la perception humaine. L'imitation définit donc ici une approche par défaut qui va constituer une base de travail solide pour nous fournir des mécanismes, des principes généraux à explorer, évaluer et, éventuellement, à suivre ou adapter.

2. Perception et représentation de la couleur

Le traitement de l'information de couleur est un domaine ou la perception humaine est particulièrement importante. En effet ici on recherche non seulement la caractérisation de couleurs mais aussi et surtout la possibilité de les comparer. Il est dès lors d'un intérêt évident que cette comparaison soit aussi conforme que possible à la perception humaine.

2.1. Perception de la couleur

Le codage le plus basique qui soit est celui sur trois canaux : rouge, vert et bleu (RGB) et présente l'avantage d'être directement relié à la façon dont les couleurs sont synthétisées sur la machine. Ce système repose directement sur la trivariance visuelle établie par des études menées au XIX^{e} siècle qui établissent la combinaison de trois canaux RGB comme étant une source suffisante pour produire toutes les couleurs observables par un humain. Celle-ci sera confirmée expérimentalement en 1965 avec l'étude des récepteurs dans l'œil humain dont des résultats sont présentés sur la Figure 4. On y distingue clairement trois types de récepteurs (cônes), qui ont également été observés quantitativement. 64% des cônes sont sensibles au rouge, à peu près 32% sont sensibles au vert, et seulement 2% sont sensibles au bleu. La forme des différentes courbes est obtenue par la mesure de l'absorption des longueurs d'ondes par les cônes, mais leurs hauteurs relatives ont été représentées comme égales par manque de données détaillées. On a vu qu'il y avait moins de cônes bleus mais la sensibilité globale au bleu est comparable aux autres, ce qui laisse à supposer un mécanisme qui en améliore la perception. On sait par expérience que les sensibilités à ces trois couleurs sont comparables, mais le procédé détaillé d'assimilation à partir de l'inégale population des cônes n'est pas connu.

Le fait de s'intéresser à la perception des couleurs par l'œil permet de souligner que l'information de couleur n'est malheureusement pas simple à modéliser. Ainsi cette perception dépend non seulement de la nature de l'objet observé mais aussi de la nature de l'illuminant (le plus souvent inconnu) et du capteur (ici l'œil humain). On peut rajouter à cela l'interaction de l'environnement de l'objet observé avec le phénomène de perception et on a une idée de la complexité de la tâche qui consiste à étudier des couleurs ; on rentre alors de plein pied dans la science à part entière qu'est la colorimétrie.

La Commission Internationale de l'Eclairage (CIE) [18] s'attache à définir des standards et des mesures pour ce domaine. Elle a, entre autres, étudié et normalisé certains espaces d'expression de couleurs. En particulier, c'est elle qui a normalisé le système RGB, tel que nous l'utilisons. Par la suite des espaces de représentation des couleurs permettant de mesurer plus précisément des différences ont été établis et ont fait l'objet d'une norme. Signalons enfin un modèle se basant plus précisément sur la perception humaine (LMS pour Long Medium et Short qui représentent les longueurs d'ondes traitées) mais qui n'est

malheureusement pas encore normalisé. Nous allons étudier les espaces de représentation que nous pouvons utiliser pour les couleurs générées par l'ordinateur pour les images que nous voulons traiter. Notre ambition est de déterminer des espaces adéquats pour que nos algorithmes puissent distinguer les couleurs que les humains distinguent et considérer comme proches des couleurs perceptuellement similaires pour un humain.

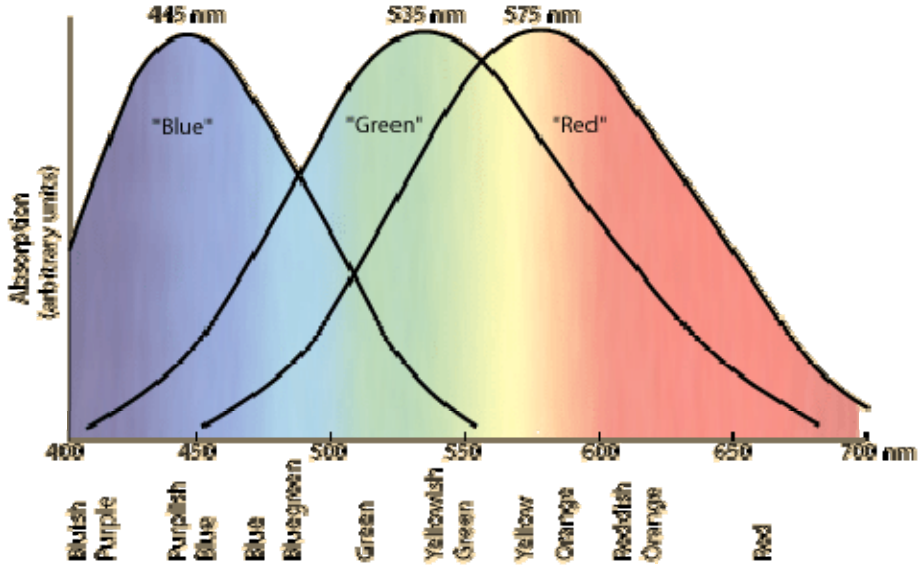


Figure 4: Récepteurs visuels chez l'homme

2.2. Espaces de couleur

Comme nous venons de le voir différents espaces de couleur visant à approcher autant que possible la perception humaine ont été définis voire normalisés ; il existe aussi d'autres espaces conçus par rapport à des besoins industriels (YUV, YCbCr, etc.). Comme tous les espaces que nous analyserons, ils respectent le principe de la trivariance visuelle que nous venons d'évoquer à l'exception notable de l'espace CMYK (Cyan, Magenta, Yellow, blacK) qui a été conçu pour représenter les différentes encres utilisées pour l'impression. Ces espaces permettent une meilleure représentation eu égard au système qui les utilise. Notre but est d'avoir un espace de couleur qui constitue un bon compromis entre son *homogénéité* par rapport à la perception humaine et les calculs nécessaires à l'expression de la couleur dans cet espace. Par homogénéité on entend le fait que deux couleurs distantes d'un certain écart seront, selon leur nature, tantôt confondues, tantôt distinguables. De ce fait nous n'étudierons pas les espaces de couleurs industriels dans cet état de l'art et nous nous concentrerons sur les espaces correspondant à notre problème : RGB, HSV/HSL, puis les espaces perceptuels CIELuv, CIELab et leur dérivé CIELch.

2.2.1. L'espace RGB

L'espace RGB est, comme nous l'avons dit un espace qui est très proche de la source de production. Il quantifie les intensités de Rouge, Vert et Bleu qui permettent de produire, par additivité, n'importe quelle autre couleur. On peut remarquer brièvement que cet espace n'est pas unique au sens où il dépend des teintes de rouge vert et bleu choisies de telle sorte que leur mélange additif produise un blanc de référence. Ainsi existe-t-il par exemple des espaces distincts pour la télévision (NTSC, EBU) ; on considèrera donc l'espace RGB

normalisé par la CIE évoqué dans l'introduction et qui constitue celui auquel nous serons confrontés en informatique. Celui-ci présente comme avantage principal celui d'être l'espace utilisé pour coder les images couleur en mémoire. C'est donc cet espace qui produira les meilleurs temps de calcul si l'application tolère ses approximations. En effet, cet espace n'est pas satisfaisant pour pouvoir mesurer des distances. En effet il présente un certain nombre d'inconvénients avec en particulier une forte corrélation entre les composantes R, G et B. De fait il existe également de nombreuses redondances au sein de cet espace. Un exemple de comparaison de couleurs similaires qui produit pourtant une distance euclidienne assez forte est donné Figure 5. Il ne s'agit que d'un exemple parmi d'autres mais il illustre assez bien le manque d'homogénéité de cet espace.

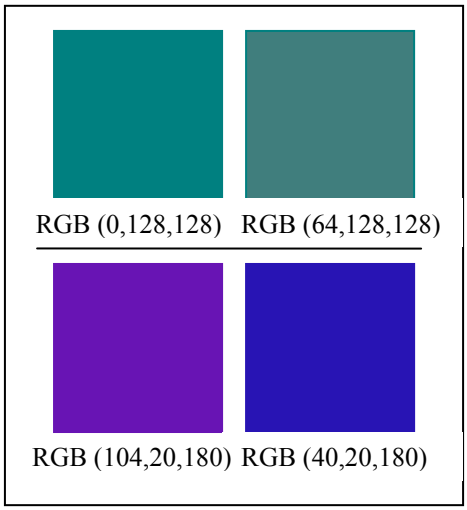


Figure 5: Codage RGB de deux paires de couleurs séparées par la même distance euclidienne

Notons enfin que le codage RGB est dépendant du système qui le produit et qu'il ne prend pas en compte la notion d'illuminant.

2.2.2. Les espaces de type HSV/HSL

Ces espaces sont obtenus à partir de RGB par une transformation qui vise à dissocier l'information de couleur en trois canaux plus informatifs que rouge vert et bleu. Pour reprendre deux acronymes très utilisés, ces espaces de couleur dissocient donc la teinte (H, pour hue), la saturation (S) et une troisième composante qui sera une composante de luminance (L) ou une valeur (V) qui quantifie l'intensité. Plus précisément nous reprendrons les définitions de ces composantes données par l'AFNOR et reproduites dans [19] :

- La luminosité est *l'attribut de la sensation visuelle selon lequel une surface éclairée par une source lumineuse déterminée pourrait émettre plus ou moins de lumière. C'est aussi le correspondant psychosensoriel (approximatif) de la grandeur photométrique « luminance lumineuse »*
- La teinte, c'est *l'attribut de la sensation visuelle qui a suscité des dénominations de couleur telles que bleu, vert, jaune, rouge, pourpre, etc.* Eu égard de la nature de la lumière, ceci correspond à la longueur d'onde prédominante perçue par le récepteur.
- La saturation est enfin définie comme *l'attribut de la sensation visuelle permettant d'estimer la proportion de couleur chromatiquement pure contenue dans la sensation totale*. Elle correspond à l'aspect plus ou moins "dilué" ou "délavé" de la couleur.

Ces descriptions, outre leur mérite de préciser le contenu des trois canaux, sont intéressantes par la présence de l'expression "sensation visuelle" qui est en effet un atout de ce type de représentation : par rapport à une représentation de type RGB, celle-ci permet beaucoup mieux de décrire une couleur selon des critères perceptuels.

Une première difficulté liée à ce format est l'absence de standard en ce domaine. On compte un très grands nombre de formats différents construits sur ce modèle, chacun calculant ses composantes d'une manière qui lui est propre. Dans leur ensemble, on peut leur noter quelques défauts : d'abord le fait que la teinte soit exprimée comme un angle, ce qui rend l'élaboration de distances plus délicate. Ensuite il faut noter que lorsque la luminance et/ou la saturation tendent vers 0, la valeur de la composante "teinte" perd toute importance, ce qui, là encore, rend toute mesure de distance délicate. Enfin ces espaces ne corrigent pas vraiment le manque d'homogénéité de l'espace RGB et restent dépendants du système qui produit la couleur.

2.2.3. Les espaces CIELuv, CIELab et CIELch

Ces espaces sont précisément le résultat d'une recherche d'homogénéité. Cette notion déjà évoquée plus haut se visualise simplement au moyen des ellipses de MacAdam. Ces ellipses, définies expérimentalement pour un individu, correspondent à des zones d'un espace de couleur au sein desquelles le sujet ne parvient pas à distinguer deux couleurs. Ces ellipses sont illustrées sur la Figure 6.

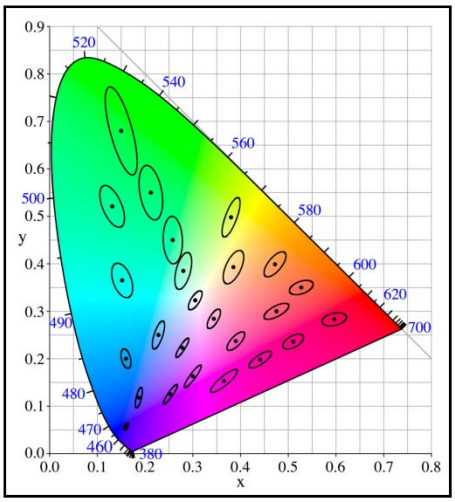


Figure 6: Exemples d'ellipses de MacAdam pour un observateur donné, dans un espace de couleurs donné

L'homogénéité se traduirait donc par la transformation de ces ellipses en cercles de diamètre constant.

Les espaces CIELAB et CIELUV sont le fruit d'études parallèles et partagent cet objectif d'homogénéité. Ils font tous deux intervenir un espace de transition : l'espace CIEXYZ et nécessitent la spécification de conditions d'acquisition (illuminant). La conversion de l'espace RGB vers l'espace CIEXYZ se base sur des valeurs uniformisées de RGB afin de rendre ces valeurs indépendantes du système qui les produit. On doit également spécifier un illuminant et un observateur de référence ce qui peut poser un problème de précision quand on désire traiter des photos sans la connaissance de ces conditions. A partir de là les deux espaces LAB et LUV sont similaires dans leur construction : un canal "L"

définissant la luminance sur lequel sont situés les niveaux de gris et deux canaux définis à luminance constante basés sur des oppositions de couleur (rouge et vert pour l'axe "A-B", bleu et jaune pour l'axe "U-V"). A partir de ces deux espaces, on peut construire une représentation sur le modèle des espaces HSV/HSL décrits précédemment en les exprimant tout simplement en coordonnées cylindriques (1).

$$\begin{aligned} L_{lch} &= L_{ab} & L_{luv} &= L_{uv} \\ C_{luv} &= \sqrt{a^2 + b^2} \quad \text{et} & C_{luv} &= \sqrt{u^2 + v^2} \\ H_{luv} &= \arctan\left(\frac{b}{a}\right) & H_{luv} &= \arctan\left(\frac{u}{v}\right) \end{aligned} \qquad (1)$$

On observera dans ce cas, le même problème de manque de pertinence de l'information H que pour les espaces de type HSL au voisinage de l'axe L. Il est toutefois aussi important de noter que la composante C (chroma) ne correspond pas exactement à la saturation de l'espace HSL (en dépit d'une confusion fréquente) : la chroma, elle, représente la "pureté" de la couleur et on peut obtenir la saturation S à partir de la chroma et de la luminance par S = C/L.

D'un point de vue homogénéité, les deux espaces sont examinés dans [19] il ressort que, s'ils sont nettement plus homogènes que RGB, ils ne sont toutefois pas homogènes. Comparés entre eux les résultats varient selon la portion considérée de l'espace de couleurs. Il semble difficile de trancher en faveur de l'un ou l'autre.

La conversion de l'espace XYZ vers les espaces CIELAB ou CIELUV implique des calculs de racines cubiques ce qui peut avoir un impact sur l'efficacité des calculs. Par ailleurs l'ignorance des conditions d'acquisition induit une erreur quand à l'illuminant choisi. Enfin la conversion RGB (discret) vers LAB/LUV (continus) induit des approximations par rapport à l'image originale qui sont d'autant plus grossières que le codage RGB a été effectué sur un nombre réduit d'échantillons par canal. Ainsi les traditionnels 8 bits par canal sont insuffisants pour avoir une couleur CIELAB/CIELUV reflétant fidèlement l'originale (l'erreur provoqué par la discrétisation produisant des différences significatives). On se retrouve donc condamnés à travailler sur une représentation informatique déjà significativement faussée par rapport à l'originale : le pas de discrétisation de RGB est défini comme bon à partir d'un échantillonnage de 12 bits par canal.

2.2.4. Discussion

Les remarques sur la précision nécessaire pour utiliser rigoureusement les espaces CIELAB/CIELUV/CIELCH, nous incitent à la prudence quant à l'espace de couleurs à utiliser. Si notre ligne directrice qui est de nous inspirer de la perception humaine tend à nous faire utiliser des espaces du type CIELAB ou CIELUV, il faudra par contre vérifier que le gain fourni par l'utilisation de ces espaces est réel et justifie le surcoût en temps de calcul.

Le traitement de la couleur est un sujet très complexe que nous nous attacherons d'aborder de manière rigoureuse à défaut de pouvoir l'aborder de manière complète (les problématiques du traitement perceptuel de la couleur dans le cadre de l'analyse d'image pourraient faire l'objet d'une thèse à elles seules). Comme nous l'avons dit la colorimétrie est une science à part entière et dans nos travaux nous nous baserons sur des ouvrages qui seront, eux, spécialisés et donc plus complets, comme [19] complété par diverses sources qui sont citées, afin de rester le plus possible fidèles à la perception humaine et, le cas échéant, d'être conscients des approximations que nous faisons.

3. **Conclusions**

Nous avons abordé de manière globale différents aspects de l'interprétation humaine et de la perception de couleurs. Comme nous l'avons suggéré, nos travaux peuvent s'inspirer de ces théories dans leurs principes de base (agglomération récursives selon les lois de similarité Gestalt, intégration interactions spatiales pour faire valoir la prépondérance de l'ensemble sur le local, etc.) afin de mettre à disposition du classificateur des informations qui font défaut dans les méthodes classiques de type "bag of visual keywords".

Par rapport à l'étude de la perception de la couleur, avons pu constater que l'expression informatique de la couleur et la compréhension de sa perception physiologique comme son interprétation sont très liées. Nous examinerons un peu plus en détail ces questions d'espace de représentation des couleurs et de distances pour les comparer dans le chapitre 3 à l'occasion de l'étude sur les descripteurs de couleur. Enfin on se doit également de relativiser l'importance de ces représentations. En effet l'exemple de la Figure 7 illustre très bien la dépendance au contexte de la perception de la couleur. En effet les deux éléments tubulaires centraux (en forme de X et perçus respectivement en bleu et en jaune) sont en fait de la même couleur (gris) matérialisée par la flèche.

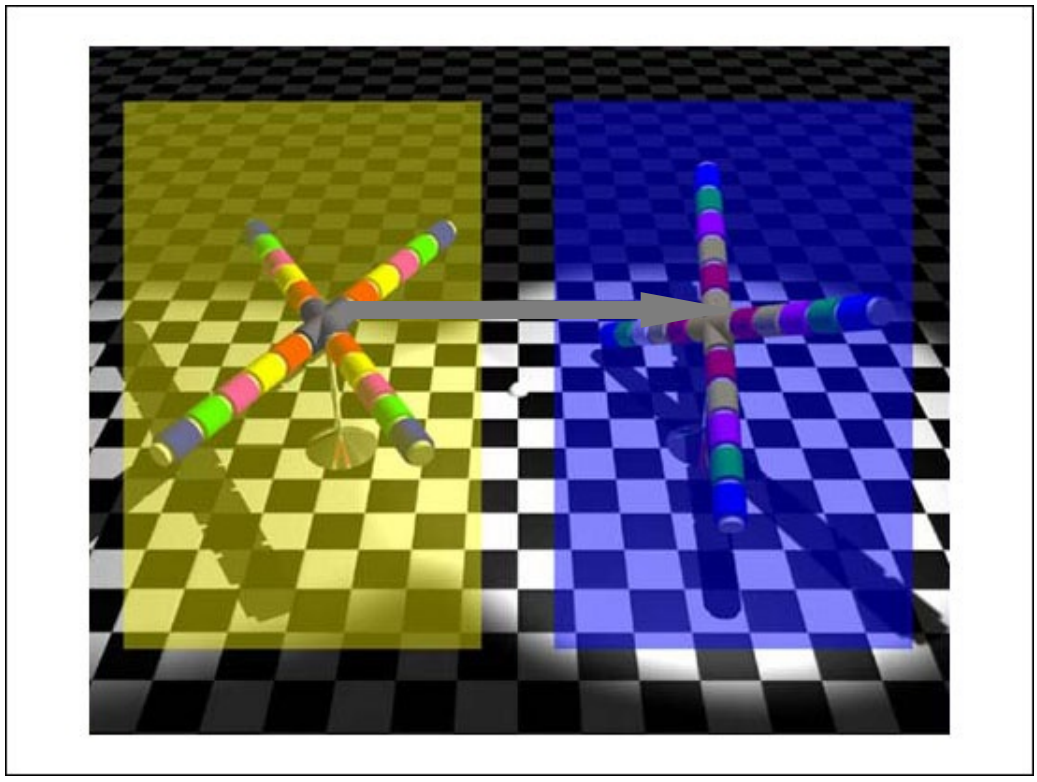


Figure 7: Perception de la couleur et contexte

Chapitre 3: Etat de l'art

Après notre brève étude sur les principes de la perception humaine des images et des couleurs, nous dressons dans ce chapitre un état de l'art sur les principaux descripteurs, les mesures de similarité qui ont été proposés dans la littérature pour la caractérisation du contenu visuel ainsi que les algorithmes de clustering existants pour regrouper ces données entre elles. L'objectif de ce chapitre est de se focaliser sur les données extraites de l'image ainsi que leur traitement indépendamment de leur utilisation. A l'image de la norme MPEG 7 qui a défini un ensemble complet de descripteurs basiques à des fins d'indexation (englobant également l'audio), on utilisera cette section afin de constituer un ensemble d'outils que nous utiliserons dans notre algorithme de segmentation. Les éventuelles insuffisances de l'existant dans notre contexte d'application nous emmèneront également vers nos contributions qui ont tenté d'y remédier.

Nous décrivons ainsi dans la section 1 les données extraites : les descripteurs de couleur, de texture et de forme utilisés actuellement. Nous discutons ensuite dans la section 2 des différentes métriques pour comparer ces données. Enfin dans la section 3 nous étudierons les différents moyens de regrouper ces données en ensembles représentatifs.

1. Caractéristiques Visuelles

Nous débuterons cet état de l'art par un inventaire des caractéristiques de base qui sont utilisées de manière classique pour tous les problèmes d'analyse d'images. Nous les distinguerons en seulement trois catégories mais la littérature regorge de méthodes pour chacune. Il est évident qu'on ne pourrait prétendre à l'exhaustivité, nous nous concentrerons donc sur certaines caractéristiques particulièrement représentatives. Une étude que nous avons menée sur une variété de systèmes de recherche d'images basée sur le contenu (CBIR) nous permettant de dégager quelques caractéristiques qui ont historiquement une forte importance. Nous les complèterons par des caractéristiques utilisées dans les systèmes d'indexation automatique et d'analyse de contenu visuel récents auxquels nous nous comparerons.

1.1. Invariance des descripteurs

Lorsqu'on extrait des descripteurs, que ce soit de manière locale ou globale, il est intéressant de faire en sorte que ce qu'ils décrivent soit aussi indépendant que possible des conditions d'observation : on va essentiellement chercher à être indépendant aux éventuelles transformations linéaires de l'objet (translations, rotations) ainsi qu'à l'échelle d'observation. Ce choix n'est pourtant pas évident au sens ou le procédé de rendre un descripteur invariant s'accompagne nécessairement d'une perte d'information. En effet si on considère par exemple l'invariance en rotation, on peut souligner l'intérêt qu'apporte l'orientation absolue des segments dans le cas où on chercherait à identifier des structures verticales comme des bâtiments ou des arbres or la notion de "vertical" disparait avec la normalisation. On remarquera que, si la notion d'invariance concerne toute les formes de descripteurs, ces exemples suggèrent qu'elle est particulièrement importante pour les descripteurs de forme.

En ce qui concerne l'invariance aux changements échelle on va le plus souvent extraire plusieurs représentations de l'image originale à différents niveaux de détail, puis on va chercher à déterminer une échelle optimale selon un critère donné. Les caractéristiques seront ensuite extraites à cette échelle. La notion d'échelle se matérialise principalement de deux façons: soit via un filtrage d'intensité variable qui va supprimer plus ou moins de détails,

soit sous la forme d'une décimation de pixels de l'image pour produire une image de taille inférieure. Ainsi les descripteurs issus d'une transformée en ondelettes dyadique (utilisée pour la caractérisation de textures, voir plus loin) utilisent ces deux principes (filtrage + décimation). Toujours dans la catégorie des descripteurs de texture, le principe de filtrage est appliqué seul dans les "wavelet frames" de M. Unser [20] qui se basent sur une pyramide de représentations de l'image filtrées par la fonction d'échelle de la transformée en ondelettes mais sans décimation. Si on s'intéresse plus spécifiquement aux descripteurs de forme, la notion de "scale-space" introduite par Lindbergh [13] est apparue peu après. Elle simule l'effet d'échelle au moyen de la convolution par un noyau gaussien de variance plus ou moins importante. Une combinaison des deux mécanismes (flou gaussien + décimation) se retrouve dans le très populaire descripteur SIFT (Scale Invariant Feature Transform) [21] que nous examinerons en détail un peu plus loin.

Nous allons désormais aborder une étude récapitulant les principaux descripteurs découverts lors de l'étude de systèmes existants de recherche d'image par le contenu (Content Based Image Retrieval). Cette étude est présentée en Annexe 1.

1.2. Descripteurs de Couleur

Visuellement d'une importance capitale, les informations de couleur présentent également l'intérêt d'être les seules que l'on puisse extraire avec un haut degré de certitude. En effet : les informations de texture ou de forme sont toutes dépendantes d'une analyse sur un voisinage et en conséquence les informations que l'on extrait sont conditionnées par la fiabilité de cette analyse. Il convient toutefois de relativiser cette affirmation par la remarque faite dans l'introduction au sujet des illusions d'optique : l'information extraite est certes exacte mais cela ne veut pas dire qu'elle est perçue de la sorte.

1.2.1. Expression des Descripteurs

a) Histogrammes

Les histogrammes sont le moyen d'expression le plus simple et le plus courant (voir l'étude sur les CBIR en annexe) pour les descripteurs de couleur. D'une manière générale ils se basent sur la partition de l'espace de couleur en "cellules" (qui peuvent, ou non, se chevaucher voire s'inclure comme c'est le cas de l'histogramme cumulé évoqué plus bas) et expriment l'information de couleur par un vecteur donnant la population associée à chaque cellule.

Une première observation que l'on peut faire est que les histogrammes nécessitent une quantification de l'espace de couleurs, ce qui pose deux problèmes. En premier lieu l'effet de seuillage qui fait que deux couleurs très proches peuvent se retrouver dans des cellules différentes. Ceci peut être résolu par une variante d'histogramme qu'est "l'histogramme cumulé" qui, au lieu de caractériser une couleur par sa population, la caractérise par la somme des populations des cellules "précédentes" de l'histogramme. Ceci rend l'histogramme un peu plus robuste à des changements d'affectations de quelques couleurs à certaines cellules. Par contre pour que le terme "précédentes" ait un sens, cette technique suppose d'être utilisée avec une relation d'ordre au sein de l'histogramme. La mise en place d'une telle relation peut être délicate pour une information définie sur plusieurs composantes comme la couleur.

Le problème de l'expression des trois composantes de la couleur se pose également directement : on peut retranscrire complètement l'information en exprimant la couleur de manière vectorielle (chaque couleur est un triplet) auquel cas, en supposant un espace de couleurs sur trois composantes A, B et C on aura un histogramme de dimensions $q_A \times q_B \times q_C$

où q_X est le nombre de cellules utilisées pour quantifier sur la composante X. Les dimensions de cet histogramme deviennent rapidement trop importantes : si on prend l'exemple de l'espace RGB et qu'on souhaite quantifier chaque composante en 6 classes, on obtient un histogramme de taille très importante ($6^3 = 216$) mélangeant, de plus, dans la même cellule des couleurs perceptuellement très différentes comme l'illustre la Figure 8. Par ailleurs on remarquera qu'avec un tel histogramme les couleurs de l'image seront essentiellement concentrées dans quelques triplets.

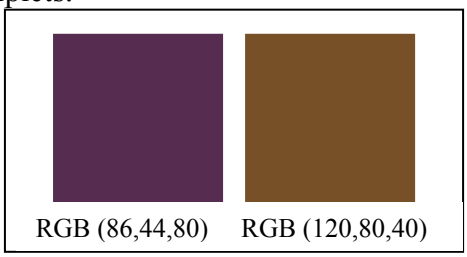


Figure 8: Deux couleurs qui se retrouveraient dans la même classe avec un histogramme 6x6x6

On peut donc choisir de séparer chaque composante et construire un histogramme pour chacune d'elles. La taille du descripteur devient beaucoup plus raisonnable ($q_A + q_B + q_C$) et il contient nettement moins de classes vides, mais cette meilleure représentation se fait au prix d'une perte d'information puisque les trois canaux sont dès lors décorellés.

Outre les problèmes liés à la quantification, l'histogramme souffre de deux autres inconvénients : d'une part l'absence de localisation spatiale et d'autre part la difficulté de comparaison.

L'absence de localisation spatiale fait que deux images très distinctes peuvent avoir deux histogrammes identiques pour peu qu'elles mettent en jeu des couleurs similaires dans des proportions semblables. Ce phénomène est encore renforcé pour l'histogramme cumulé évoqué plus haut. On peut toutefois limiter l'impact de ce phénomène en employant ce descripteur sur un sous-ensemble localisé de l'image (région).

La comparaison de deux histogrammes n'est pas triviale et peut même être assez coûteuse si les histogrammes sont de grande dimension. Les simples distances de Minkowski utilisées entre deux cellules correspondantes d'histogrammes à comparer ne présentant pas une expression fiable de la similarité ; Stricker et al. [20] ont montré que les mesures de type L_1 produisaient un grand nombre de faux négatifs et que les mesures de type L_2 produisaient un grand nombre de faux positifs. Dans la littérature, on retrouve donc surtout des distances plus élaborées comme la mesure d'entropie relative de Kullback Leibler et ses variantes [23] qui ont, pour leur part, donné de bons résultats (voir, dans ce chapitre, la section 2 sur les distances).

b) Moments de couleur

Publiée dans les travaux de M. Stricker et M.Orengo [24], cette méthode représente la couleur de manière très compacte par un vecteur comportant la moyenne, la variance et le coefficient d'asymétrie (i.e. : respectivement, les moments d'ordre 1, 2 et 3). Une étude mentionnée dans [25] fait état de performances seulement légèrement plus mauvaises que celles d'un histogramme de grande dimension.

L'inconvénient évident est que ce descripteur n'est pas exclusivement représentatif de ce qu'il caractérise avec là encore l'absence d'information spatiale. C'est-à-dire qu'il est donc statistiquement possible d'avoir des moments proches pour des images très différentes et qu'il est de plus impossible d'avoir une description des couleurs présentes dans l'image à partir de ce descripteur (au même titre qu'une moyenne ne représente pas l'ensemble des couleurs si

elles sont trop variées ; les moments d'ordre 2 et 3 donnent une idée de cette représentativité). De ce fait, ce descripteur peut être par contre utilisé de manière assez intéressante lorsque sa compacité devient nécessaire ou pour caractériser la distribution des couleurs quantifiées dans une région après une segmentation en couleur : la segmentation garantit que les couleurs de la région sont globalement similaires et les informations statistiques vont permettre de quantifier la répartition des couleurs représentées autour de la couleur moyenne de la région. On remarquera enfin que le problème de la représentation vectorielle de la couleur réapparait ici sous un aspect différent, puisqu'on peut remplacer la moyenne par le centre de gravité des couleurs, et utiliser les distances entre le centre de gravité et les couleurs pour calculer la variance et l'asymétrie. Le descripteur garde ainsi une dimension de 3. On peut aussi étudier les trois moments sur chaque canal. Une telle représentation, si elle provoque une perte d'information, comme évoqué précédemment, mais fournit des informations sur chaque canal qui, ici, n'étaient pas présentes dans le descripteur "vectoriel". On peut aussi remarquer que, même en combinant les deux (descripteur vectoriel, et descripteur par canaux), on se retrouve avec des caractéristiques de taille relativement faible (12 au maximum).

c) "Joint histograms"

Ces histogrammes introduits dans [26] ont pour but d'associer l'information de couleur à d'autres informations, abaissant la possibilité d'obtenir des descripteurs voisins pour des images distinctes. Les informations choisies dans [26] sont des informations basées sur l'intensité et les contours, traduisant donc une structure locale de l'image. Un critère bien plus indicatif que celui de la simple couleur. Ce descripteur est alors combiné à un histogramme de couleur classique. On se retrouve donc avec une représentation de l'information par un histogramme à plusieurs dimensions : la couleur (une ou trois dimensions selon le choix de représentation) complétée par une dimension par information supplémentaire choisie.

A noter que ces histogrammes sont sujets aux mêmes inconvénients que les histogrammes classiques concernant les mesures de distances à utiliser pour les comparer entre eux. Les auteurs utilisent une simple mesure de type L1 mais des mesures plus robustes peuvent être envisagées. Du fait d'un plus grand nombre de dimensions, ce descripteur, évalué dans [26], est toutefois décrit comme 4 fois plus lent qu'un histogramme classique.

d) Vecteurs de cohérence

Nés de la nécessité d'intégrer l'information spatiale de répartition des couleurs pour une similarité plus fiable, ce descripteur décrit par Pass et al. [27] se propose de séparer les couleurs "cohérentes" des couleurs "incohérentes". Par "cohérentes" il désigne les couleurs qui sont dans une zone spatiale de couleurs voisines. Une couleur d'un histogramme sera donc identifiée par la paire désignant respectivement ses effectifs cohérents et incohérents. On se retrouve ainsi avec une caractérisation de l'information de couleur par deux histogrammes : celui des populations des cellules de couleur cohérentes et celui des populations des cellules de couleur incohérentes.

Par ailleurs, l'auteur propose lui-même un affinement de ce type de descripteur par l'augmentation du nombre d'informations pour chaque élément d'histogramme en incluant des informations de voisinage pour les couleurs "incohérentes". Ceci en fait ainsi un descripteur assez précis.

On notera que l'utilité devient moindre si l'on utilise ce descripteur au sein d'une approche régions avec régions segmentées en couleurs homogènes (puisque de ce fait, les couleurs au sein d'une région ne correspondront qu'à des régions cohérentes).

e) "Color correlogram"

Lui aussi orienté vers une intégration de la disposition spatiale de couleurs, le "correlogramme" [28] va rechercher des motifs dans un voisinage donné. Le correlogramme est assimilable a une matrice (n x n x r) ou n est le nombre de couleurs utilisé et r la distance maximale du voisinage considéré. Dans cette matrice, le nombre en (i, j, k) désignera la probabilité de trouver un pixel de couleur i à une distance k d'un pixel de couleur j. Les auteurs préconisent éventuellement de travailler avec un jeu de distances fixe pour réduire la taille de descripteur en explorant des valeurs de k plus importantes (en prenant par exemple pour valeurs possibles de k les distances 1, 3, 5 et 7). La représentation se fait le plus souvent par un vecteur résultant de la concaténation des lignes de la matrice.

Il est évident que le coût en termes de calcul est multiplié par ce passage de une à 3 dimensions par rapport à un histogramme. Aussi les auteurs décrivent-ils dans [28] des méthodes pour les calculer et comparer aussi rapidement que possible. Ils font aussi la remarque que le voisinage immédiat est plus informatif que le voisinage global, aussi de petites valeurs de r constituent tout de même un bon choix.

f) Discussion

Le choix du descripteur adapté est bien évidemment dépendant de l'application. On retiendra que les meilleurs résultats sont logiquement obtenus par la combinaison de l'information de couleur avec des informations supplémentaires au prix d'une notable augmentation du temps de calcul et des dimensions du descripteur. A ce titre, le correlogramme est assez répandu dans la littérature, celui-ci se basant uniquement sur l'addition d'informations spatiales. On peut toutefois noter qu'une telle caractéristique va caractériser une structure spatiale de couleur ce qui s'apparente à une information de texture (voir plus loin). Les histogrammes et les moments gardent un intérêt grâce à leur rapidité d'acquisition ainsi qu'à leur compacité (tout particulièrement les moments qui en plus d'être particulièrement compacts sont faciles à comparer entre eux).

On peut enfin remarquer que la segmentation permet d'effectuer une analyse locale ce qui permet d'incorporer de l'information spatiale et donc de compléter l'information produite par un descripteur simple. Comme nous l'avons noté, si la segmentation est, en plus, effectuée sur un critère d'homogénéité de couleur les moments de couleur deviennent particulièrement interessants car ils traduisent alors la dynamique de couleurs autour de la couleur moyenne au sein de la région.

1.3. Descripteurs de texture

La définition même de la texture est particulièrement délicate. On peut reprendre un exemple donné dans [29] par Landy et Graham avec l'illustration représentée Figure 9. Si la différence entre le ciel et le champ se fait aisément par des mesures de niveau de gris, la différence entre deux zèbres ne peut se faire que par distinction de propriétés de leur texture (à savoir orientation, densité, …). On peut donc la qualifier par la définition "littéraire" de motif répété de manière plus ou moins homogène sur une région avec des propriétés sur la densité de répétition, l'orientation et la forme de ce motif. Dans cette approche de définition on trouve deux expressions qui caractérisent la difficulté de l'extraction de la texture : "motif" et "plus ou moins homogène". La notion de motif, tout d'abord nécessiterait de mettre en commun les informations de plusieurs pixels pour constituer un motif, puis d'analyser son caractère répétitif [30]. D'autre part, le fait que les propriétés soient "plus ou moins"

homogènes rend difficile la délimitation de la région dans l'espace. Cette tache requiert paradoxalement peu d'efforts pour un humain.

De par ces difficultés, la texture va le plus souvent s'identifier par des critères statistiques (ex : détection d'une certaine densité de contours selon une certaine orientation) sans chercher à retrouver précisément le motif. Certains travaux vont aussi remarquer que la notion de texture comprend des propriétés visuelles comme le *contraste*, la *granularité*, la *ruguosité*, la *finesse*, etc. Ainsi, Tamura et al. ont ainsi proposé d'analyser des textures en se basant uniquement sur une quantification de ces attributs visuels [31]. Nous présenterons dans cette section des méthodes utilisées dans le domaine de la recherche et l'indexation d'images par le contenu en nous concentrant plus particulièrement sur les plus utilisées dans le domaine de l'analyse d'image par le contenu (voir l'étude 1. en annexe). Enfin, on remarquera, la perception de la texture, telle qu'elle est grossièrement définie ici, s'inscrit par ailleurs tout à fait dans les principes de la *théorie des Gestalt* évoquée en introduction : on regroupe ici des objets par une proximité spatiale, un alignement, des caractéristiques commune de forme, d'orientation, etc. La récursivité des principes gestalt nous amène aussi à remarquer que la texture peut être composée de "motifs de motifs" et pose ainsi le problème de l'échelle à laquelle on doit analyser la texture.

Figure 9: Exemple de texture

1.3.1. Perception de la texture

S'il est déjà difficile de donner une définition précise de la texture il n'est pas surprenant de dire que les mécanismes de sa perception ne sont pas très bien connus. Landy et Graham évoquent, dans [29], les probables sources de la perception "mécanique" qui seraient les cellules du cortex visuel primaire sélectives selon l'orientation et la fréquence spatiale. C'est à ce titre qu'on peut justifier la démarche de filtres sélectifs selon les orientations ; les filtres dits "de Gabor" [32] en sont les représentants les plus connus et, comme nous en avions parlé en introduction, le parallèle entre cette approche et la vision humaine a été abordé en détail dans [8]. Reste que ces éléments ne sont pas les seuls à intervenir dans le procédé de perception de la texture. On observe d'autres traitements comme des effets de normalisation des intensités, d'autres interactions entre neurones ont également été remarquées. Comme

décrit dans [29], leur exacte nature prête encore à discussion et en particulier les procédés qui permettent à un humain de d'analyser l'homogénéité spatiale et donc de déterminer les frontières des régions de texture homogènes. Si [29] mentionne des expériences qui ont permis des observations sur les circonstances dans lesquelles un observateur humain est efficace, les connaissances sur l'analyse humaine de la texture restent assez limitées.

1.3.2. Expression des descripteurs

Comme nous venons de le voir la texture est une caractéristique assez difficile à définir et il existe en conséquence une multitude de façons de la caractériser, que ce soit dans la définition de ce que l'on cherche à caractériser ou les moyens mis en œuvre pour y parvenir. On notera aussi que ces descripteurs nécessitent une analyse spatiale sur une portion de l'image et on pourra ainsi les distinguer selon leur complexité.

a) Descripteurs "purement" statistiques

Les descripteurs statistiques de texture analysent une région en termes de distribution de ses intensités. L'idée de fond est de considérer la texture comme une disposition aléatoire mais visuellement homogène. On recherche alors à caractériser cette homogénéité par un jeu de statistiques qui seraient invariantes sur des sous-ensembles d'une région de même texture. On distinguera les descripteurs "purement" statistiques, au sens où une autre famille de descripteurs, que nous étudierons juste après, inclut une étape préalable de filtrage de l'image mais la caractérise aussi par des critères statistiques. Nous étudierons ici tout particulièrement deux méthodes purement statistiques qui restent les plus populaires dans les systèmes d'analyse d'image par le contenu (voir l'annexe 1). Le premier modèle de texture a été développé par Julesz dans les années 1960. Il suggérait que la perception de la texture pouvait être expliquée par l'extraction des statistiques d'ordre k, c'est-à-dire des statistiques de cooccurence pour les intensités des k-tuples de pixels. Dans [33], Haralick formalise deux descripteurs pour la texture qui restent encore très utilisés : la matrice de cooccurrence et la fonction d'autocorrélation (elle aussi déjà connue auparavant d'une manière plus générale [34]) qui sont représentatifs de ce type d'approches. Nous décrirons ces descripteurs plus en détail en laissant volontairement de côté d'autres méthodes un peu moins représentées dans la littérature mais se basant sur des principes globaux similaires.

Le principe de la matrice de cooccurrence est le suivant : étant donnée une fonction de destination p (par exemple : "le pixel immédiatement à droite du pixel considéré"), on calcule la matrice P de l'image, dont les éléments P_{ij} sont tels qu'en considérant tous les éléments d'intensité i, on a P_{ij} occurrences d'éléments d'intensité j à la position p. On a donc une matrice de taille $I_q \times I_q$ où I_q est un nombre quantifié de valeurs d'intensités possibles. Par normalisation de la matrice P en divisant ses éléments par le nombre de pixels dans la région considérée, on obtient la matrice de cooccurrence C. Il existe naturellement de nombreuses variantes à ce descripteur, comme définir p sans préciser de direction et simplement une distance (ex : "tout pixel adjacent"), la matrice P devient alors symétrique. On peut aussi rajouter une dimension, d en définissant la fonction p comme "tout point présent à une distance d du point considéré". On retrouve alors le correlogramme [28] évoqué plus haut et qui décrit des structures locales de couleur et par là même donne une information de texture. La plupart du temps la matrice C n'est pas utilisée directement comme descripteur. On en extrait des caractéristiques plus compactes, le Tableau 1 en donne les plus courantes.

Caractéristique	Calcul
Maximum	$\max_{ij}(C_{ij})$
Homogénéité	$\sum_i \sum_j \left(\frac{C_{ij}^2}{1+\lvert i-j \rvert} \right)$
Moment d'ordre k de la différence	$\sum_i \sum_j \left(C_{ij} \cdot (i-j)^k \right)$
Contraste (moment d'ordre 2)	$\sum_i \sum_j \left((i-j)^2 C_{ij} \right)$
Entropie	$-\sum_i \sum_j \left(C_{ij} \cdot \log(C_{ij}) \right)$
Energie (Uniformité)	$\sum_i \sum_j \left(C_{ij}^2 \right)$
Corrélation	$\sum_i \sum_j \frac{\left((i-\mu_i) \cdot (j-\mu_j) \cdot C_{ij} \right)}{\sigma_i \sigma_j}$ Où μ_i, μ_j, σ_i et σ_j sont respectivement les moyennes et les variances des lignes et des colonnes :

Tableau 1: Exemples de caractéristiques d'une matrice de cooccurrence

Les caractéristiques d'autocorrélation, quant à elles, ont connu de nombreuses déclinaisons pour de nombreuses utilisations, que ce soit pour la détection de visages [35] ou encore pour la caractérisation de texture avec leur combinaison avec des techniques plus récentes [36]. Leur principe de base est le suivant : comparer l'image originale avec une image décalée. Par rapport à la matrice de cooccurrence où on considérait un déplacement avec la fonction p, on va ici considérer des déplacements selon chaque axe : Δx et Δy. Etant donné cet ensemble de déplacements envisagés, la fonction d'autocorrélation centrée normée est définie de manière discrète et en dimension 2 par (2) :

$$f_{MN}(\Delta x, \Delta y) = \sum_{x=1}^{M-\Delta x} \sum_{y=1}^{N-\Delta y} \left(\frac{I(x,y) \cdot I(x+\Delta x, y+\Delta y)}{(M-\Delta x)(N-\Delta y)} \right) \qquad (2)$$

Où on considère une région R de dimensions M x N et où la fonction I associe son intensité à un point de coordonnées (x,y). Pour peu que le voisinage exploré ne soit pas trop important, ces caractéristiques présentent l'avantage d'être compactes et rapidement calculables. Comme noté dans [35], x_f peut être éventuellement divisé par une puissance de la somme sur R des f(r) pour le rendre invariant en intensité. On notera aussi que les déplacements p peuvent être généralisés à un ensemble de points à prendre en compte au sein du voisinage comme cela est fait dans [35] et [36]. La Figure 10 (extraite de [36]) donne un exemple de jeux de points pris en considération, chaque jeu produisant une caractéristique.

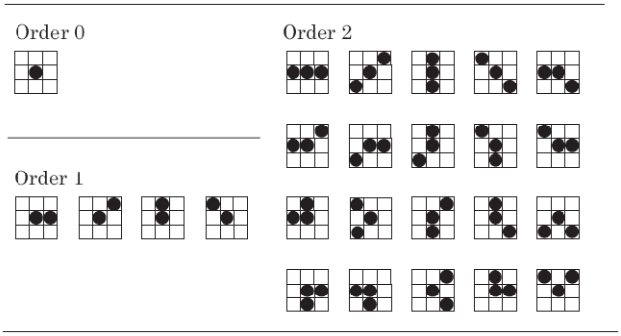


Figure 10: Jeux de points pris en considération pour une fonction d'autocorrélation. La matrice est centrée sur le point où est calculée la fonction. L'ordre est déterminé par le nombre de points pris en considération à l'exception du point central.

b) Extraction de textures par filtrage préalable

Une idée apparue assez rapidement dans le domaine de la texture est d'utiliser le domaine fréquentiel pour tenter de la caractériser. Ceci vient tout naturellement de l'aspect de périodicité qui pousse à étudier les fréquences spatiales pour retrouver une trace du motif. La transformée de Fourier fut le premier outil à émerger dans ce but. Comme on peut le remarquer dans un de ces travaux initiaux [37], un problème qui se pose est l'absence de localisation spatiale, ce qui impose de diviser l'espace au moyen d'une grille régulière (une fenêtre carrée passant sur chaque pixel de l'image étant jugée comme trop coûteuse). Si ce problème rend délicate la détection de texture sur une photo, l'étude de la transformée de Fourier d'une région de texture apporte néanmoins des informations particulièrement intéressantes. Ainsi Bajcy et Lieberman [38] ont calculé des spectres de puissance dans différentes fenêtres sur l'image. L'étude de la forme du spectre de puissance suivant le rayon et la direction permet de caractériser la texture par son orientation et sa périodicité. D'autres travaux (comme [39]) ont permis de dégager des propriétés de la texture telles que la régularité, la "directionnalité", la linéarité ou la finesse par l'étude de portion spécifiques du spectre.

La sélectivité en orientations du cortex visuel primaire a suggéré un modèle qui, de par sa proximité au modèle de la perception humaine et ses résultats intéressants, a engendré une grande quantité de variantes. Ce modèle est composé de trois étapes : d'abord un filtrage spatial sensible aux orientations (de manière similaire aux cellules du cortex visuel primaire), ensuite la sortie de ces filtres est transformée de manière non linéaire (filtrage, saturation et/ou calcul des énergies), enfin un dernier filtrage spatial est appliqué pour homogénéiser la sortie et permettre de définir des contours pour la zone texturée. La Figure 11 (extraite de [29]) illustre ce procédé.

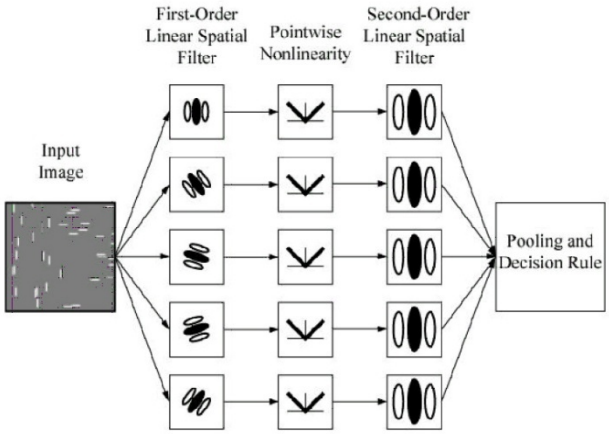


Figure 11: Procédé type de traitement de la texture

On utilise un filtrage qui, en plus d'être sélectif selon une orientation, caractérise la périodicité et permet donc d'extraire localement des informations de même nature que celles qu'on pourrait extraire au moyen d'une transformée de Fourier. Ce principe est devenu particulièrement populaire avec l'apparition des filtres par ondelettes et plus particulièrement des filtres de Gabor [32] qui correspondent au produit d'une fonction gaussienne et d'une fonction cosinus. L'image est donc filtrée avec un jeu de fonctions qui vont isoler différentes orientations et différentes fréquences spatiales. Comme il a été souligné dans [32], il s'agit en fait d'une famille de fonctions; (3) représente la fonction type "g" donnée dans [32] avec sa transformée de Fourier (4) "G". La convolution de l'image d'origine par la fonction "g" produisant l'image filtrée.

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \cdot e^{-\frac{x'^2}{2\sigma_x^2} - \frac{y'^2}{2\sigma_y^2} + 2\pi \cdot j \cdot f_c \cdot (x'\cos\beta + y'\sin\beta)}$$

Avec (3)

$$\begin{cases} x' = x\cos\alpha + y\sin\alpha \\ y' = -x\sin\alpha + y\cos\alpha \end{cases}$$

Les paramètres σ_x et σ_y contrôlent les variances de la gaussienne et représentent donc la sélectivité spatiale du filtre, x' et y' sont des paramètres de rotation, enfin l'utilisation de f_c et de β est plus claire en regardant sa transformée de Fourier (4).

$$G(\omega,\tau) = e^{-\frac{1}{2}\left(\frac{(\omega' - f_c\cos\beta)^2}{\sigma_\omega^2} + \frac{(\tau' - f_c\sin\beta)^2}{\sigma_\tau^2}\right)}$$

Avec (4)

$$\begin{cases} \sigma_\omega = \dfrac{1}{2\pi\sigma_x} \\ \sigma_\tau = \dfrac{1}{2\pi\sigma_y} \\ \omega' = \omega\cos\alpha + \tau\sin\alpha \\ v' = -\omega\sin\alpha + \tau\cos\alpha \end{cases}$$

On y voit plus clairement que β et f_c permettent d'influer sur la réponse fréquentielle.

A titre d'exemple on donne une fonction de gabor (5) utilisée dans [40] :

$$g_{\lambda,\Theta,\varphi}(x,y) = e^{-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2\lambda^2}} \cdot \cos\left(2\pi\frac{x'}{\lambda}+\varphi\right) \qquad (5)$$

σ et γ étant des paramètres définis, θ définissant l'orientation, φ la phase et λ la fréquence spatiale.

Comme cela est montré dans [40], il faut insister sur le fait que le choix de la fonction non-linéaire utilisée, a une importance capitale : les résultats du simple filtrage ne sont pas utilisables directement.

Sur le même schéma, on trouve la grande famille d'extraction de caractéristiques de texture qui se base sur des fonctions "ondelettes" définies comme oscillantes et présentant un support fini (compact) dans le temps (et respectivement dans l'espace pour deux dimensions). Cette transformée en ondelettes continue [41] sur un espace à une dimension t appliquée à un signal f(t) est définie par (6).

$$\Psi_{\chi}^{\psi}(\tau,s) = \frac{1}{\sqrt{|s|}}\int f(t)\cdot\psi^*\left(\frac{t-\tau}{s}\right)dt \qquad (6)$$

Le signal transformé est donc une fonction de deux paramètres : τ et "s" qui représentent respectivement le temps et l'échelle. La fonction ψ est centrée au voisinage de τ et dilatée par "s". L'ondelette mère ψ est ici plus ou moins complexe, l'idée étant que "s" augmentant on dilate la fonction et on analyse des fréquences de plus en plus basses. Il existe en fait une grande variété d'ondelettes mères présentant chacune diverses propriétés. On évoquera d'une part le nombre de ses moments nuls et d'autre part sa compacité spatiale. Le but est de combiner au mieux une bonne résolution spatiale à une bonne résolution fréquentielle ce qui, grossièrement, correspond en pratique à une fonction qui présente à la fois un grand nombre de moments nuls et un support compact. Afin de pouvoir appliquer cette fonction rapidement par filtrage on utilise la "transformées en ondelettes rapide" qui permet de transformer l'opération en une simple convolution circulaire.

Contrairement aux filtres de Gabor évoqués plus haut, es filtres ne font pas intervenir de rotation et sont donc appliqués séparément selon x, y puis simultanément sur les deux donnant un filtrage respectivement horizontal, vertical et diagonal. L'étude se poursuit à des échelles de plus en plus grossières en appliquant une fonction d'échelle puis en recommençant l'analyse. Pour une échelle, une orientation et un voisinage donnés, on caractérise le plus souvent la texture par des critères statistiques simples comme la moyenne et la variance des énergies ; on reste toutefois dans le cadre du schéma illustré Figure 11. L'introduction de cette notion d'échelle est à souligner, et l'analyse de l'image avec plusieurs niveaux de détails se révèle particulièrement utile. A ce titre, un mode d'analyse particulièrement répandu est la décomposition dyadique, c'est-à-dire selon des échantillonnages en échelle suivant une suite géométrique de raison 2 et au moyens de filtres "à reconstruction parfaite". De tels filtres présentent en effet l'intérêt de garantir qu'il n'y a pas perte d'information après filtrage et qu'on peut reconstruire à l'identique une image décomposée à partir du résultat de sa

décomposition. Ces filtres garantissent d'une part une représentation complète et sont d'autre part faciles à appliquer. M. Unser [20] a toutefois montré que la redondance produite par une non-décimation permettait de mieux caractériser la texture.

c) Autres modèles

En dehors des méthodes évoquées précédemment qui représentent une large majorité des caractéristiques de texture, on évoquera les modèles qui visent à analyser la construction de la texture via un modèle probabiliste. On y retrouve essentiellement deux modèles : le modèle SAR (Simultaneous AutoRegressive) [43] et les champs de Markov [44]. Ces approches sont assez minoritaires et nous les aborderons donc assez brièvement pour en montrer le principe. Globalement ces descripteurs considèrent qu'un point d'une région texturée est déduit de son environnement.

Les modèles SAR modélisent un point de la texture comme la transformation d'un autre combiné à un bruit gaussien (7). On voit ici directement apparaître la notion de structure au sens où un point est directement relié aux autres.

$$f(x,y) = \sum_{(p,q)\in R} \alpha(p,q) f(x+p)(y+q) + \omega(x,y) \qquad (7)$$

La fonction α produisant un poids, R étant un voisinage considéré et ω un bruit gaussien de variance σ^2.

Le voisinage R étant donné, les paramètres α et σ sont estimés via une estimation de moindres carrés (Least Square Estimation) ou du maximum de vraisemblance (Maximum Likelihood Estimation). Comme cela est remarqué dans [45], ceci induit le problème du choix d'une taille de voisinage au sein duquel les pixels sont considérés comme interdépendants qui est très variable selon la texture.

Une autre catégorie de méthodes probabilistes considère la texture comme la réalisation d'un champ de Markov. Ainsi dans [44], les auteurs considèrent que chaque point dans la texture est une distribution binomiale dont les paramètres sont contrôlés par le voisinage et le "nombre d'expériences" est le nombre de niveaux de gris considérés. Des champs de Markov sont utilisés pour déterminer les caractéristiques du modèle. Les auteurs soulignent eux-mêmes que cette technique n'est pas valide pour tout type de texture.

d) Discussion

D'une manière globale les approches par filtrage semblent donner les meilleurs résultats sur les bases de tests utilisées [42]. Il faut toutefois noter que ces approches sont caractérisées par de lourdes convolutions qui donnent tout leur intérêt à des descripteurs certes à priori moins efficaces mais beaucoup plus légers en termes de calculs.

Si l'utilisation de filtres de Gabor est actuellement le système le plus répandu, une étude tempère toutefois son efficacité [42]. Cette étude a été effectuée sur plusieurs bases de textures qui sont chacune assez fournies et reconnues : l'album Brodatz, la "MIT Vision Texture Database" et la "MeasTex Image Texture Database". Le mélange des trois bases est en lui-même intéressant au-delà du nombre d'exemples qu'il procure puisqu'il fournit des images acquises dans des conditions différentes et avec des équipements différents.

Il en ressort en premier lieu que les performances, pour l'ensemble des textures, ne sont globalement pas exceptionnelles (une moyenne 70% de classification correctes sur un jeu

de textures d'essai, les résultats variant entre 66% et 74%), les jeux de filtres de Gabor, en particulier, donnant des performances très moyennes ; la raison invoquée étant une mauvaise séparation des fréquences dans l'image. Un filtre basé sur les simples coefficients DCT, donne de bons résultats à l'exception de quelques catégories spécifiques de textures ; cette méthode reste cependant un intéressant compromis entre vitesse de calcul et performances. Les filtres les plus efficaces sont en fait la famille des filtres par ondelettes (utilisant des ondelettes mères qui ont une meilleure séparation en fréquence) et surtout des QMF (Quadrature Mirror Filters). Les auteurs soulignent eux-mêmes qu'il faut relativiser ce classement par la proximité absolue des résultats : même si certains filtres sont le plus souvent meilleurs que d'autres, aucun n'est substantiellement le meilleur. En conséquence de quoi ils recommandent si possible de tester plusieurs approches, mais dans l'absolu conseillent l'utilisation de filtres de type QMF en suggérant toutefois d'utiliser le filtre DCT si on souhaite réduire le temps de calcul. Enfin on notera que les méthodes statistiques, abordées dans cette étude, présentent des résultats acceptables mais légèrement inférieurs à ceux obtenus avec des filtres. Le choix de l'ondelette mère reste un problème récurrent, même si les travaux de M.Unser [20] ont montré qu'il était possible de caractériser assez simplement la texture par les coefficients des ondelettes en utilisant une ondelette mère relativement simple (comme les ondelettes de Daubechies).

Enfin on notera que l'espace d'expression des couleurs de l'image joue un rôle important dans les éléments extraits (contours, statistiques après convolution, etc.) et a donc un impact significatif sur les résultats.

1.4. Descripteurs de forme

Les descripteurs de forme sont très répandus dans l'analyse d'image, et pour cause : les images sont précisément analysées pour rechercher des objets ou plus généralement un contenu caractérisable par sa forme. Ces descripteurs se rattachent à la notion de contour qui s'extrait par l'étude des intensités de l'image.

1.4.1. Extraction de l'information locale de contour

L'extraction de contours à partir des intensités des pixels peut se faire de nombreuses façons. La plupart des méthodes impliquent le calcul du gradient d'intensité entre un pixel et ses voisins qui peut, là encore, se faire de diverses façons ; les plus répandues étant l'application de filtres de Sobel ou de Prewitt (évoqués dans [46]) qui sont de simples filtres de convolution ; on peut aussi citer le filtre de Roberts ou le gradient morphologique (cités dans [19]).

Un autre aspect à considérer est l'information de couleur : traditionnellement on utilise des images en niveau de gris pour l'extraction des contours, ce qui provoque une perte de certains contours (cas où un contour est constitué par deux couleurs distinctes correspondent au même niveau de gris). On dispose donc de plusieurs façons d'intégrer cette information. La plus simple est de détecter les contours sur chaque canal ; on applique ensuite un opérateur de fusion (le plus souvent un "ou") ; des méthodes de fusion plus avancées ont également été développées [47]. Enfin les méthodes de calcul de gradient vectoriel citées dans [19] et introduites dans [48] proposent des méthodes plus coûteuses en termes de calcul mais plus précises par rapport à la perception, le vecteur gradient étant calculé par rapport à ses trois composantes en recherchant la direction pour laquelle ses variations sont les plus élevées.

Par ailleurs, une fois ce filtrage initial effectué, on obtient une carte d'intensités et d'orientations de gradients que l'on peut ou exploiter comme telles ou utiliser pour affiner la détection de contours et prendre en particulier une décision binaire "contour/non-contour". Le filtre le plus populaire dans ce domaine est le détecteur de Canny [49], qui utilise un phénomène d'hystérésis pour détecter un contour et le poursuivre en dépit d'éventuelles variations d'intensités. En ne retenant que les maxima locaux d'intensité, ce filtre présente également l'intéressante propriété de garantir des contours d'épaisseur 1.

Un autre type de traitement plus avancé permettant d'obtenir une décision quant à la présence ou non de contours a été introduite par Perona et Malik [50] et utilise la diffusion anisotrope. Ce procédé reprend le principe de la diffusion de température qui, en physique, homogénéise la température des objets. Cette diffusion sert de première étape afin de supprimer les perturbations locales du signal. Il est alors possible dans un second temps d'effectuer une recherche des contours. L'équation de diffusion de base (isotrope) correspond en fait à la simple application d'un filtre gaussien (8) :

$$\frac{\partial T(x,t)}{\partial t} = c \cdot \frac{\partial^2 T(x,t)}{\partial t^2} \tag{8}$$

Avec "c" coefficient de conduction.

Un filtre gaussien endommageant les contours, Perona et Malik [50], ont donc choisi de rendre la diffusion anisotrope en appliquant l'équation correspondante (9) :

$$\frac{\partial T(x,t)}{\partial t} = div\left(c(x,t) \cdot \nabla T(x,t)\right) \tag{9}$$

Où c devient variable. Il doit alors être proche de 0 au niveau des contours et proche de 1 dans les zones à homogénéiser.

Le filtrage ainsi construit est appliqué itérativement à l'image, on remarque par contre que c doit être recalculé en chaque point de l'espace. Ce type de solution devenant, de fait, un peu plus lourde. Il faut mentionner que l'extraction de contours n'est pas la vocation première de cette méthode. Elle présente le très grand intérêt d'homogénéiser les régions et à ce titre peut constituer une étape préliminaire intéressante avant une segmentation afin d'éliminer le bruit dans l'image. Enfin on notera que si Perona et Malik indiquent explicitement dans [50] l'application de leur algorithme à l'extraction de contours, et sont cités à ce titre (ainsi qu'en tant que fondateurs de cette méthode) nombre d'autres méthodes, plus performantes, ont vu le jour depuis.

Là encore on peut relativiser les résultats extraits à la lumière de la perception humaine, on peut mentionner le phénomène de contours "imaginaires" c'est-à-dire des contours tracés par le principe Gestalt de fermeture. Ainsi si on reprend la Figure 9, on distingue certaines parties du contour des zèbres non pas par des frontières directement visibles mais en fermant un contour deviné grâce aux extrémités des rayures. Si dans ce cas là une analyse de texture permettrait de déterminer ces contours (c'est d'ailleurs ce que permet, dans une certaine mesure, l'algorithme de segmentation Edgeflow [51]), la Figure 12 illustre un cas ou la perception de contours par un humain est particulièrement difficile à modéliser.

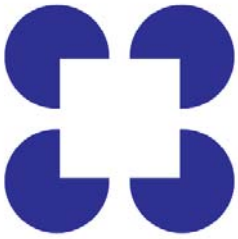


Figure 12: Exemple de contours imaginaires

1.4.2. Expression des descripteurs

a) Descripteurs de forme globale

Lorsqu'on recherche un objet en particulier, caractériser sa forme globale produit naturellement des indications précieuses. En particulier, on a développé de nombreuses méthodes pour rechercher précisément une forme connue dans une image. Les méthodes basées sur les contours actifs ou leurs dérivés (ex : la méthode des éléments fins [52], utilisée dans le CBIR "photobook" [53] ou les contours actifs géodésiques [54]) permettent, à partir d'un exemple ou d'un modèle, de trouver des formes similaires. L'idée étant d'associer la forme requête aux contours de l'image cible en minimisant l'énergie nécessaire à l'adaptation ; ces systèmes sont assez répandus pour des applications comme le "tracking" en vidéo [55]. Toutefois, ils sont assez couteux à mettre en œuvre sans information à priori (localisation délicate par essais successifs) et nécessitent de disposer de modèles couvrant exhaustivement les objets à rechercher ; ces modèles étant globaux, ils sont également sensibles à l'occlusion même si les mesures de variations d'énérgie entre le modèle et l'objet detecté peuvent permettre des adaptations (des modèles ont notamment été proposés pour. On notera également la difficulté posée par le fait qu'on est dans la plupart du temps dans l'impossibilité de déterminer précisément les contours d'un objet dans l'image.

Si on étudie plutôt les méthodes de caractérisation des formes, on trouve une variété de méthodes plus ou moins complexes. Ces descripteurs sont abordés dans l'étude de Peura et Iivarinen [56]. En fait, parmi les systèmes CBIR que nous avons étudiés (cf. Annexe 1) nous avons retrouvé une partie de ces descripteurs avec d'une part certains basés sur l'étude précise des courbures des contours (les descripteurs de Fourier étant les plus répandus dans cette catégorie [57]) et d'autre part des caractérisations de forme globale (voir Figure 13, extraite de [56]).

Nous ne nous attarderons pas sur les techniques d'analyse précise des contours : pour les raisons citées plus haut, si elles restent valides pour une approche basée sur une requête utilisateur, elles ne peuvent être appliquées efficacement à notre problème. On se concentrera sur des méthodes plus robustes (ou tout au moins qui ne nécessitent pas de couteux modèle pour devenir robuste à l'occlusion et aux léger changements d'aspect) et éventuellement applicables à des régions qui concernent des parties d'objets.

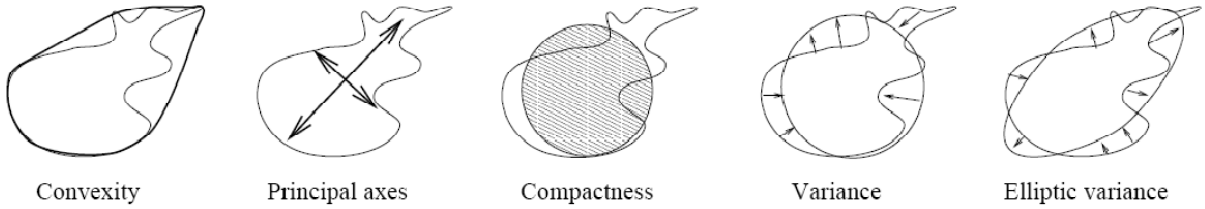


Figure 13: 5 exemples de descripteurs de forme basiques

On trouve ainsi des descripteurs qui vont décrire des propriétés spatiales très basiques comme la taille du rectangle englobant ou les axes principaux et l'orientation de l'ellipse englobante ; sur le même principe on citera les caractéristiques utilisées dans le célèbre système CBIR d'IBM "QBIC" [58] : surface, circularité, orientations principales. Enfin, une méthode très répandue est de décrire une forme en se basant sur ses moments (10).

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y) \qquad (10)$$

Où I(x,y) désigne l'intensité du point de coordonnées x,y.

On peut aussi considérer l'image comme une fonction de densité de probabilité. On divisera alors le moment par la somme des intensités sur la région considérée. Il est également possible de faire abstraction de l'intensité et de considérer une image binarisée (réduisant les valeurs possible de I à 0 et 1). D'une manière plus courante, on travaille sur les moments centraux μ (11) qui ont la propriété importante d'être invariants en translation.

$$\mu_{ij} = \sum_x \sum_y (x-\bar{x})^i (y-\bar{y})^j I(x,y) \qquad (11)$$

Où $\bar{x}$ et $\bar{y}$ sont les coordonnées du centre de gravité de la région ($\bar{x} = \frac{M_{10}}{M_{00}}; \bar{y} = \frac{M_{01}}{M_{00}}$).

Ces moments peuvent être rendus invariants en échelle (12) :

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{i+j}{2}\right)}} \qquad (12)$$

A partir de ces informations, Hu (cité dans l'étude [59]) a mis au point, dès 1962, un jeu de 7 invariants en translation, rotation et échelle (Tableau 2). Ceux-ci sont calculés à partir des moments centraux et permettent de décrire des propriétés de forme d'un objet ou d'une région (symétrie, etc…).

I_1	$\eta_{20} + \eta_{02}$
I_2	$(\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2$

I_3	$(\eta_{30}-3\eta_{12})^2+(3\eta_{21}-3\eta_{03})^2$
I_4	$(\eta_{30}+\eta_{12})^2+(\eta_{21}+\eta_{03})^2$
I_5	$(\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})\left[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2\right]+$ $(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})\left[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right]$
I_6	$(\eta_{20}-\eta_{02})\left[(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})\right]+4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03})$
I_7	$(3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})\left[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2\right]-$ $(\eta_{30}-3\eta_{12})(\eta_{21}+\eta_{03})\left[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right]$

Tableau 2: Les 7 moments invariants de Hu

Il s'agit des moments les plus simples utilisés dans le cadre de l'analyse d'images et présentent l'atout d'être simples à extraire et à calculer. Il existe en fait une variété d'autres moments. Leurs atouts et leurs principes de base restent les mêmes : ils se calculent à partir de la position de l'ensemble des pixels d'une région et peuvent donc être appliqués à des régions complètes et sont donc plus robustes à des petits changements de point de vue et à l'occlusion. On citera deux autres caractéristiques évaluées dans [60] comme plus performantes que les moments de Hu : les moments de Zernike et les moments de Chebyshev. Ces deux derniers étant construits sur le même principe : on définit une famille de fonctions des coordonnées des points de la forme à caractériser et on se base sur ces fonctions pour calculer des moments invariants. A titre d'exemple, nous donnerons l'expression des populaires moments de Zernike.

Les moments de Zernike se calculent comme il suit ; pour calculer les moments d'ordre n avec une répétition l, on définit les polynômes complexes de Zernike dans un espace discret par (13).

$$Z_{nl}(x,y)=Z_{nl}(r\cos(\theta),y\sin(\theta))=\zeta_{nl}(r,\theta)=R_{nl}(r)e^{jl\theta} \quad (13)$$

Avec

$$R_{nl}(r)=\sum_{s=0}^{\frac{n-|l|}{2}}(-1)^s\frac{(n-s)!}{s!\left(\frac{n+|l|}{2}-s\right)!\left(\frac{n-|l|}{2}-s\right)!}r^{n-2s}$$

r désignant la distance entre le centre du repère et le point considéré, n + |l| étant toujours pair.

Les moments de Zernike se calculent alors selon (14) sur une région M x N :

$$A_{nl}=\frac{n+1}{\pi}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}I(i,j)\cdot R_{nl}(r_{ij})\cdot e^{-il\theta_{ij}} \quad (14)$$

Il faut noter qu'il est nécessaire de normaliser le système de coordonnées de l'image car ces moments se calculent à l'intérieur du cercle unité. Un algorithme de normalisation comme "shape compacting" mentionné dans [60] doit être utilisé pour les rendre invariants en échelle et en translation.

b) Informations globales basées sur le gradient

Plutôt que de décrire nécessairement des formes connues et identifiables, Jain et Vailaya, dans leur travaux sur la classification, [61], [62] ont proposé un descripteur qui décrit une image ou une région à partir des contours qu'elle contient. On s'écarte ici de la notion d'objet et on devient par là même plus robuste dans ce . Le principe de base est simple : on considère l'image en terme d'intensités et de directions de ses gradients, peu importe la manière dont elle a été obtenue (gradient vectoriel [48], calculé à partir de filtrages horizontaux et verticaux, etc.). A partir de cette information le descripteur crée est un simple histogramme de population pour chaque orientation. Les travaux originaux de Jain et Vailaya utilisant un détecteur de Canny [49] qui produit une image de contours binaire, l'information d'intensité du gradient n'est prise en compte qu'indirectement sous la forme de la prise en compte des points ou non, selon leur intensité. Enfin il est intéressant de noter que les informations de contour prises sur une région complète contiennent dans une certaine mesure des informations de texture, puisqu'elles donnent des informations sur les orientations et les intensités des contours au sein de cette région (propriétés de la texture évoquées dans la section correspondante). Ces informations sont illustrées par la Figure 14 (extraite des travaux de Vailaya).

Une autre méthode de caractérisation de la forme est évoquée dans [63]. Il s'agit d'un descripteur qui se présente sous la forme d'une matrice de cooccurrence, très similaire à ceux utilisés pour la couleur et la texture présentés précédemment. En effet il s'agit ici de rechercher des points de contour dans un voisinage donné et l'information figurant dans la matrice de cooccurrence est C_i^j où i et j représentent les orientations. Dans l'évaluation proposée dans [63], cette méthode présente de bons résultats sur une tache de classification ; légèrement supérieurs à l'histogramme de contours.

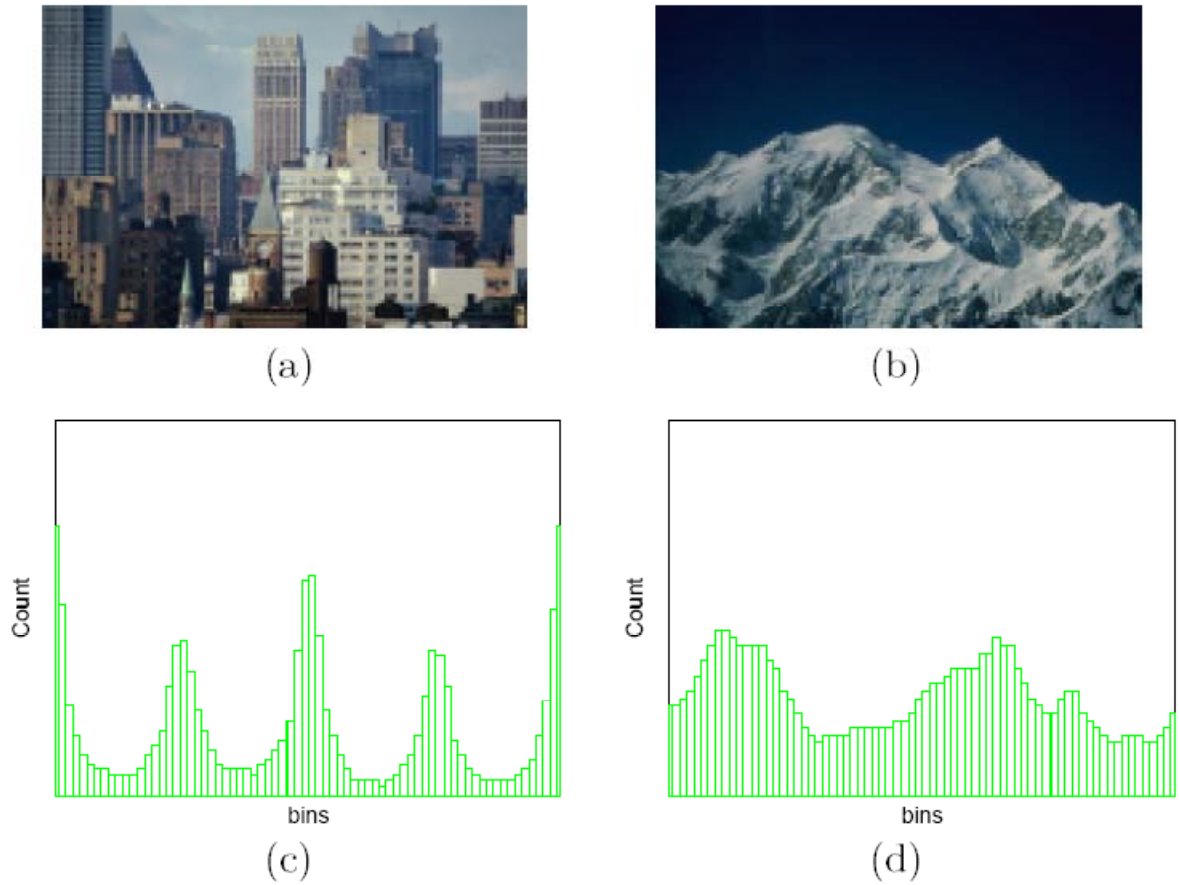


Figure 14: Histogramme de contours

1.4.3. Approches par points d'intérêt

Brièvement évoqués en introduction, les points d'intérêt permettent d'extraire des caractéristiques de forme localement. On étudiera séparément le choix de ces points et les caractéristiques extraites même si certains descripteurs comme SIFT comprennent à la fois un algorithme de détection de points d'intérêt et des caractéristiques d'extraction.

Le choix des points d'intérêt peut se faire de plusieurs façons : selon une grille, de manière aléatoire plus ou moins dirigée ou selon des détecteurs basés sur des critères plus poussés comme Harris-Laplace [64] ou le détecteur SIFT [21].

Un premier détecteur très populaire a été développé par C. Harris and M.J. Stephens [65] en partant du postulat que les points situés sur des coins étaient des points d'intérêt. La détection des coins s'effectue au moyen d'une adaptation de la fonction d'autocorellation vue plus haut. Si on considère pour une image I que ses images de contours horizontaux et verticaux sont I_x et I_y extraites par filtrage (Sobel, …) et l'application d'un filtre gaussien f(I), la matrice H de Harris s'exprime par (15) :

$$H = \begin{bmatrix} f\left(I_x^2\right) & f\left(I_x I_y\right) \\ f\left(I_x I_y\right) & f\left(I_y^2\right) \end{bmatrix} \qquad (15)$$

H étant calculée en un point donné, les valeurs propres de H permettent d'obtenir des informations topologiques sur la situation du point considéré. Pour avoir un point isolé ou un point, les deux valeurs propres doivent être de valeur importante. Plutôt que de les calculer directement on définit la réponse k du détecteur par (16) :

$$k = \det(H) - \lambda trace(H)^2 \qquad (16)$$

λ étant un paramètre fixé (valeur suggérée 0,04).

La réponse k est positive au voisinage d'un coin, négative au voisinage d'un contour et faible dans une région d'intensité constante. Les points d'intérêt s'obtiennent grâce à des maxima locaux de la valeur de k.

Sous cette forme le détecteur ne donne pas d'indication sur l'échelle à laquelle doivent être extraites les caractéristiques. C'est pour pallier à cette absence d'indication d'échelle qu'on a cherché à compléter ce détecteur par une analyse multi-échelles, les travaux de Dufournaud et al. [66] puis de Mikolajczyk et Schmid [64]. Dans [66] les points sont simplement détectés à des maxima locaux de la fonction de Harris appliquée à différentes échelles alors que dans [64], on intègre la notion de "scale-space" telle qu'elle est définie par Lindeberg dans [13] directement dans le procédé de détection.

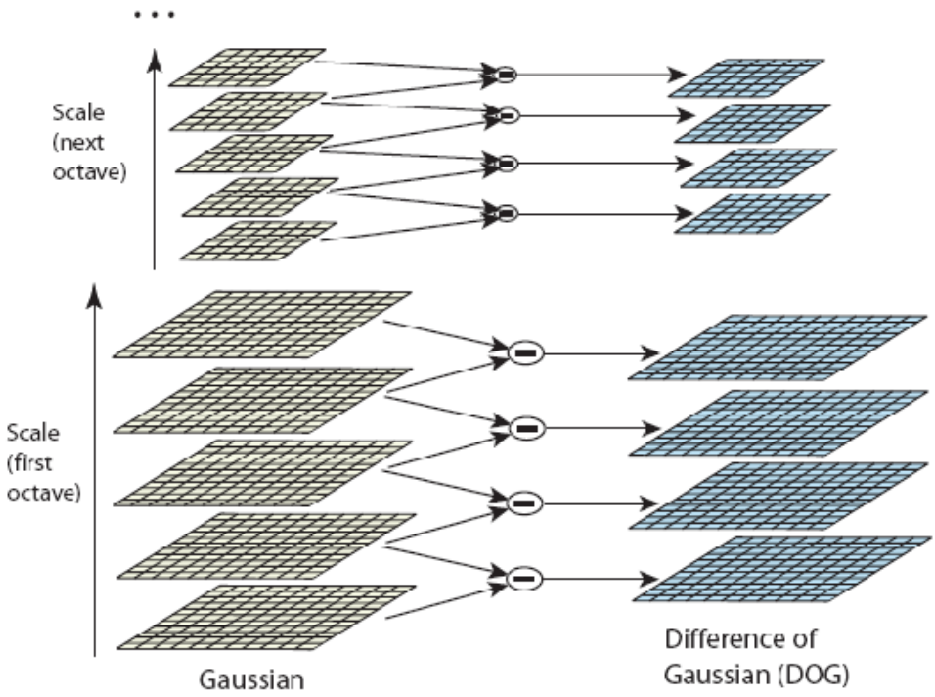


Figure 15:Représentation de l'image dans le détecteur SIFT

Enfin on citera le mécanisme de détection utilisé pour le descripteur SIFT [21], celui-ci utilise une combinaison d'une réductions de l'image avec la notion de "scale-space" qui applique un flou gaussien d'une variance évoluant avec l'échelle. Les représentations retenues ne sont pas directement les images transformées mais les différences avec le niveau précédent (cette représentation est connue comme DoG pour Différence of Gaussians). SIFT caractérise ainsi une série d'images DoG pour chaque échelle, comme l'illustre la Figure 15 (extraite de [21]).

A partir de cette représentation, on recherche les extrema locaux. Ceci produit à l'évidence un très grand nombre de points ; aussi, différents filtrages sont opérés afin de produire le jeu final de points d'intérêt. Il faut remarquer que le jeu de points d'intérêt ainsi produit est assez fortement dépendant d'un jeu paramètres à déterminer préalablement.

La méthode d'extraction la plus efficace n'est pas évidente à déterminer car même les algorithmes avancés ne permettent pas forcément de trouver des points sur les régions importantes de l'image plus sûrement que ne le ferait un simple tirage aléatoire ou le choix de points espacés régulièrement selon une grille de pas fixe. En effet la typicité d'un point par rapport à une structure est une notion assez difficile à généraliser.

Par ailleurs l'utilisation d'une grille, une méthode en apparence assez primitive, a le mérite de bien recouvrir l'espace et a, par ailleurs, produit de manière assez nette les meilleurs résultats lorsqu'elle a été évaluée comparativement à d'autres méthodes d'extraction de caractéristiques dans [67].

Les caractéristiques utilisées sont évidemment extraites dans un voisinage immédiat du point afin de conserver les propriétés qui font leur attrait (leur reproductibilité, leur robustesse, notamment à l'occlusion, …), ces caractéristiques locales recherchent principalement à renforcer l'atout des points d'intérêt en visant l'invariance aux transformations affines et aux changements d'échelle. Mindru et al. proposent dans [68] l'utilisation des moments de couleur généralisés (17) pour une image codée dans l'espace RGB :

$$M_{pq}^{abc} = \iint_{\Omega} x^p y^q \left[R(x,y)\right]^a \left[G(x,y)\right]^b \left[B(x,y)\right]^c dxdy \qquad (17)$$

Ils se calculent l'ordre p+q et au degré a+b+c. Ces moments caractérisent la forme, l'intensité et la distribution de couleur au sein d'une région Ω. On utilise le plus souvent des invariants jusqu'à l'ordre 2 et au premier degré, produisant ainsi 18 caractéristiques.

Une autre famille d'invariants existe sous la forme des invariants différentiels. On citera tout d'abord le "local jet" par Koenderink et al. [69], qui propose de décrire le voisinage du point d'intérêt au moyen d'une approximation par la convolution de l'image par des dérivées d'un noyau gaussien ; on définit ainsi L^{σ}_{ij} d'ordre i+j avec σ facteur d'échelle (18):

$$L_{ij}^{\sigma} = G_{ij}^{\sigma} * I(x,y) \qquad (18)$$

Avec

$$G_{ij}^{\sigma} = \frac{\partial^{i+j}}{\partial x^i \partial y^j} G^{\sigma}$$

$$G^{\sigma} = \frac{1}{2\pi\sigma^2} e^{\frac{-\left(x^2+y^2\right)}{2\sigma^2}}$$

Le principe de calcul des invariants est de combiner les différentes composantes du jet local de manière à obtenir des grandeurs qui soient invariantes à divers changements d'aspect, notamment transformation affines et changement d'illumination. Sur ce même principe, plusieurs caractérisations ont été construites ultérieurement se basant également des combinaisons de filtrages différentiels.

Une autre approche consiste à vouloir caractériser la structure locale des contours environnants. On retombe ainsi sur des mécanismes qui rappellent les matrices de cooccurrence, où l'on parcourt l'espace au voisinage du point en représentant les intensités/orientations/positions du gradient. On mentionne tout d'abord le descripteur SPIN utilisé dans [70] dérivé des "spin-images" définies dans [71] pour la caractérisation d'objets 3D. Cette caractéristique s'exprime sous la forme d'une matrice M où chaque élément M_{ij} représente la probabilité de trouver l'intensité quantifiée i à une distance quantifiée j du point d'intérêt. On obtient ainsi une simple description de la structure locale de manière invariante en rotation.

Le descripteur associé au détecteur SIFT [21] décrit un peu plus précisément les informations locales du voisinage. En particulier, en se basant sur des informations de gradient, il va intégrer trois notions qui sont la localisation grossière par rapport au point d'intérêt, l'intensité et l'orientation du gradient. L'extraction est illustrée sur la Figure 16; son principe est le suivant : on divise l'espace autour du point d'intérêt en régions de 4x4 pixels (4 sur l'illustration) et on construit un histogramme d'orientations pour chacune de ces régions. Sur l'illustration on a choisi 8 bins pour l'orientation, pour chaque pixel de la région concernée, le bin de l'angle correspondant est incrémenté de l'intensité du gradient. La dimension du vecteur de caractéristiques final est de (nombre de bins d'orientation) x (nombre de régions). Le descripteur SIFT, tel qu'il est décrit dans [21], comprend 8 bins d'orientation et 16 régions pour un vecteur de caractéristiques de taille 8 x 16 = 128.

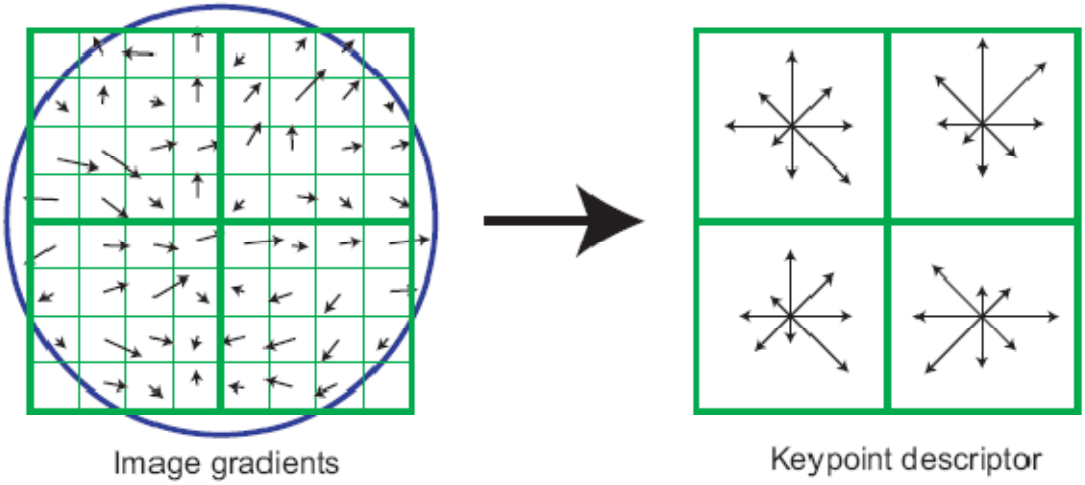


Figure 16: Caractéristiques associées au descripteur SIFT

On citera enfin le descripteur RIFT [72], qui reprend les principes du descripteur SIFT avec quelques modifications afin de le rendre invariant en rotation. Le principe général reste le même : on décompose l'espace en régions dans lesquelles on calcule un histogramme d'orientations. La différence est qu'on utilise là des régions concentriques et que les orientations du gradient en un point sont calculées par rapport à la direction du centre en ce point. Les auteurs proposent 4 cercles et 8 bins d'orientation pour un vecteur de taille 32 (illustré sur la Figure 17, extraite de [72]).

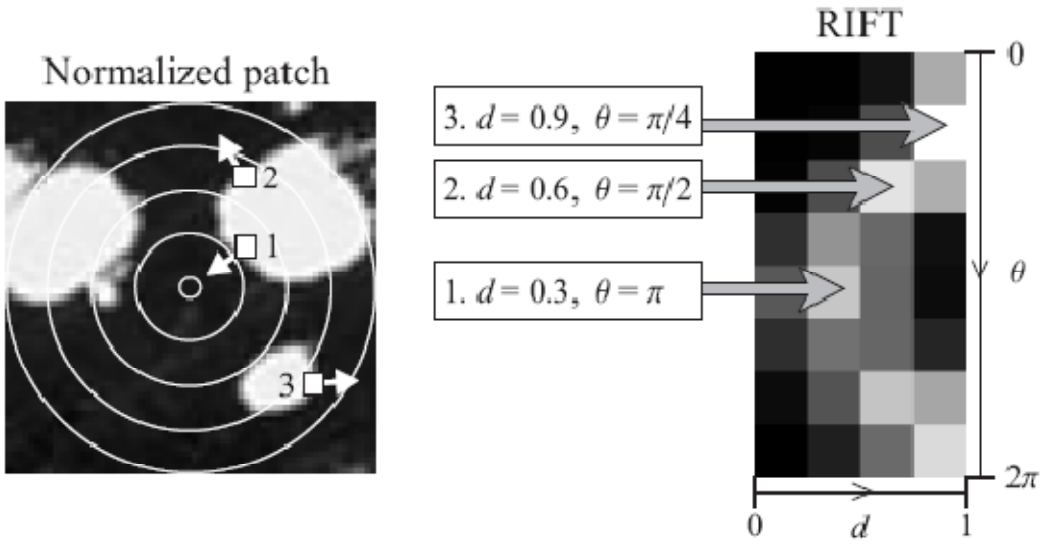


Figure 17: Caractéristiques du descripteur RIFT

1.4.4. Discussion

Comme nous l'avons évoqué dans notre étude sur la perception humaine, la perception de la forme est un élément capital dans l'acte d'interprétation d'une image. Les descripteurs abordés ici sont donc, dans l'absolu particulièrement intéressants : ils abordent le problème soit de manière globale pour les premiers, soit de manière locale pour les suivants. Les premiers même s'ils paraissent plus intéressants (prépondérance du global sur le local) sont tributaires d'une décomposition qui n'est à priori pas fiable. Les descripteurs locaux, en revanche, vont pouvoir permettre d'extraire de manière assez fiable des informations partielles comme des coins ou des régions géométriquement caractéristiques. De par le caractère partiel de leur extraction, on peut toutefois s'interroger sur la répétabilité de leur extraction dans différentes conditions. Il semble plus sûr de les multiplier mais ceci alourdit la méthode et peut générer du "bruit" par un excès de descripteurs non-significatifs.

On remarquera également que la remarque faite sur les descripteurs de texture au sujet des espaces de couleur reste valide pour les différents descripteurs abordés ici. Par exemple, la mesure du gradient entre deux pixels de couleur sera très dépendante de l'espace d'expression de celles-ci.

Enfin, comme nous l'avons fait remarquer, l'invariance est un problème qui se pose tout particulièrement pour les descripteurs de forme. A ce sujet, on notera les difficultés majeures que posent les transformations en trois dimensions et les déformations internes d'une même forme. On trouve certains systèmes très spécialisés (ex : reconnaissance de la main [73]) qui arrivent à contourner le problème en envisageant différents cas possibles, d'autres systèmes, en suivi temporel, proposent une modélisation de la déformation [74]. Ces systèmes ont en commun une spécialisation qui leur permet d'adapter leur modèle au problème traité ; dans le cadre d'une classification généraliste comme la notre, on se retrouve contraints d'apprendre les objets transformés comme s'ils étaient chacun une instance distincte.

2. Distances et mesures de similarité

Les distances entre des caractéristiques ou des ensembles de caractéristiques sont au cœur des méthodes de classification et permettent les opérations élémentaires de

transformation de caractéristiques. L'étude des différentes distances à notre disposition est par conséquent une étape incontournable. Dans cette section nous ferons l'inventaire des différentes distances à notre disposition en nous attardant un peu sur la comparaison de couleurs.

D'une manière générale nous chercherons à comparer deux vecteurs X et Y dont les composantes seront respectivement $[x_0, x_{1, ..., } x_N]$ et $[y_0, y_{1, ..., } y_N]$. Nous spécifierons à chaque fois les conditions d'application si besoin est.

2.1. Distances de Minkowski

Les distances les plus répandues sont les simples distances de Minkowski notées L_p dont sont extraites la distance euclidienne et la distance de Manhattan. On les calcule selon (19) :

$$L_p(X,Y) = \left[\sum_{i=0}^{N} |x_i - y_i|^p \right]^{\frac{1}{p}} \tag{19}$$

La distance L_2 désignant la distance euclidienne et la distance L_1 correspond à la distance de Manhattan. On utilise parfois également la distance infinie (20) :

$$L_\infty(X,Y) = \lim_{p\to\infty} \left[\sum_{i=0}^{N} |x_i - y_i|^p \right]^{\frac{1}{p}} = \sup_i |x_i - y_i| \tag{20}$$

2.1.1. Distances entre Histogrammes

L'histogramme est un mode de représentation des données simple et particulièrement répandu. La comparaison d'histogrammes pose toutefois problème, comme nous l'avons évoqué dans le chapitre sur la couleur (effet de "seuil" et plus généralement une sensibilité aux légers décalages). A ce titre des mesures plus efficaces ont été mises au point afin de pouvoir les comparer. On reprendra les mêmes notations avec X et Y représentant les histogrammes à comparer et x_i et y_j leurs populations respectives pour les "bins" i et j.

2.1.2. Intersection d'histogrammes

Développée par Swain et Ballard [12], elle est faite pour comparer deux histogrammes de couleur. Cette distance, comprise entre zéro et un, n'en est pas une au sens mathématique du terme : elle est singulière parce qu'elle ne vaut pas zéro pour deux histogrammes identiques (elle vaut un) et est d'autant plus grande que les histogrammes sont similaires.

$$d_\cap(X,Y) = \frac{\sum_{i=0}^{N} \min(x_i, y_i)}{\min\left(\sum_{i=0}^{N} x_i, \sum_{i=0}^{N} y_i \right)} \tag{21}$$

.

2.1.3. Distance quadratique

Cette distance a été proposée dans [73] pour évaluer des distances entre histogrammes de couleur in intégrant l'intervention d'une mesure de similarité entre les couleurs. Sa formulation originale est exprimée par (22) :

$$d_Q(X,Y) = \sum_{i=0}^{N}\sum_{j=0}^{N}(x_i - y_i)(x_j - y_j)a_{ij} \qquad (22)$$

Avec A matrice de similarité entre les couleurs i et j définie comme il suit

$$A = \lfloor a_{ij} \rfloor$$

Où

$$a_{ij} = 1 - \frac{d_{ij}}{d_{\max}}$$

d_{ij} étant la distance entre les couleurs quantifiées i et j et d_{max} le maximum global de cette distance

On peut la généraliser à une distance entre deux ensembles de même cardinalité (où a_{ij} serait remplacée par la distance entre deux éléments) ou entre deux vecteurs dont on souhaiterait pondérer l'importance de chaque composante.

2.1.4. Earth Mover Distance (EMD)

Introduite dans [76], la distance EMD mesure le coût de transformation de l'histogramme X en l'histogramme Y. Si la distance entre deux "bins" est définie par une autre distance adaptée aux données comparées (distance de Minkowski, distance spécifique, etc.), la distance entre deux ensembles est le minimum des sommes des distances de "déplacement" de chaque point de l'espace X. Comme précédemment d_{ij} représente la distance entre les bins i et j.

$$d_{EMD}(X,Y) = \frac{\sum_{i=0}^{N_X}\sum_{j=0}^{N_Y} d_{ij} f_{ij}}{\sum_{i=0}^{N_X}\sum_{j=0}^{N_Y} f_{ij}} \qquad (25)$$

Où on définit f_{ij} comme le "flot" optimal entre les deux ensembles, c'est-à-dire qu'il minimise d_{EMD}. Celui-ci est sujet aux contraintes suivantes :

$$f_{ij} \geq 0 \qquad 0 \leq i \leq N_X, 0 \leq j \leq N_Y$$

$$\sum_{i=0}^{N_X} f_{ij} \leq y_j \qquad 0 \leq j \leq N_Y$$

$$\sum_{j=0}^{N_Y} f_{ij} \leq x_i \qquad 0 \leq i \leq N_X$$

$$\sum_{i=0}^{N_X}\sum_{j=0}^{N_Y} f_{ij} = \min\left(\sum_{i=0}^{N_X} x_i, \sum_{j=0}^{N_Y} y_j\right)$$

Notons que cette mesure est efficace même avec des histogrammes de taille et de construction différente. On peut de cette façon l'appliquer pour comparer des signatures (éventuellement de tailles différentes) pour peu que la distance d soit définie. Cette mesure requiert par contre la résolution d'un problème d'optimisation linéaire, ce qui la rend notablement plus couteuse que les autres mesures présentées ici.

2.2. Distances entre distributions

Ces distances s'utilisent en considérant X et Y comme des distributions et impliquent donc que x_i et y_i soient des probabilités. Ces distances restent simplement applicables sur des histogrammes normalisés.

2.2.1. Divergence de Kullback Leibler

Il ne s'agit pas d'une distance à proprement parler au sens ou cette mesure n'en remplit pas les conditions de base. Issue de la théorie de l'information, elle mesure la différence d'entropie de X par rapport à Y.

$$d_{KL}(X,Y) = \sum_{i=0}^{N} x_i \log\left(\frac{x_i}{y_i}\right) \tag{22}$$

2.2.2. "Jensen difference divergence"

Définie dans [77] à partir de la divergence de Kullback Leiber, cette mesure retranscrit exactement la même information mais présente en plus l'avantage d'être symétrique.

$$d_J(X,Y) = \sum_{i=0}^{N}\left[x_i \log\left(\frac{x_i}{z_i}\right) + y_i \log\left(\frac{y_i}{z_i}\right)\right] \tag{23}$$

Avec

$$z_i = \frac{x_i + y_i}{2}$$

2.2.3. Test du χ^2

Il s'agit également d'une mesure qui n'est pas une distance à proprement parler. Le test du χ^2 teste l'hypothèse que les échantillons x_i observés sont issus de la population représentée par les y_i. On en déduit la distance entre X et Y.

$$d_{\chi^2}(X,Y) = \sum_{i=0}^{N} \frac{(x_i - z_i)^2}{z_i} \quad (24)$$

Avec

$$z_i = \frac{x_i + y_i}{2}$$

2.2.4. Distance de Bhattacharya

Cette distance est utilisée pour mesurer la séparabilité de deux distributions (elle est fortement liée à l'expression de l'erreur minimale de classification dans le cas de deux classes [78]). A noter que l'on parle de "distance" bien que cette mesure n'en vérifie pas les propriétés de base (elle est seulement symétrique).

$$d_{Bha}(X,Y) = \sum_{i=0}^{N} \sqrt{x_i y_i} \quad (25)$$

2.3. Distances relatives aux ensembles

Ces distances sont considérées pour mesurer des distances entre deux ensembles de dimension N_X et N_Y contenant des vecteurs x_i et y_i de dimension n. Ils nécessitent l'utilisation d'une autre distance $d(x_i,y_i)$ pour comparer les vecteurs entre eux.

2.3.1. Distance de Hausdorff

La distance de Hausdorff est une distance qui permet de mesurer la similarité de deux ensembles compacts de points. Cette mesure se calcule en recherchant pour chaque point de chaque ensemble la distance d entre ce point et le point le plus proche appartenant à l'autre ensemble ; la distance étant donnée par le maximum de d. Elle correspond donc à une dissimilarité maximale entre les ensembles.

$$d_H(X,Y) = \max\left[\max_{x_i \in X}\left(\min_{y_j \in Y}\left(d(x_i, y_j)\right)\right), \max_{y_j \in Y}\left(\min_{x_i \in X}\left(d(y_i, x_i)\right)\right)\right] \quad (26)$$

2.3.2. Distance de Mahalanobis

La distance de Mahalanobis permet calculer les distances d'un point par rapport à un ensemble en considérant les variances de l'ensemble selon ces axes. Elle permet donc de retranscrire des ensembles de forme ellipsoïde. Elle s'exprime ainsi au moyen de la matrice de covariance Ψ de l'ensemble considéré : si μ est son centre de gravité son expression est donnée par (27) :

$$d_{Mah}(p,X) = (p-\mu)^T \Psi (p-\mu) \quad (27)$$

2.3.3. Distance de Fischer

La distance de Fischer a pour but de comparer deux ensembles en mettant en rapport les variances à l'intérieur des ensembles et la distance entre leurs centres. De ce fait elle est appropriée pour déterminer si deux ensembles doivent fusionner. Elle s'exprime selon (28) où n_1 et n_2 représentent les populations des deux ensembles, μ_1 et μ_2 leurs centres de gravité et σ_1 et σ_2 leurs variances respectives.

$$D(X,Y) = \frac{(n_1 + n_2)(\mu_1 - \mu_2)^2}{n_1\sigma_1^2 + n_2\sigma_2^2} \tag{28}$$

2.4. Distances Spécifiques à la couleur

2.4.1. Principes

Comme nous l'avons déjà observé dans la section chapitre sur les espaces de couleur, les expériences mentionnées dans [19] montrent que même les espaces CIELab et CIELuv ne sont pas perceptuellement homogènes, ce qui signifie que la distance euclidienne entre deux couleurs ne permet pas de retranscrire la différence perçue par un humain. La Commission Internationale de l'Eclairage ainsi que des groupes industriels se sont donc attachés à créer des mesures reflétant plus précisément la perception humaine. Pour cela ils utilisent un espace présentant à la base une certaine homogénéité perceptuelle (CIELab) et utilisent une mesure de différence qui prenne en compte la forme ellipsoïde des régions homogènes dans cet espace. De nouvelles formules de plus en plus complexes voient le jour au cours des années, reflétant de mieux en mieux le jugement visuel humain.

2.4.2. Exemple : la distance CMC

A titre d'exemple nous allons aborder la distance CMC qui a été définie par l'industrie textile. Son but est de quantifier la notion de "Just Noticeable Difference" qui a pour but de déterminer le moment où la perception arrive à distinguer deux couleurs. Il ne s'agit pas de la plus récente ou de la plus efficace (voir les études effectuées dans [79]) mais elle est d'une part représentative des méthodes de calcul utilisées et d'autre part présente un bon rapport entre sa complexité et sa précision. Cette mesure n'est pas symétrique aussi par la suite on se place dans le cas où l'on cherche à déterminer la distance CMC notée ΔE entre une couleur de référence $C_{ref}(L_1, a_1, b_1)$ et une couleur qu'on lui compare $C_{comp}(L_2, a_2, b_2)$.

Le calcul de la distance commence avec celui de trois différences selon trois composantes qui sont considérées comme les axes de l'ellipsoïde au sein de laquelle deux couleurs sont considérées comme perceptuellement similaires. Ces axes sont disposés selon les composantes de l'espace $CIELch_{Lab}$. On définit donc trois différences selon chaque composante par ΔL, ΔC et ΔH qui se calculent selon (29, 30, 31) :

$$\Delta L = L_1 - L_2 \tag{29}$$

$$\Delta C = \sqrt{a_1^2 + b_1^2} - \sqrt{a_2^2 + b_2^2} \tag{30}$$

$$\Delta H = \sqrt{(\Delta a)^2 + (\Delta b)^2 - (\Delta C)^2} \tag{31}$$

Δa, Δb et ΔC désignant respectivement les différences a_2- a_1, b_2- b_1 et C_2-C_1.

Viennent ensuite 3 coefficients S_l, S_c et S_h, déterminés expérimentalement et qui pondèrent ces trois différences en intégrant des comportements différents pour des luminances faibles et certaines teintes par rapport à d'autres.

$$S_L = \begin{cases} 0.511 \quad pour \quad L_1 < 16 \\ \dfrac{0.040975 \cdot L_1}{1 + 0.01765 \cdot L_1} \quad pour \quad L_1 \geq 16 \end{cases} \tag{32}$$

$$S_C = \frac{0.0638 \cdot C_1}{1 + 0.0131 \cdot C_1} + 0.638 \tag{33}$$

$$S_H = S_C\left(FT + 1 - F\right) \tag{34}$$

Avec

$$F = \sqrt{\frac{C_1^4}{C_1^4 + 1900}}$$

$$T = \begin{cases} 0.56 + \left|0.2\cos\left(H_1 + 168\right)\right| \quad pour \quad 164 \leq H_1 \leq 345 \\ 0.36 + \left|0.4\cos\left(H_1 + 35\right)\right| \quad autrement \end{cases}$$

On notera à cette occasion que l'angle H_1 s'exprime en degrés.

On définit enfin deux paramètres pondérant directement l'importance de la luminance et de la chroma : l et c. Ces paramètres sont habituellement spécifiés dans le nom de la mesure sous la forme CMC(l :c). Les mesures les plus utilisées sont les mesures CMC(2 :1) et CMC(1 :1) qui correspondent respectivement à des distances pour lesquelles 1 est respectivement le seuil d'acceptabilité (i.e. : en deçà de 1 on tolère que ces couleurs soient considérées comme identiques) et le seuil de perception (i.e. : en deçà de 1 on considère que ces couleurs sont perceptuellement identiques : c'est le seuil de la Just Noticeable Difference). Ayant défini tous ces paramètres, la distance CMC(l :c) s'exprime comme noté dans (35).

$$\Delta E = \sqrt{\left(\frac{\Delta L}{l \cdot S_L}\right)^2 + \left(\frac{\Delta C}{c \cdot S_C}\right)^2 + \left(\frac{\Delta H}{S_H}\right)^2} \tag{35}$$

2.4.3. Utilisations ?

La question de la pertinence de l'utilisation de telles mesures dans une application de traitement d'images numériques se pose. En effet nous avons vu plus haut que l'utilisation

même de l'espace CIELab exigeait, en toute rigueur, une expression de la couleur sur un nombre plus importants de bits que ce que nous permettent les images numériques actuelles. Or ces mesures vont non seulement bien plus loin mais présentent une couche supplémentaire de calculs qui vont alourdir d'autant les calculs qui les utilisent. On ne peut donc raisonnablement envisager l'utilisation de telles formules qu'après évaluation de leur apport par rapport à des mesures moins coûteuses et dans des tâches "offline" ou pour des couleurs dont les coordonnées dans l'espace Lch ont déjà été calculées pour d'autres besoins.

3. Clustering

Rassembler des points "proches" dans différents ensembles (ou "clusters") au sein d'un espace de dimension n est un problème récurrent qui est apparu à tous les niveaux de nos travaux. Les problématiques principales sont :

- La pertinence des clusters produits (conformité aux résultats attendus, éventuelle robustesse au bruit, …)
- La nécessité d'un travail d'initialisation et la sensibilité des résultats à cette initialisation
- La nécessité ou non de déterminer un nombre de clusters et, le cas échéant, le mode de détermination du nombre de clusters
- La complexité de la tâche de clustering (temps d'exécution)

La pertinence des clusters produit est une évaluation dépendante de l'application tout comme l'importance de la complexité du clustering (qui reste toutefois un facteur limitant sur la taille de données). Nous allons donc étudier les algorithmes de clustering en fonction de leur capacité à déterminer le nombre de clusters à produire. Il est par ailleurs évident que l'on ne peut prétendre à l'exhaustivité dans ce domaine qui est l'objet de très nombreuses recherches. On mentionnera donc des approches particulièrement répandues et/ou représentatives d'un mode de traitement des données (voir les études complètes dans [80] et [81]).

Dans cette section on considèrera un ensemble de N vecteurs x_i à composantes réelles et de dimension n. Le nombre de clusters défini (cas d'un nombre de clusters fixe) ou à une itération donnée sera noté β, les centres des différents clusters seront notés c_i.

3.1. Considérations préliminaires

Avant de partitionner un espace de données il y a un certain nombre d'aspects à considérer. Le premier point à considérer se situe en amont de la tâche de clustering et concerne l'expression même des données. Le nombre de dimensions de l'espace de clustering est en effet une donnée critique par rapport à la qualité du clustering. Le problème de la "curse of dimensionality" [14] affecte doublement les algorithmes de clustering ; d'abord par une augmentation du coût calculatoire des opérations (ce qui est d'autant plus critique que les algorithmes de clustering sont basés sur de multiples itérations) mais aussi parce que les distances classiques (comme la distance euclidienne) deviennent inutiles. En effet dans un espace de très grande dimension presque toutes les paires de données constituables sont quasiment à la même distance (les points considérés étant à une distance qui sera la racine carrée de la somme des carrés de différences qui s'équilibrent entre elles). Il est donc essentiel de travailler avec des données de taille raisonnable ou d'utiliser un procédé de réduction des dimensions (Analyse Discriminante, …).

Le second point à considérer concerne l'échelle des données manipulées. Celle-ci pose directement le problème de la normalisation. Si en théorie la plupart des algorithmes de clustering peuvent s'affranchir d'une étape de normalisation, des problèmes d'implémentation peuvent se poser sous la forme des arrondis numériques. Le choix d'une méthode de normalisation des données est donc une première étape préalable à l'application d'un algorithme de clustering. Un autre problème que l'on peut rencontrer en l'absence de normalisation est que, dans le cas de l'utilisation d'une distance de Minkowski, les composantes de grande valeur absolue auront tendance à écraser les autres.

Le dernier point à examiner est tout simplement celui de la distance à employer. En effet l'utilisation classique de la distance euclidienne se fait bien souvent par défaut, mais il faut considérer que se baser sur une distance Euclidienne ne permettra, sur certains algorithmes, de ne produire que des clusters sphériques. L'utilisation, si possible, d'une mesure plus adaptée au problème (mesure spécifique au type de données manipulé, distance de Mahalanobis, etc.) peut produire des clusters plus pertinents.

3.2. Clustering vers un nombre fixe de clusters

3.2.1. Clustering Hiérarchique

a) Principe et utilisation

Par définition les algorithmes de clustering hiérarchique peuvent d'abord être ascendants c'est-à-dire à partir d'un ensemble de clusters (par exemple un cluster par vecteur) en agglomérer deux pour diminuer de 1 le nombre de clusters, cette opération étant répétée jusqu'à l'obtention du nombre désiré de clusters. Ces algorithmes peuvent au contraire être descendants, partant d'un cluster unique un cluster étant divisé en deux à chaque itération jusqu'à obtention du nombre de clusters souhaité. Les algorithmes de type ascendant sont très majoritaires car il est beaucoup plus simple de trouver des critères pour regrouper des clusters que pour les diviser (la plupart des algorithmes existants reposent sur une Analyse en Composantes Principales). Comme nous allons le voir ces algorithmes présentent l'avantage d'être parfaitement déterministes (pas d'initialisation ni d'aléatoire).

b) Description de l'algorithme

La plupart des algorithmes de ce type actuellement utilisés sont dérivés de l'algorithme "single-link" de Sneath et Sokal [82] ou de l'algorithme "complete-link" de King [83] (cités dans [80]). Ces deux algorithmes sont de type ascendant, la Figure 18 ("dendrogramme" issu de [80]) illustre leur fonctionnement : la ligne en pointillées montre l'itération à laquelle il faudrait idéalement arrêter l'algorithme. Les procédés "single-link" et "complete-link" diffèrent uniquement dans leur manière de constituer des paires de clusters. Dans "single-link" la distance entre deux clusters est le minimum des distances entre toutes les paires d'éléments issus de chaque cluster. Dans "complete-link" cette distance est le maximum. Cette différence est importante au sens où "complete-link" aura tendance à produire des clusters compacts alors que "single-link" sera vulnerable à des problèmes de type "chaînage" où un même cluster va absorber des éléments lointains de proche en proche (voir Figure 19 , extraite de [80]). On notera que cette catégorie d'algorithmes peut déterminer dynamiquement le nombre de clusters au moyen d'un seuil qui serait posé sur la distance maximale entre deux clusters à fusionner.

c) Complexité

Il faut noter que ces algorithmes calculent un nombre important de distances à chaque itération puisqu'il s'agit de calculer toutes les paires de clusters possibles ce qui, sans prétraitement préalable, veut dire qu'avec N clusters à l'itération 1 on aura un nombre d'opérations η exprimé en (36) :

$$\eta = \sum_{j=N}^{\beta} C_j^2 = \frac{N\ !}{2(N-2)!} + \ldots + \frac{\beta\ !}{2(\beta-2)!} = \frac{\sum_{j=N}^{\beta}\left(j^2 - j\right)}{2} \qquad (36)$$

La complexité par itération étant donc de l'ordre de n^2 et la complexité totale en $n^2\log(n)$.

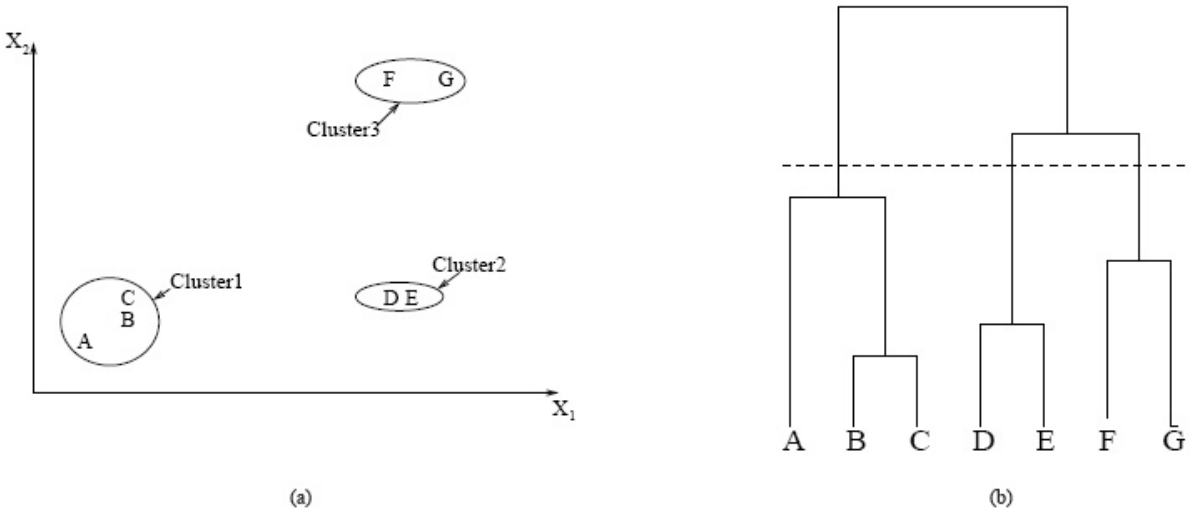


Figure 18: Exemple de clustering hierarchique ;
(a) données d'origine,
(b) ordre de regroupement (de bas en haut)

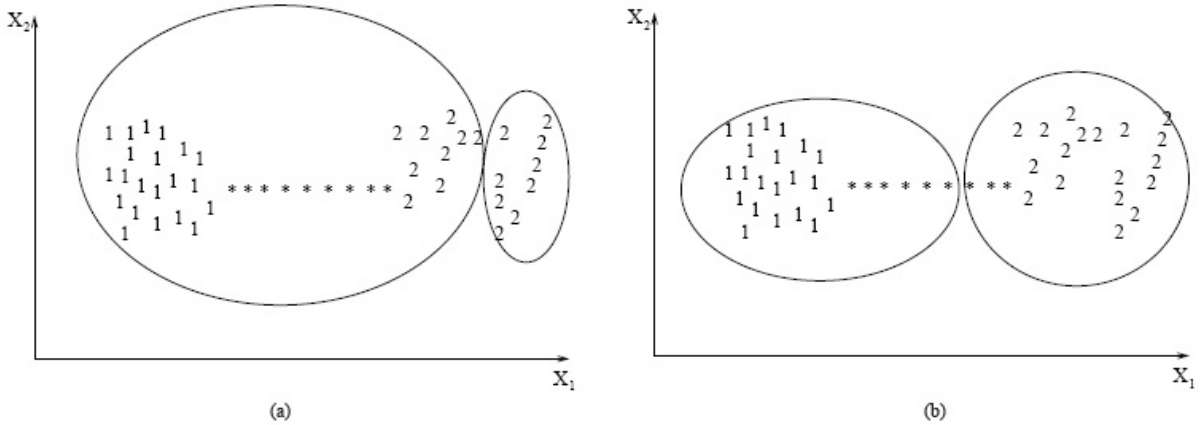


Figure 19: Illustration du problème du "chainage";
(a) "single-link",
(b) "complete link"

3.2.2. K-Means

a) Principe et utilisation

Probablement l'algorithme de clustering le plus répandu. Il se base sur une initialisation préalable des centres de chaque cluster puis les affine au fil d'itérations. L'initialisation est une étape particulièrement importante car tout cluster ignoré par l'initialisation sera absorbé par le cluster le plus proche. Cet algorithme est simple à mettre en œuvre et, comme nous le verrons, présente une faible complexité en termes de calcul.

b) Description de l'algorithme

Le déroulement de l'algorithme est le suivant :

- Association de chaque vecteur au centre du cluster le plus proche
- Calcul des nouvelles coordonnées du centre par rapport aux vecteurs qui y sont associés

Les itérations se poursuivent jusqu'à ce qu'on atteigne une condition terminale, qui peut être un nombre fixe "i" d'itérations, descente du nombre de vecteurs assignés à un autre cluster sous un nombre minimal ou encore des conditions sur l'erreur quadratique entre un centre et les vecteurs qui y sont associées (oscillations, arrivée à un minimum, …).

c) Complexité

Le nombre de calculs de distance effectués ici est très simple puisqu'à chaque itération on va calculer la distance de chaque point à chaque centre soit $N \cdot C$ calculs de distance.

3.2.3. Fuzzy C-Means (FCM)

a) Principe et utilisation

Evolution du K-Means, il fait intervenir l'appartenance floue des points à un cluster. Comme nous le verrons la complexité de l'algorithme reste plus faible mais il permet un clustering plus fin puisqu'on détermine pour chaque point un degré d'appartenance aux différents clusters. Cette information plus nuancée que l'appartenance définitive à une classe peut se révéler utile pour des traitements ultérieurs de l'information. Cet algorithme garde, par contre, une forte sensibilité à l'initialisation tout comme le K-Means.

b) Description de l'algorithme

La phase d'initialisation est identique, mais va s'accompagner de l'initialisation de valeurs d'appartenance u_{ij} du vecteur x_i au cluster de centre c_j. Ces valeurs sont comprises entre 0 et 1 dans le cas du Fuzzy C-Means ; on notera juste l'existence d'autres algorithmes flous utilisant d'autres fonctions d'appartenance. Après cette phase d'initialisation, la résolution du problème de clustering se fait par la minimisation de la fonction objectif définie par (37), m étant un paramètre constant supérieur à 1 (généralement pris égal à 2).

$$J_m = \sum_{i=0}^{N} \sum_{j=0}^{\beta} u_{ij}^m d^2\left(x_i, c_j\right) \qquad (37)$$

Ceci se traduit dans les deux étapes qui seront répétées itérativement : la mise à jour des appartenances de chaque vecteur à chaque cluster (38):

$$u_{ij} = \frac{1}{\sum_{k=0}^{\beta} \left(\frac{d(x_i, c_j)}{d(x_i, c_k)} \right)^{\frac{2}{m-1}}} \quad (38)$$

…et la mise à jour des centres de chaque cluster (39):

$$c_j = \frac{\sum_{i=0}^{N} u_{ij}^m \cdot x_i}{\sum_{i=0}^{N} u_{ij}^m} \quad (39)$$

Ici encore les critères d'arrêt de l'algorithme varient, le plus rependu étant une absence d'évolution (le cas échéant l'oscillation) des appartenances u_{ij}.

c) Complexité

Les formules de calcul des centres sont certes différentes et on rajoute le calcul des appartenances, mais les opérations de base demeurant les mêmes, la complexité globale reste, quant à elle, le même que pour un algorithme de type K-Means.

3.2.4. Gaussian Mixture Models (GMM) et Expectation Maximization (EM)

a) Principe et utilisation

On assimile ici les clusters à des gaussiennes ce qui suppose que la répartition globale des vecteurs dans l'espace suit une répartition générée par un ensemble de gaussiennes. Ce modèle est particulièrement intéressant puisque, plus qu'un clustering, il effectue une modélisation de la distribution et peut donc permettre de générer ultérieurement des données conformes au modèle obtenu. On doit pour cela estimer les paramètres de ces gaussiennes, à savoir leurs centres c_j, leurs covariances Σ_j ainsi que leurs importances relatives π_j en étudiant la densité des vecteurs et en calculant le maximum de vraisemblance pour chaque paramètre. L'algorithme le plus utilisé pour cela est l'algorithme EM (Expectation Maximization) [84] qui permet de résoudre des situations d'optimisation avec des données inconnues. Nous allons brièvement en étudier l'application spécifique à ce problème.

b) Description de l'algorithme

Cet algorithme repose sur une optimisation itérative qui se déroule en deux étapes "Expectation" et "Maximization". On part d'une solution initiale, pour laquelle on calcule les centres, les covariances et les coefficients. On en calcule le log-vraisemblance (42) On applique ensuite à chaque itération les étapes suivantes:

- Etape "Expectation" : evaluations des contributions entre chaque gaussienne et chaque vecteur x_i avec les paramètres actuels :

$$\gamma_{ij} = \frac{\pi_j N\left(x_i \middle| c_j, \Sigma_j\right)}{\sum_{k=0}^{\beta} \pi_k N\left(x_i \middle| c_k, \Sigma_k\right)} \tag{40}$$

On note $N(x_i|c_j, \Sigma_j)$ la valeur de la gaussienne de centre c_j, de matrice de covariance Σ_j au point x_i.

- Etape "Maximization", on cherche une nouvelle solution pour les trois paramètres :

$$\begin{aligned} c_j^{NEW} &= \frac{1}{N_j} \sum_{i=0}^{N} \gamma_{ij} x_i \\ \Sigma_j^{NEW} &= \frac{1}{N_j} \sum_{i=0}^{N} \gamma_{ij} \left(x_i - c_j^{NEW}\right)\left(x_i - c_j^{NEW}\right)^T \\ \pi_j^{NEW} &= \frac{N_j}{N} \end{aligned} \tag{41}$$

Avec

$$N_j = \sum_{i=0}^{N} \gamma_{ij}$$

- Calcul du log-vraisemblance du modèle conditionné par les valeurs observées (notées X).

$$\ln\left(p\left(X \middle| c, \Sigma, \pi\right)\right) = \sum_{i=0}^{N} \ln\left[\sum_{j=0}^{\beta} \pi_j N\left(x_i \middle| c_j, \Sigma_j\right)\right] \tag{42}$$

La convergence de l'algorithme se détermine en observant l'évolution des paramètres et du log-vraisemblance.

c) Complexité

Le calcul d'une itération implique le calcul de chaque vecteur par rapport chaque gaussienne soit βN étapes puis les sommations et le calcul du log-vraisemblance elles aussi avec une complexité en βN. D'autre part, l'algorithme EM prend un nombre relativement important d'itérations pour converger, et d'autre part chaque étape est relativement lourde comparée à un algorithme comme K-Means. Par conséquent la phase d'initialisation peut être réalisée par un K-Means qui produira une partition permettant de calculer simplement des valeurs initiales pertinentes de c_j et Σ_j. π_j peut être choisi comme étant la proportion de points assignés à ce cluster.

3.3. Détermination automatique du nombre de clusters

3.3.1. Dynamic Local Search (DLS)

a) Principe et utilisation

Cet algorithme, proposé par Karkkainen et Franti dans [85] est basé sur une fonction d'évaluation de qualité du clustering f(X, Cl, P) où X est l'ensemble des vecteurs, Cl l'ensemble des centres des clusters et P l'ensemble des partitions de l'espace constituées par l'opération de clustering. L'algorithme faisant évoluer aléatoirement les clusters en fonction de l'évaluation donnée par f. Cet algorithme est une amélioration de l'algorithme Random Local Search [86] qui travaille avec un nombre fixe de clusters cible et illustre la classe des algorithmes évolutionnaires qui contient par ailleurs des méthodes basées sur des algorithmes comme les algorithmes génétiques. Ces algorithmes ont une évolution aléatoire, ce qui rend moins probable la convergence vers des optimums locaux et diminue la sensibilité à l'initialisation mais rend aussi les partitions moins reproductibles et leur exécution plus lente.

b) Description de l'algorithme

Les étapes de l'algorithme DLS sont les suivantes : étant donnés un nombre minimal et maximal de clusters possibles, on génère une première solution. Ceci peut par exemple se faire simplement avec répartition aléatoire des centres et génération de la partition en affectant les vecteurs aux centres les plus proches. A partir de là on entame le procédé évolutionnaire que l'on répète jusqu'à satisfaction d'une condition de sortie :

- On génère de nouveaux centres Cl_{new} à partir d'une *opération* sur Cl
- On affine Cl_{new} et on définit la partition correspondante P_{new} via l'exécution d'un Generalized Lloyd Algorithm (noté GLA qui est une variante de l'algorithme K-Means)
- Si f(X, Cl, P) est meilleure que f(X, Cl_{new}, P_{new}) alors on remplace Cl par Cl_{new} et P par P_{new}.

Les *opérations* possibles étant une parmi les 3 suivantes :

- "Echange" : Suppression d'un centre et création d'un autre centre à un emplacement aléatoire
- "Création" : Création (dans les limites de nombre de clusters données) d'un ou plusieurs centres générés aléatoirement à partir de l'ensemble X
- "Suppression" : Suppression (dans les limites de nombre de clusters données) d'un ou plusieurs centres parmi l'ensemble Cl

L'opération effectuée étant choisie aléatoirement. Les auteurs suggèrent des probabilités de 50% pour un échange, 25% pour une création et 25% pour une suppression. La condition de sortie suggérée est de considérer, après un nombre minimal d'itérations, un seuil sur l'amélioration (selon f) apportée par la dernière moitié d'itérations divisé par l'amélioration apportée par la première moitié d'itérations.

c) Complexité

La complexité de cet algorithme est ici relativement importante, puisqu'on exécute un algorithme de type K-Means par itération. Les auteurs ne donnent pas précisément de quantité moyenne d'itérations avant d'obtenir un résultat mais montrent tout de même que 1000 itérations sont nécessaires pour des problèmes de taille assez modeste (50 clusters circulaires pour un nombre non défini de points dans un espace à deux dimensions).

3.3.2. Competitive Agglomeration (et ses variantes)

a) Principe et utilisation

Cet algorithme, proposé par Frigui et Krishnapuram [87] est une extension du Fuzzy C-Means présenté plus haut avec introduction d'une diminution progressive du nombre de clusters en fonction de la somme des appartenances des vecteurs aux clusters. Nous verrons qu'il en garde la complexité tout en permettant d'évaluer un nombre optimal de clusters

b) Description de l'algorithme

La fonction objectif définie pour le fuzzy C-Means (37) avec m = 2 est complétée par une fonction qui traduit la volonté de l'algorithme à réduire le nombre de classes, la fonction objectif complète pour l'algorithme CA s'écrit donc (43) :

$$J = \sum_{i=0}^{N}\sum_{j=0}^{\beta} u_{ij}^2 d^2\left(x_i, c_j\right) - \alpha \sum_{j=0}^{\beta}\left[\sum_{i=0}^{N} u_{ij}\right]^2 \qquad (43)$$

α étant un paramètre évoluant au cours des itérations (46)

La résolution du problème de minimisation ainsi posé donne de nouvelles formules pour les formules de calcul des appartenances (44) :

$$u_{ij} = u_{ij}^{FCM} + u_{ij}^{Biais} \qquad (44)$$

Avec u_{ij}^{FCM} exprimé comme selon (38) avec m=2 et u_{ij}^{Biais} exprimé comme défini en (45) :

$$u_{ij}^{Biais} = \frac{\alpha}{d^2\left(x_i, c_j\right)}\left(N_j - \overline{N}_i\right) \qquad (45)$$

$$N_j = \sum_{k=0}^{N} u_{kj}$$

$$\overline{N}_i = \frac{\sum_{k=0}^{\beta} \frac{1}{d^2\left(x_j, c_k\right)} N_k}{\sum_{k=0}^{\beta} \frac{1}{d^2\left(x_i, c_k\right)}}$$

Avec α poids de la compétition (apparaissant plus haut et défini ci-dessous par 46)

La conséquence de cette compétition est que les classes peu représentatives auront tendance à se dépeupler (N_j faible) au profit des autres classes. Les classes ayant de trop faibles valeurs de N_j étant supprimées.

La compétition entre l'optimisation des clusters (par fuzzy C-Means) et l'optimisation du nombre de clusters est régulée par le paramètre α que nous avons précédemment évoqué. Celui-ci se calcule à l'itération κ selon (46) :

$$\alpha = \eta_0 e^{\frac{-\kappa}{\tau}} \frac{\sum_{j=0}^{\beta}\sum_{i=0}^{N} u_{ij}^2 d^2\left(x_i, c_j\right)}{\sum_{j=0}^{\beta}\left[\sum_{i=0}^{N} u_{ij}\right]^2} \qquad (46)$$

Avec le taux de décroissance τ et l'amplitude η_0 comme paramètres constants de l'algorithme.

On voit donc par rapport à (43) et (46) que, α décroissant l'algorithme va d'abord donner la priorité à l'élimination de clusters avant de progressivement se transformer en fuzzy C-Means classique. Les valeurs des centres sont mises à jour comme pour un fuzzy C-Means classique : selon (39) et avec m = 2 et l'appartenance u_{ij} définie en (44, 45, 46).

Le déroulement global de l'algorithme débute par une phase d'initialisation :

- Initialisation des centres pour β classes β étant choisi comme maximal
- Initialisation de la matrice d'appartenances comme pour un Fuzzy C-Means
- Initialisation des N_i

Débute alors le procédé itératif :

- Incrémentation de κ
- Mise à jour des distances $d^2(x_i, c_j)$, de α (45), des appartenances u_{ij} (43,44,45), des populations N_i
- Suppression des classes telles que $N_i<\varepsilon$
- Si deux centres sont trop proches, les fusionner
- Mise à jour du nombre de classes β et des centres c_j
- Calcul des deux critères de convergence k_1 et k_2 (47)

Où ε est une constante, paramètre de la méthode et k_1 et k_2 sont calculés comme il suit (47). L'algorithme est stoppé si k_1 et k_2 sont faibles.

$$\begin{aligned} k_1 &= \max_{i,j}\left(u_{ij}^{Biais}\right) \\ k_2 &= \max_{j}\left(\left\| c_j^{\kappa+1} - c_j^{\kappa} \right\|\right) \end{aligned} \qquad (47)$$

c) Complexité

La complexité de l'algorithme reste celle d'un fuzzy C-Means. On notera que cet algorithme a fait l'objet de diverses améliorations comme l'utilisation de statistiques robustes [88] et l'adjonction d'un cluster "bruit" où sont regroupées les données incohérentes

combinée une adaptation de la mesure d'appartenance pour éviter l'élimination de clusters de faible densité [89].

3.4. Explorations de distributions

Cette dernière catégorie est un peu à part : ces algorithmes permettent de découvrir une distribution pour en fournir une représentation. Si les trois exemples évoqués par la suite ne représentent pas la totalité des approches disponibles dans cette catégorie, chacun des algorithmes présentés ici présente un intérêt particulier qui sera précisé.

3.4.1. Self organizing maps (SOM)

a) Principe et utilisation

Cet algorithme proposé par Kohonen [90] permet de projeter un espace de dimension n sur un espace de dimension 2 ce qui en facilite la description et éventuellement la représentation. Initialement cet espace est recouvert par des nœuds disposés selon une grille le plus souvent hexagonale ou rectangulaire. Le nombre de nœuds étant inférieur à N et de préférence relativement important pour bien décrire la distribution à modéliser. Chaque nœud est associé à un "poids" qui est un vecteur de dimension n et initialisé soit de manière aléatoire, soit organisés selon la projection d'un échantillon représentatif des données à explorer sur les deux meilleurs axes d'une analyse en composantes principales (auquel cas la distribution sera explorée plus rapidement : les nœuds ayant déjà des poids décrivant bien la distribution).

b) Description de l'algorithme

L'exploration de la distribution se déroule comme il suit : on choisit aléatoirement un vecteur parmi l'ensemble des vecteurs de la distribution et on calcule la distance d entre ce vecteur et les poids de tous les nœuds, on obtient ainsi un nœud "gagnant" noté Best Matching Unit (BMU). Etant donnée la BMU on détermine un voisinage autour d'elle dans l'espace projeté ; ce voisinage peut se créer en sélectionnant les nœuds directement adjacents, dans un voisinage gaussien, dans un cercle de rayon défini... On fait alors évoluer les poids de la BMU et du voisinage ainsi déterminé selon (48) :

$$W_v(t+1) = W_v(t) + \varphi(d(v))\alpha(t)(D(t) - W_v(t)) \qquad (48)$$

Avec $W_v(t)$ désignant le poids d'un vecteur v à l'itération t. La fonction φ désigne une fonction de la distance entre le noeud v et la BMU (pour éventuellement affecter). La fonction a est une fonction monotone décroissante dans le temps ; ainsi l'évolution est de moins en moins importante et la représentation à tendance à se stabiliser. Enfin D(t) désigne le vecteur de la distribution qui a été choisi à cette itération.

c) Complexité

La complexité de l'algorithme dépend essentiellement du nombre d'itérations "i" et du nombre de nœuds C choisis pour la représentation. Le nombre d'itérations recommandé est de faire en sorte que chaque vecteur de la distribution x_i puisse être utilisé "plusieurs fois" aussi trouve-t-on des nombres d'itérations de l'ordre de $k \cdot N$. Etant donné que l'on vérifie à chaque itération les distances entre le vecteur sélectionné et tous les nœuds, le nombre de distances

calculées sera de l'ordre de k.N.β. Le nombre de nœuds étant suggéré grand, tout comme le nombre d'itérations, le temps de calcul est un peu plus important qu'il n'y parait.

3.4.2. Neural Gas (NG)

a) Principe et utilisation

L'algorithme Neural Gas introduit par Martinetz et Schulten [91] s'inspire des principes de fonctionnement des SOM : on choisit un nombre arbitraire de nœuds qui découvrent la distribution au moyen de l'introduction d'exemples choisis un par un. L'intérêt en est par contre très différent : les nœuds ne se situent pas dans un espace en deux dimensions mais dans l'espace d'expression de la distribution originale. Cet algorithme produit donc une simplification des données fournies et en choisissant un nombre de vecteurs égaux à un nombre estimé de clusters, on peut assimiler les données produites à des centres. Cet algorithme peut donc être utilisé à la manière d'un algorithme de clustering auquel on fournirait initialement le nombre de clusters.

b) Description de l'algorithme

On part d'une génération aléatoire de nœuds dans l'espace des vecteurs à explorer. Le principe étant, à partir de là, le même que pour les SOM : on choisit tour à tour des vecteurs de la distribution à explorer, les nœuds convergent vers ce nouvel exemple et la convergence décroit avec le temps. Dans ce cas, toutefois, l'ensemble des nœuds converge vers le vecteur exemple, mais la convergence Δc_i du nœud d'index "i" est fonction de sa proximité à l'exemple (49).

$$\Delta c_i = \varepsilon(t) \cdot h_\lambda\left(k_i(\xi, A_t)\right) \cdot (\xi - c_i) \qquad (49)$$

Avec

$$h_\lambda(k) = e^{\left(\frac{-k}{\lambda(t)}\right)} \text{ où } \lambda(t) = \lambda_i\left(\lambda_f / \lambda_i\right)^{\frac{t}{t_{\max}}}$$

et

$$\varepsilon(t) = \varepsilon_i\left(\varepsilon_f / \varepsilon_i\right)^{\frac{t}{t_{\max}}}$$

ξ est le vecteur exemple fourni. A_t l'ensemble des nœuds à l'itération t. La fonction $k_i(\xi,A_t)$ associe à un vecteur exemple et un ensemble de nœuds la position du nœud d'index "i" dans la liste des nœuds triée par ordre de proximité au vecteur exemple selon la distance définie "d". t et t_{max} désignent respectivement l'itération en cours et la limite choisie pour le nombre d'itérations. Enfin les valeurs ε_i, ε_f, λ_i et λ_f (i pour initial et f pour final) sont des paramètres du modèle. Il faut noter que ces paramètres ne sont pas critiques pour l'obtention d'une bonne solution, les valeurs suggérées à partir de [92] sont $\varepsilon_i = 0.5$, $\varepsilon_f = 0.005$, $\lambda_i = 10$ et $\lambda_f = 0.01$. On notera enfin que les importants déplacements Δw_i aux premières itérations rendent l'algorithme peu sensible à l'initialisation en pratique.

c) Complexité

A chaque itération on effectue une mesure de distance entre le vecteur exemple et tous les nœuds. Cette mesure est suivie par un tri des nœuds par ordre de distance au vecteur

exemple. La complexité d'une itération est donc de $\beta + \beta.\log(\beta)$. Le nombre d'itérations total t_{max} est défini de la même façon que les SOM, à savoir que l'on veut faire en sorte que chaque vecteur de la distribution x_i puisse être utilisé "plusieurs fois" comme exemple. On obtient donc une complexité de type $k.N.(\beta + \beta.\log(\beta))$. Sachant qu'ici β peut être plus petit que pour des SOM.

3.4.3. Growing Neural Gas (GNG)

a) Principe et utilisation

Cet algorithme introduit par Fritzke [93] reste sur le principe d'apprentissage défini par les SOM et le NG. A savoir la production successive de vecteurs issus de la distribution à modéliser qui entraine une convergence de nœuds vers ce vecteur exemple. L'originalité introduite par l'algorithme GNG est que le nombre de nœuds évolue avec le temps et que les nœuds sont reliés entre eux et forment ainsi des clusters si les données d'origine sont suffisamment séparées. On peut notamment utiliser cet algorithme pour obtenir un graphe à partir de la distribution et effectuer une tache de clustering basée sur les graphes (cités dans [80]).

b) Description de l'algorithme

L'algorithme s'initialise par deux nœuds positionnés aléatoirement et reliés entre eux. Comme on l'a dit, à chaque itération on va choisir un vecteur x_i issu de la distribution à explorer. Et modifier les nœuds et les relations entre eux :

- On détermine dans un premier temps, les deux nœuds les plus proches w_s et w_t (w_s étant le plus proche). La convergence se limitera aux points connectés à w_s.
- On met à jour l'erreur associée au nœud w_s en y ajoutant $d(x_i,w_s)$
- On met à jour la position de w_s et des nœuds qui y sont connectés w_n (50) :

$$\begin{aligned} w_s &= w_s + e_w(x_i - w_s) \\ w_n &= w_n + e_n(x_i - w_n) \end{aligned} \qquad (50)$$

e_w et e_n sont des paramètres de convergence de la méthode.

- On incrémente l'âge des liaisons entre w_s et ses voisins
- On crée une liaison d'âge 0 entre w_s et w_t où on met l'âge de la liaison existante à 0
- On efface toute liaison d'âge supérieur à a_{max} (paramètre de la méthode)

Toutes les λ itérations on insère un nouveau nœud w_r entre le nœud de plus grande erreur w_e et le nœud w_s relié à w_e et d'erreur maximale à la position $(w_s+w_e)/2$:

- On supprime la liaison entre w_s et w_e et on crée deux liaisons d'âge 0 entre w_r et w_s puis entre w_r et w_e.
- On met à jour les erreurs de w_e, w_r et w_s. Les erreurs de w_e et w_s sont multipliées par un coefficient α (inférieur à 1) et l'erreur de w_r est considérée égale à l'erreur mise à jour de w_e.
- On met à jour les erreurs de l'ensemble des nœuds du graphe en les diminuant selon un facteur γ

On notera que la valeur de λ peut être constante ou être déterminée en fonction d'autres paramètres comme l'erreur globale.

c) Complexité

A chaque itération, on va calculer la distance entre le vecteur choisi et l'ensemble des nœuds afin de déterminer les deux plus proches à partir de là les opérations de mise à jour des erreurs et de déplacement se font directement. On a donc une complexité de l'ordre du nombre de nœuds. Lors d'une insertion de nœud, on doit rechercher le nœud d'erreur maximale et mettre à jour les erreurs ce qui est également une opération de complexité linéaire par rapport au nombre de nœuds. Le nombre d'opérations nécessaire sera donc de l'ordre de $k \cdot N \cdot \beta$. Le nombre d'itérations sera par contre nettement plus important que pour un NG car il est important de laisser l'algorithme bien explorer l'espace avant de rajouter un nœud [93].

3.5. Discussion

Nous avons ici fait un petit inventaire des algorithmes de clustering disponibles. Comme noté en introduction, ces algorithmes sont très importants dans notre démarche puisque nous manipulons des volumes de données importants et que ceux-ci nous donnent l'opportunité de les réduire, sinon de les analyser en dégageant des groupes. Cette liste résume un ensemble de méthodes de clustering ou de modélisation dont elle illustre les principes mais elle n'est pas exhaustive : citons par exemple le critère MDL (Minimum Description Length), introduit en 1978 par Jorma Rissanen (décrit dans [140]) comme autre algorithme de modélisation dynamique (détermination automatique du nombre de clusters). Son objectif est de déterminer un codage idéal pour un ensemble de données selon le principe que toute régularité dans les données à coder est utilisable pour les compresser. Cet algorithme va comparer différents modèles en mettant en balance leur simplicité et leur capacité à décrire les données. Notre but etait ici d'évoquer différents axes d'approche du problème de partition d'un espace en clusters.

Etant données la grande utilité de ces algorithmes et la grande variété des approches disponibles, il est intéressant de noter que chacun des algorithmes évoqués ici à son domaine d'application : une tâche offline pourra utiliser des algorithmes complexes mais performants comme le DLS, la connaissance ou non du nombre de clusters cible guide elle aussi le choix de l'algorithme en rend des algorithmes.

En ce qui concerne nos travaux, quelques points retiennent tout particulièrement notre attention. Le premier est la robustesse à l'initialisation, qui est importante quand on travaille dans un espace de grandes dimensions et/ou lorsque les données sont très réparties dans l'espace : si on considère également la complexité des algorithmes, il s'agit d'un point fort de l'algorithme "Neural Gas". Le second est le déterminisme et la garantie de convergence, souhaitables dans des algorithmes que l'on souhaite fiables et répétables et qui attire notre attention sur les algorithmes de type "complete link". Enfin l'ignorance du nombre de clusters cible est un problème récurrent qui rend l'algorithme "Competitive Agglomeration" particulièrement intéressant même s'il est pénalisé par le fait qu'il ne soit pas déterministe.

4. Conclusion

Dans ce chapitre nous avons fait l'inventaire de différentes techniques appliquées à l'analyse d'images. Si elles présentent toutes des qualités et des défauts nous allons devoir déterminer les méthodes les plus adaptées à notre objectif d'indexation automatique d'images.

Si on considère tout d'abord les descripteurs de couleur, on peut remarquer en premier lieu que ces descripteurs sont extraits de manière fiable et avec un coût relativement faible comparé à d'autre types de descripteurs. Il s'agit donc de descripteurs particulièrement intéressants. Nous avons observé plus spécifiquement différents compromis entre vitesse de traitement et précision des informations fournies; le critère du temps de traitement entre en ligne de compte car, l'extraction de caractéristiques est une tâche qui doit s'effectuer "en ligne". Un autre élément qui va orienter notre décision est le fait que, comme nous l'avons évoqué précédemment, la perception humaine suggérerait une approche basée sur un découpage en régions de couleur homogène. Nos travaux vont dans ce sens et nous discuterons ce choix plus en détail dans le chapitre suivant. Dès lors on considère que le descripteur des moments de couleur semble le mieux adapté de par son faible coût et le fait que les moments décrivent bien la répartition des couleurs autour d'une couleur moyenne dans une région.

Si on considère ensuite les descripteurs de texture et de forme, on retient plus particulièrement deux éléments. Le premier est le coût élevé de l'extraction d'une information de texture avec les méthodes qui donnent les meilleurs résultats dans les évaluations mentionnées dans notre état de l'art. Le second est que les méthodes statistiques de caractérisation de contours caractérisent aussi, dans une certaine mesure, la texture: la densité de petits contours. On écarte volontairement les méthodes qui impliquent un apprentissage de contours spécifiques (contours actifs) qui ne correspondent pas à la diversité et au polymorphisme des "objets" à identifier dans notre objectif de classification. Notre choix est donc d'opter pour un descripteur qui caractérise à la fois les contours et la texture. Notre contribution est détaillée dans le chapitre 5. Nous envisageons toutefois d'étudier ultérieurement la complémentarité de notre descripteur avec un descripteur performant dédié à la texture (comme le très répandu statistiques d'énergie sur l'image filtrée par une transformée en ondelettes).

Enfin les deux derniers éléments abordés dans notre état de l'art seront utilisés à plusieurs reprises et sont à considérer en fonction de l'utilisation envisagée. Nous reviendrons ainsi ultérieurement sur le choix d'une métrique ou d'une méthode de clustering eu égard de ce que nous recherchons. Notons également que nous n'avons pas inclus d'étude sur les algorithmes de classification, ceux-ci sont certes un élément clé de la classification et en conditionnent grandement les résultats mais nous n'avons matériellement pas pu prendre le temps d'expérimenter les réglages fins de ces différents algorithmes. Ainsi lors de la phase de classification dans le chapitre 6 nous avons expérimenté un ensemble de classificateurs pour obtenir des résultats performants (voir également les expérimentations les concernant en annexe) mais l'étude approfondie d'un classificateur approprié et son optimisation constitue une étape future de notre travail.

Chapitre 4: Décomposition de l'image en régions

Comme nous l'avons vu dans le chapitre 2, trois éléments plaident fortement pour l'utilisation d'une approche basée sur une décomposition en régions dans le but de l'interprétation d'une image. Le premier élément est suggéré par la théorie Gestalt sous la forme de l'application récursive de la loi de continuité. Ceci suggère l'appariement de données au sein de régions homogènes. Un second élément vient d'une affirmation forte de la théorie comme quoi "le tout est différent de la somme des parties" qui prône la prise en compte des relations entre les régions d'une image comme une information à part entière, ceci se retrouve dans nos propres travaux [17]. Enfin le dernier élément fort qui s'est dégagé de notre étude est l'affirmation de Navon [16] au sujet de la précédence du global sur le local, qui suggère également de préférer une vision globale des objets.

C'est donc la perception humaine que nous a conduit à penser que des approches basées sur l'étude de régions pouvaient générer des informations sinon plus pertinentes du moins complémentaires par rapport à des approches locales de type "bag of visual keywords" [5],[6],[7] actuellement plus répandues et donnant les meilleurs résultats sur les tâches de catégorisation d'objets visuels [3].

1. Position du Problème

Il s'agit donc de procéder à une segmentation d'image en régions afin de mieux caractériser son contenu sémantique visuel. Dans la littérature, s'il existe un nombre très important de techniques de segmentation d'images, rares sont celles qui donnent satisfaction au regard de notre tâche de catégorisation génétique d'images qui nécessite qu'un même algorithme soit appliqué sur des images très diversifiées à l'image de la base pascal VOC2007 [3]. Nous nous sommes aperçus que des algorithmes basés sur des seuils (comme l'algorithme CSC que nous avons implémenté) comme les algorithmes adaptatifs (là encore nous avons évalué un clustering basé sur l'algorithme CA) se heurtaient à la diversité de la base. En effet avec un jeu de paramètres qui donnaient de bons résultats dans la majorité des cas testés, certaines images produisaient des segmentations inacceptables, comme par exemple une seule région pour toute l'image, ou au contraire une très forte sursegmentation. Aussi, avons-nous développé un algorithme de segmentation spécifique capable de s'adapter à cette diversité d'images.

Plus précisément, notre objectif étant de pouvoir traiter un problème de classification d'objets visuels, un algorithme de segmentation que nous cherchons doit posséder les caractéristiques suivantes :

- une adaptabilité qui permettent d'avoir des résultats de segmentation exploitables quelle que soit l'image d'entrée et qui plus est sans avoir à redéfinir les paramètres de l'algorithme
- Les régions segmentées devraient être de taille significative pour que les caractéristiques qui en seront extraites aient un sens ;
- Enfin, la performance est un paramètre non négligeable car lors qu'il s'agit d'extraire des descripteurs basés sur les régions segmentées pour la classification d'une image, la tâche de segmentation ne peut pas être "offline"

2. Etat de l'art et notre approche

La segmentation d'images est un problème qui est aussi vieux que l'analyse numérique d'images en général et reste un domaine particulièrement actif encore aujourd'hui. Le but de cette section sera de faire un inventaire des principales approches pour la résolution de ce problème afin de mieux positionner notre propre travail et nos contributions.

2.1. Le Problème de segmentation d'images

Il existe une grande quantité de définitions pour le terme "segmentation" qui diffèrent par quelques subtilités. Dans notre travail nous retiendrons la suivante issue de [94] : la segmentation est un procédé qui divise complètement une image en un ensemble de régions homogènes, reliées entre elles et reliées aux objets constituant l'image. Nous insistons en effet ici sur trois points essentiels :

- La complétude de la segmentation qui fait qu'il n'y aura pas de "trous" dans la représentation de l'image.
- Le fait que les régions soient reliées entre elles qui met en relief l'importance que nous attachons à la relation spatiale
- L'homogénéité qui définit notre critère de segmentation et qui distingue notre procédé d'une décomposition par une grille régulière : il a un sens par rapport aux objets constituant l'image

Si les bénéfices potentiels de l'utilisation de régions ne sont pas questionnables, on peut en revanche évoquer la complexité de la tâche pour discuter de sa pertinence. En particulier, une méthode de segmentation d'image doit surmonter les trois problèmes majeurs suivants: la méconnaissance de l'objectif exact, la pertinence du critère d'homogénéité et l'imperfection de l'image traitée.

La difficulté à définir précisément l'objectif de la segmentation est un premier écueil à la mise au point d'un algorithme. En effet il est précisément difficile de caractériser ce qu'est une "bonne" segmentation. Si l'on regarde le jeu d'évaluations de segmentations proposé par l'université de Berkeley (travaux de Martin et al. [95]), on remarque que plusieurs segmentations humaines très différentes sont proposées pour une même image (voir Figure 20). L'objectif n'est donc pas unique, ce qui fait de la segmentation un problème mal posé au sens de Hadamard [96].

La pertinence du critère d'homogénéité est un second obstacle à la réalisation d'un algorithme efficace. En effet comme nous allons l'étudier par la suite, le procédé de segmentation informatique s'effectue selon un ou plusieurs critères parmi la texture, la couleur et les contours. Or chaque critère présente des faiblesses caractéristiques : deux objets distincts mais de même couleur peuvent se trouver côte à côte, la texture peut varier au sein d'une même région ou au contraire rester similaire dans deux régions distinctes (voir la photographie des zèbres Figure 9) enfin les contours sont à la fois présents au sein d'une même région et absents au sein de frontières perceptuelles (Figure 12). D'une part la combinaison de ces trois critères est coûteuse mais en plus elle ne garantit pas que l'on va combiner les performances plutôt qu'additionner les erreurs inhérentes à chaque méthode.

L'imperfection inhérente de l'image traitée retranscrit également une perte d'informations par rapport à la vision humaine. Le simple phénomène d'acquisition induit la projection d'un monde 3d sur une image 2d. S'y rajoutent le bruit de l'acquisition, l'imprécision dans la retranscription des couleurs et finalement la qualité de la prise de vue (maitrise de l'exposition, de la mise au point, de la profondeur de champ…). On peut objecter

que ces imperfections n'empêchent pas les humains de segmenter des photographies, mais l'observation de la Figure 20 nous montre que les humains segmentent par rapport aux objets sémantiques plus que par rapport à des indices visuels. Ainsi on voit que sur la première image, certains humains ont réduit l'arrière plan à une seule région et la tête de l'animal à une seule région malgré d'importantes différences à l'intérieur de ces régions (quel que soit le critère retenu). Il en est de même pour la musicienne qui est parfois considérée comme une seule région, parfois comme un ensemble régions sémantiques (vêtements, tête, main,…). Il est aussi intéressant de remarquer que quand les objets deviennent moins familiers (par exemple l'instrument de musique – biwa – de la deuxième photo) les segmentations ont plus tendance à se rattacher à des indices visuels qu'à un objet sémantique. Ainsi la majeure partie des humains ont séparé l'endroit ou sont attachées les cordes du biwa qui font pourtant bien partie de l'instrument.

Par ailleurs si l'on prête attention aux images de la Figure 20 on peut y déceler une notion d'échelle d'interprétation. Cette échelle fait fi de toute notion d'homogénéité visuelle (comme en atteste le regroupement par certains observateurs de la tête et du vêtement de la musicienne sur la deuxième photo) : même significativement différentes, les régions de taille inférieure sont absorbées par leurs voisins. S'il est à priori possible d'intégrer ce phénomène à l'algorithme de segmentation, le problème est que le critère de fusion est purement sémantique : pourquoi fusionner la main de la musicienne avec ses vêtements plutôt qu'avec l'instrument dont la couleur est plus proche et qui l'englobe spatialement ? Ce genre de problème est donc particulièrement difficile à résoudre, et il apparait donc illusoire de vouloir retranscrire ce phénomène d'échelle d'interprétation. On distinguera cette notion d'échelle d'interprétation de l'échelle de perception qui, elle, chercherait à retranscrire la vision que l'on pourrait avoir d'un même élément avec un niveau de détails moindre (i.e. : "vu de plus loin"). Si l'on peut générer différents niveaux de vision par différents niveaux de fusion des régions les plus petites, cette opération peut aussi se faire à posteriori ou encore par application de segmentations successives en adoptant une représentation de type "scale space" avec différentes représentation de l'image par filtrages gaussiens. La notion d'échelle sera donc abordée dans le chapitre sur la classification.

Malgré toutes ces difficultés il existe une grande variété d'approches pour proposer des décompositions de l'image. Il faut pour cela commencer par préciser l'objectif de l'algorithme, c'est-à-dire les critères qui permettent de considérer une segmentation comme acceptable. Si les critères précis pour juger de la qualité d'une segmentation sont, à l'évidence, spécifiques à l'application, Haralick et Shapiro [97] (cités dans [94]) ont proposé une série de principes de base que devrait suivre une "bonne" segmentation :

"Les régions issues d'une segmentation devraient être uniformes et homogènes eu égard à une caractéristique comme le niveau de gris ou la texture. L'intérieur des régions devraient être simples et ne pas comprendre de trous de petite taille. Les régions adjacentes issues d'une segmentation devraient avoir des valeurs significativement différentes eu égard à la caractéristique selon laquelle elles sont uniformes. Les frontières de chaque segment devraient être simples, pas en dents de scie."

Il s'agit bien de lignes directrices simples qui présentent un objectif réaliste pour une tâche informatique. C'est en suivant ces principes que nous allons aborder la suite du travail.

Le critère d'homogénéité d'une région peut se reposer sur l'un des trois types de caractéristiques ou leur combinaison : texture, couleur et contours. Chaque approche présente ses atouts et ses inconvénients. On retrouve tout d'abord le problème de la fiabilité d'extraction évoqué dans le chapitre sur les descripteurs. Si un pixel de l'image peut être

immédiatement associé à une couleur, la définition de contours et de textures est un peu plus délicate. Les contours, tout d'abord, sont extraits par rapport à un voisinage immédiat et sur des critères d'extrema locaux dont la pertinence n'est pas garantie. En particulier ils dépendent souvent d'un ou plusieurs seuils qui peuvent générer des contours parasites ou au contraire dissimuler des contours significatifs. La texture, enfin, est une information qui n'a pas de sens sur un pixel et concerne nécessairement une région. L'information est donc particulièrement délicate à localiser spatialement en plus d'être, comme nous l'avons signalé dans le chapitre sur les caractéristiques, délicate à caractériser (avec notamment le problème d'échelle de perception).

Apparait ensuite la possibilité de confusion entre régions à l'évidence distinctes pour un humain. Les contours "imaginaires" (Figure 12) ne sont pas particulièrement fréquents et de ce fait la possibilité de confusion viendra plus vraisemblablement de contours non détectés. Pour la couleur, en revanche, la proximité de deux objets de couleur similaire est tout aussi fréquente que la présence de deux teintes similaires au sein d'un même objet. De plus les dégradés posent problème sur la nécessité de les séparer ou non. On retrouve les mêmes difficultés avec l'information de texture. On peut bien sûr noter que la correspondance région-objet est un objectif irréalisable (les objets ne répondent bien souvent à aucun critère d'homogénéité), mais il serait souhaitable d'éviter autant que possible de mélanger des objets distincts au sein d'une même région eu égard à notre ambition de détecter des objets visuels.

A partir de ces descripteurs un grand nombre de stratégies ont été envisagées pour effectuer la tâche de segmentation. Leurs propriétés étant souvent complémentaires, certaines approches utilisent à la fois les informations de texture et de couleur. Plus généralement on peut ranger les approches utilisant ces deux informations dans une même catégorie au sens ou où ce sont des approches recherchant l'homogénéité sur des caractéristiques extraites pour chaque pixel de l'image. Les approches basées sur le contour travaillent dans l'autre sens en se basant en revanche sur une information de discontinuité (les contours).

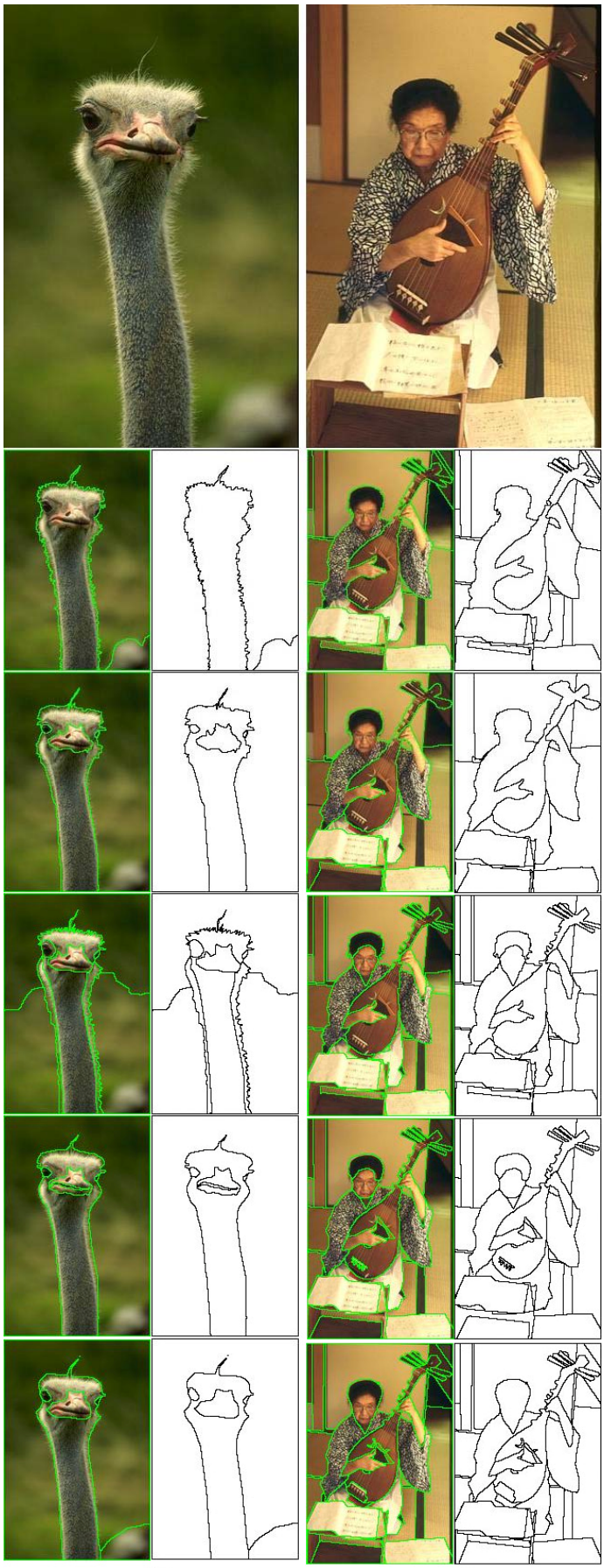


Figure 20: Exemples de segmentations humaines

2.1. Méthodes basées sur les contours

Les méthodes basées sur les contours sont en fait assez minoritaires lorsque le but est de partitionner complètement l'image. Elles présentent l'intérêt de se baser sur une information de taille réduite (liste de contours). La plupart des premières approches évoquées dans [94] se sont basées sur la résolution d'un problème : à partir d'une carte de contours essayer de détecter des contours fermés pour constituer des régions. Cette étape se résout avec différentes techniques selon le souhait ou non d'employer une heuristique de détection ou de se restreindre à des formes spécifiques. On trouvera ainsi des approches par parcours de graphes, suivi des contours, programmation dynamique ou transformée de Hough.

Ces approches sont très dépendantes des paramètres de l'algorithme de détection de contours et elles peuvent se révéler, en dépit de leur initiale simplicité, particulièrement complexes dans le cas d'images riches en contours. Elles sont donc peu répandues actuellement. On peut toutefois noter l'existence des segmentations basées sur les contours dans les travaux de Benois et Barba [98] puis ultérieurement de Jie et Peng Fei [99] avec une utilisation collaborative de la détection de contours avec un algorithme basé régions.

On mentionnera enfin les méthodes basées sur les contours actifs (méthodes des éléments finis [100], contours actifs [101], …) qui, utilisées comme telles, ne sont pas des segmentations au sens où nous l'avons défini (elles ne permettent pas un partitionnement complet de l'image), mais qui ont, elles aussi, été utilisées pour collaborer avec des approches basées sur les régions [102]. Leur principe est le suivant : à partir d'une initialisation on dispose d'un modèle de contours (qui peut correspondre à une forme recherchée) que l'on va chercher à faire correspondre aux contours détectés sur l'image avec un minimum de déformations. On en trouve surtout l'intérêt dans les applications de suivi d'objets en vidéo et, à la notable exception de [102], sont rarement utilisées pour de la segmentation au sens où nous l'avons défini en raison de leur coût et des difficultés d'initialisation.

2.2. Méthodes basées sur les régions

Ces méthodes vont partir de l'intégralité des informations de l'image pour dégager des régions selon un critère d'homogénéité donné. Nous les grouperons en trois catégories définies par [19] : les méthodes spatiales, basées sur la construction de régions, les méthodes de classifications de pixels basées sur le regroupement des pixels selon le critère d'homogénéité seul (la position des pixels n'est pas prise en compte) et les méthodes hybrides qui combinent ces deux approches.

2.2.1. Méthodes spatiales

a) Croissance et/ou division de régions

Les premières méthodes à être apparues dans cette catégorie sont les méthodes basées sur la croissance de régions ; le principe de base est le suivant : à partir d'un pixel source, on le fusionne avec ses voisins pour peu que ceux-ci soient suffisamment proches selon le critère d'homogénéité choisi. Une première approche (citée dans [94]) est de considérer initialement chaque pixel comme une région puis, à chaque itération, de comparer les régions voisines entre elles et de les fusionner si elles sont similaires. On peut citer aussi les travaux de Chassery et Garbay [103] qui diffèrent par l'initialisation : un élément "germe" est pris dans la partie non-segmentée de l'image puis on le regroupe avec des éléments voisins s'ils sont similaires ; on continue jusqu'à ce qu'on n'ait plus d'éléments à regrouper puis on recommence avec un autre "germe".

Avec ces méthodes, on se retrouve dans des problématiques d'agrégation : si les nouveaux pixels potentiels d'une région sont comparés avec la couleur moyenne de la région on peut rencontrer un problème de chaînage (un cas typique est celui des dégradés pour la couleur) où chaque nouveau pixel fera légèrement évoluer la couleur moyenne de la région et permettant à celle-ci d'englober des pixels très différents de ceux qui la constituaient à l'origine. Ce problème (qui rappelle les problématiques rencontrées en clustering sur les algorithmes "single link") peut se résoudre en alourdissant un peu le procédé de fusion de régions et en imposant une distance maximale entre le pixel candidat et le pixel de la région le plus éloigné selon le critère d'homogénéité choisi. Reste que quoi qu'il arrive le procédé de segmentation est dépendant du choix du "germe" (ou de la région initiale) et l'ordre dans lequel sont agglomérées les régions.

A l'opposé de la démarche par agglomération on trouve une approche de division récursive de l'image proposée par Horowitz [104]. Une méthode de division de type quadtree est appliquée jusqu'à ce que la subdivision soit homogène on cherche alors à fusionner les blocs homogènes adjacents. On a ainsi une approche combinée de type division-fusion. Se basant sur un quadtree, cet algorithme étudie des pixels sur la base de régions de forme carrée et aura donc une propension à produire des régions carrées ; de plus la fusion de régions s'opérant dans un certain ordre, le résultat est directement influencé par l'ordre de fusion.

Nous allons étudier un peu plus en détail un algorithme performant de cette catégorie. Il s'agit également d'un travail combinant les approches de division et de fusion : l'algorithme Color Structure Code (CSC) [105], qui opère en effectuant d'abord une fusion puis une division. Nous l'avons utilisé pour tester dans un premier temps dans notre descripteur basé sur des segments (voir le chapitre suivant).

On introduit une structure (appelée île) composée de 6 éléments qui peuvent être soit un pixel (niveau 0), soit une île (niveau i>0). Cette structure est illustrée Figure 21. Au niveau 0 les pixels sont comparés entre eux au sein de chaque île : on fabrique ainsi des régions internes à chaque île étant donné un seuil de similarité.

A chaque itération suivante on continue à étendre les régions au sein d'une même île de niveau supérieur. L'intérêt de cette structure est dans le recouvrement qu'elle induit (représenté sur la Figure 21) : pour savoir si deux régions contenues dans des îles de niveau inférieur fusionnent lors du passage au niveau supérieur, on compare si elles partagent des éléments de niveau inferieur dans la région commune.

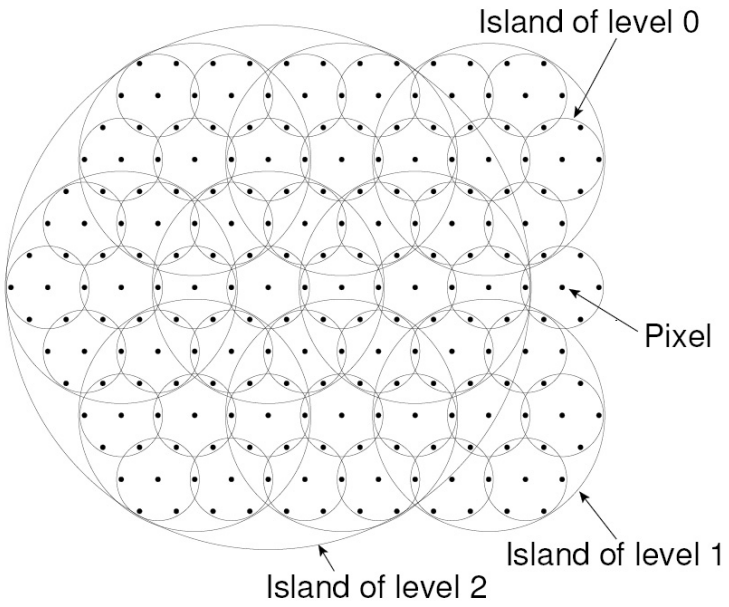


Figure 21: Structure hexagonale de l'algorithme CSC

Pour éviter le problème de chaînage mentionné pour les algorithmes de type "croissance de régions", l'algorithme CSC poursuit cette première étape par une phase de division appliquée lorsque deux régions doivent fusionner : deux régions qui devraient fusionner voient leurs éléments constituants comparés entre eux. Les régions sont séparées si deux éléments non similaires apparaissent, auquel cas les pixels de la région commune sont réaffectés. La Figure 22 donne un exemple de division : I_1 et I_2 doivent fusionner car ils ont des pixels communs dans S. On compare donc les pixels de r1 et r2. Si r1 et r2 sont différents au-delà du seuil de similarité alors les pixels qui étaient communs à r1 et r2 sont répartis entre r1 et r2.

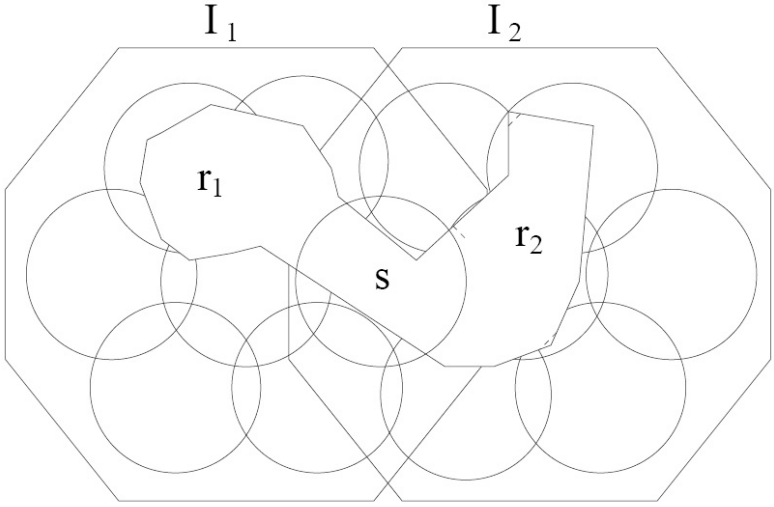


Figure 22: Exemple de fusion/division dans l'algorithme CSC

Même si les diverses procédures ici décrites les atténuent, l'algorithme ne s'affranchit pas totalement de certains défauts évoqués pour les algorithmes de type division-fusion, à savoir la dépendance des résultats à l'ordre de traitement des îles et la tendance à produire des régions hexagonales sous certaines conditions.

Enfin on peut noter que [19] mentionne des approches de types division-fusion basées sur les champs de Markov. Le principe est celui d'algorithmes de type division-fusion, la spécificité de cette approche est que les régions sont considérées comme des champs gaussiens de Markov et que l'appartenance d'un pixel à une région est déterminée par une formule probabiliste conditionnée par le voisinage du pixel. Les opérations de fusion sont aussi conditionnées par des calculs de probabilités. Ces approches se révèlent particulièrement coûteuses en termes de temps de calcul, aussi sont elles assez peu fréquentes dans la littérature et nous n'entrerons pas dans le détail de leur réalisation. La modélisation d'un bruit gaussien, la prise en compte du voisinage d'un point pour calculer son appartenance à une région constituent toutefois des aspects intéressants, en particulier sur les images bruitées.

En résumé, on a abordé une variété d'approches qui prennent en compte à la fois un critère d'homogénéité et l'information spatiale. Ces méthodes présentent par contre de manière commune la nécessité de définir un seuil de fusion de couleurs, ce qui constitue un paramètre critique, dépendant de la nature de l'image, qui conditionne le résultat de la segmentation.

b) Algorithme de "partage des eaux" (watershed)

La technique de segmentation par "partage des eaux" (watershed) a été proposée à l'origine par Digabel et Lantuejoul [106], [107] (cités dans [108]) se définit à partir de la représentation de l'image comme une surface topographique construite par rapport aux intensités du critère d'homogénéité. Le principe en est simple : les minima de cette fonction sont considérés comme des bassins et les maxima comme des seuils de partage des eaux. On décide alors de procéder comme si on faisait monter le niveau de l'eau dans la surface ainsi établie (un exemple sur un problème de dimension 1 est illustré Figure 23), une région étant considérée, à la base, comme un bassin.

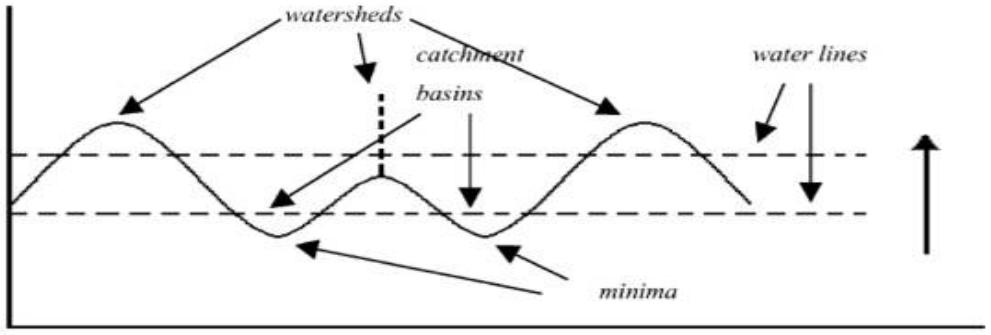


Figure 23: Principe de l'algorithme Watershed

On peut toutefois facilement imaginer que cette définition de base conduit à une sursegmentation massive de l'image étant donnée la présence probable d'un grand nombre de minima locaux, sans parler de l'influence éventuelle de bruit. Plusieurs variantes existent dès lors comme variante hiérarchique où l'on fait successivement monter le niveau de l'eau jusqu'à des seuils de plus en plus hauts, créant ainsi une succession de décompositions de plus en plus grossières.

Cet algorithme qui présente l'avantage d'être déterministe présente quelques inconvénients. D'abord il se base sur une carte d'intensités qui doit être définie de manière pertinente. Si le gradient (qui, comme nous l'avons vu, peut se calculer en intégrant les informations de couleur) est souvent retenu comme information de base, ceci ne permet pas

de gérer le cas des dégradés qui ne seront jamais reconnus comme des seuils. Un autre inconvénient est celui du choix niveau de l'eau à adopter pour l'image finale qui s'apparente au choix d'un seuil.

c) Méthodes variationelles

Nous terminons cette revue des méthodes de segmentation spatiales par l'étude des approches variationelles introduites avec le procédé de diffusion anisotrope de Perona et Malik [50]. Nous avons déjà mentionné ce type de procédés au sujet de l'extraction de contours en décrivant l'approche de [50] comme inspirée de la propagation de la chaleur dans un matériau et qui simule ce comportement en suivant une équation de propagation (9), la conductivité du matériau étant faible (idéalement nulle) sur les contours et forte dans les zones homogènes. On peut alors remarquer que ce type d'approche peut également nous fournir une cartographie de ces régions homogènes. Plus généralement on désignera comme approche variationelle une approche qui se base sur la propagation d'informations dans l'image de manière itérative. Nous illustrerons cette catégorie par la méthode de segmentation "Edgeflow" de Ma et Manjunath [51] qui est reconnue comme particulièrement performante.

Le principe de base est le suivant : on définit un flot à chaque pixel calculé à partir de caractéristiques locales (gradient, texture) auquel on va associer une orientation que l'on veut calculer comme étant dirigée vers le contour le plus proche de la région contenant ce pixel. Pour cela on considère que la frontière entre deux régions sera une zone de fort changement selon une ou plusieurs caractéristiques (critère d'homogénéité). Ainsi la probabilité de rencontrer une frontière dans une direction est calculée à partir de la comparaison entre un pixel distant dans cette direction et le pixel actuel.

Le flot se propage d'un pixel à un autre jusqu'à rencontrer un flot de direction opposée. Cette propagation provoque un lissage des orientations locales et permet ainsi de générer des orientations stables au sein d'une même région. Les points de rencontre de flots de direction opposée sont déterminés comme étant les frontières de la région. La Figure 24 (extraite de [51]) illustre la génération d'orientations et le phénomène de propagation.

Cette méthode présente l'intérêt de pouvoir intégrer de multiples caractéristiques, éventuellement au-delà des caractéristiques de texture et de couleur utilisées par les auteurs. Les exemples donnés par les auteurs montrent également que, dans une certaine mesure, l'effet de lissage produit par la propagation permet de détecter des frontières imaginaires. Il faut toutefois noter que la propagation itérative est relativement lourde en termes de calculs ; de plus le réglage des paramètres est dépendant de l'image à segmenter, enfin l'utilisation de caractéristiques de texture (dont la détermination de l'homogénéité est problématique) tend à conduire à de la sursegmentation sur des images complexes.

2.2.2. Méthodes "classification"

Ces méthodes ne considèrent pas l'information spatiale et considèrent uniquement les coordonnées des points de l'image dans l'espace de caractéristiques retenu comme critère d'homogénéité. Le principe de ces algorithmes est donc de décomposer l'ensemble de points ainsi obtenus en nuages de points distincts. On distinguera essentiellement de méthodes dans cette catégorie : les méthodes basées sur l'analyse d'histogrammes (pour la couleur) et les méthodes basées sur le clustering.

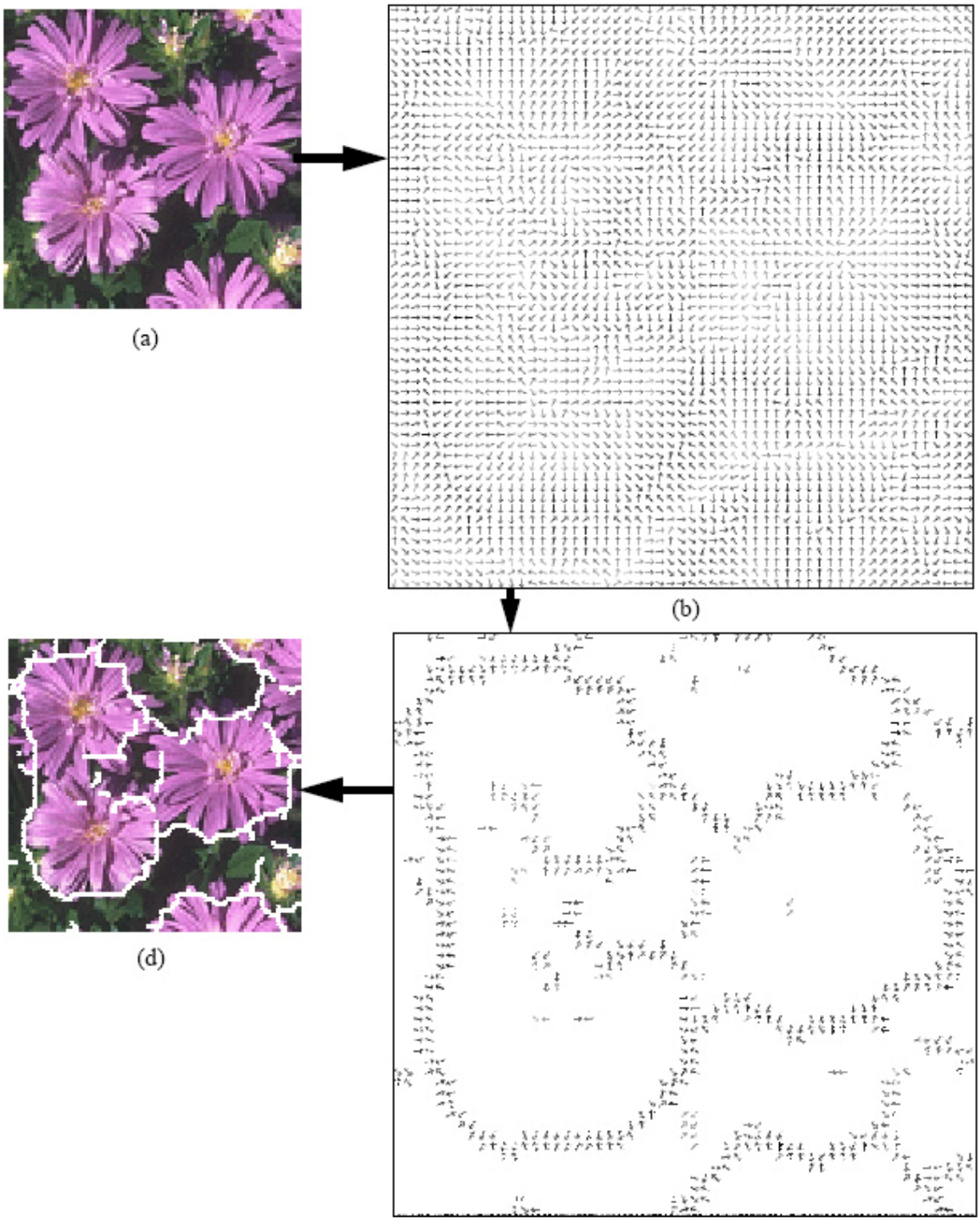


Figure 24: Génération et propagation du flot par l'algorithme EdgeFlow

a) Analyse d'histogrammes

L'analyse d'histogrammes se fait par rapport à l'information de couleur qui s'exprime tout naturellement au moyen d'histogrammes. Elle peut être effectuée ou à partir d'un histogramme 3D ou en décorrélant l'information de couleur pour analyser séparément trois histogrammes. On part quoi qu'il en soit du principe qu'une région de couleur homogène va créer des modes dans les histogrammes. On extrait donc d'abord les modes de chaque histogramme et on cherche à assigner les pixels de l'image chacun à un mode. Comme nous

l'avons déjà mentionné dans la section sur les descripteurs de couleurs, représenter l'espace de couleurs dans un histogramme 3D produit une information particulièrement volumineuse aussi la très grande majorité des travaux décomposent l'image en signaux 1D pour faciliter le traitement. Il est toutefois évident qu'un ensemble d'histogrammes 1D présente un gros problème en ce qui concerne la constitution des régions puisqu'une catégorie distinguée sur une dimension peut en fait correspondre à plusieurs catégories dans l'espace de couleurs l'analyse se fait donc de manière récursive, en réexaminant le résultat de l'extraction de régions constituées à partir des modes de l'histogramme.

Le traitement récursif est apparu avec [109] (cité dans [19]). Par la suite des travaux ont été poursuivis pour étudier les décompositions qui produisent les meilleures partitions. Ainsi Ohta et al. [110] ont insisté sur l'intérêt de travailler sur des composantes décorrélées pour obtenir de meilleurs résultats. Plus généralement, comme noté dans [19] les différentes approches se différencient par les composantes analysées, mais également par la méthode d'extraction des modes (souvent assimilés à des gaussiennes), la détermination de l'importance des composantes (ces algorithmes sont sensibles à l'ordre dans lequel sont utilisées les différentes composantes) et les critères d'arrêt de l'analyse d'histogrammes (celui-ci étant un procédé récursif).

b) Clustering

Une autre approche possible au problème de segmentation est d'effectuer un clustering dans l'espace de caractéristiques. Ceci pose un problème de volume de données : par exemple, prise telle quelle, l'information de couleur d'une image conduirait à un clustering parmi plusieurs dizaines de milliers de couleurs. Les informations de texture sont encore plus volumineuses de par la taille des vecteurs de caractéristiques produits par les caractérisations classiques (voir la section sur ce sujet). Dans ces conditions, trois classes de problèmes se posent : d'abord la complexité de l'algorithme utilisé pour le clustering qui devient un critère déterminant, ensuite les difficultés propres à l'algorithme choisi (détermination du nombre de clusters, initialisation,…), enfin les difficultés propres à la mesure de distance choisie (la plupart des algorithmes construisent des clusters hypersphériques par rapport à la distance choisie).

On peut par exemple citer l'approche de Zhang et Wang [111] qui utilise un clustering classique de type K-Means mais appliqué avec une mesure de distance adaptée pour séparer les couleurs de manière plus pertinente. Fauqueur et Boujemaa [112] proposent une segmentation utilisant l'algorithme CA (décrit dans la section "clustering") qui permet une détermination automatique du nombre de clusters appropriés. L'initialisation est effectuée par un échantillonnage de couleurs de l'image d'origine selon une grille régulière. Une distance de Mahalanobis est utilisée pour permettre des clusters de forme ellipsoïde. D'autres algorithmes déterminent le nombre de couleurs quantifiées par l'étude préalable des histogrammes [113]. Enfin on notera les travaux de Liew et al. [114] qui intègrent une notion de voisinage en modifiant les formules de calcul d'appartenance du fuzzy C-means.

c) Discussion

Les méthodes basées sur la segmentation bénéficient d'une vue globale sur les données à regrouper. Il faut par contre noter que les méthodes basées sur la décomposition d'histogrammes sont obligées de faire des suppositions sur la répartition des couleurs de l'image dans l'histogramme (souvent supposé gaussienne). Ils sont également dépendants de la qualité des projections choisies pour constituer leurs histogrammes 1D ainsi que de leur capacité à distinguer les modes (particulièrement les moins marqués) du bruit. Pour les

méthodes basées sur le clustering, on remarquera qu'il est parfois difficile de s'affranchir des limitations des algorithmes utilisés comme la difficulté à établir un nombre pertinent de clusters cible, la dépendance aux conditions initiales.

Après l'application d'une méthode de type "classification", il est également nécessaire de s'occuper des régions qui, par construction, peuvent être spatialement disjointes. Les algorithmes de cette catégorie sont donc souvent suivis de l'application d'un algorithme spatial simple.

2.2.3. Methodes hybrides

Nous terminerons notre étude en mentionnant quelques exemples de travaux qui font collaborer plusieurs des méthodes décrites plus haut.

Un premier exemple est donné dans [115]. L'algorithme débute par un clustering fuzzy C-Means qui va produire un grand nombre de clusters hypersphériques. Par la suite un algorithme de type "croissance de région" est appliqué pour regrouper ces clusters. L'algorithme débute donc avec une phase globale avant d'affiner les régions localement en utilisant l'information spatiale. Ces travaux illustrent une tendance globale qui est de compléter les méthodes "classification" par des méthodes spatiales.

On mentionnera enfin une coopération entre les algorithmes basés sur les contours actifs et les algorithmes de croissance de région avec les travaux de Zhu et Yuille [102]. Cette méthode part de germes de régions qui vont produire des régions selon le critère de longueur de description minimale (MDL) [116] basé sur l'homogénéité des régions et la régularité de leurs contours. On procède alors à un déplacement des frontières des régions selon des principes issus des algorithmes de contours actifs.

2.1. Notre approche

On s'attachera à tenir notre ligne directrice tracée par les principes de la perception humaine. Ceux-ci devront donc se refléter autant que possible dans les différents aspects de nos algorithmes.

Dans nos travaux, nous nous sommes orientés vers un algorithme basé sur les régions en utilisant un critère d'homogénéité perceptuelle. Comme nous l'avons évoqué, ce critère d'homogénéité perceptuel pourrait être basé sur la couleur et/ou la texture et c'est la couleur qui a été retenue dans notre algorithme de segmentation car une caractérisation fiable de latexture est difficile tout comme la précision de la condition d'homogénéité correspondante (voir la section sur les descripteurs de texture).

La construction de notre algorithme de segmentation s'articule autour de deux phases : une phase quantification de couleurs et une phase de segmentation. La quantification de couleurs d'une image vise à réduire la palette de couleurs d'une image à un nombre perceptuellement significatif [117] tout en permettant d'améliorer la performance de l'algorithme de segmentation, tandis que la phase segmentation proprement dite utilisera cette quantification et déterminera automatiquement un nombre idéal de couleurs quantifiées par une mesure de distorsion MSE eu égard de la tâche de segmentation [118]. Notre première phase de quantification est nécessaire car elle nous permet d'utiliser des algorithmes plus complexes (par rapport au nombre de couleurs) durant la phase de segmentation. L'algorithme de quantification ne déterminant pas automatiquement un nombre de couleurs cible, nous réduirons donc durant cette phase le nombre de couleurs à une quantité fixe qui correspond à

une limite supérieure du nombre de régions déduite des études sur la segmentation humaine de [95].

Nous développerons dans les deux sections suivantes d'une part notre méthode de quantification de couleurs, Self-information based color quantization (SICR), qui est basée sur une combinaison de la représentativité d'une couleur et son apport de l'information, et d'autre part notre méthode de segmentation globale qui déterminera le nombre optimal de couleurs quantifiées. Le procédé de segmentation ainsi développé sera utilisé tel quel dans les chapitres suivants pour l'extraction de caractéristiques. On notera que les versions présentées ici sont des versions améliorées de celles qui ont été publiées en [117] et [118]

3. Quantification de couleurs basée sur la "self information" (SICR)

La quantification de couleurs dans une image est un prétraitement à une image de couleur qui vise à réduire la palette de couleurs dans une image afin d'améliorer les performances de traitements ultérieurs. Il existe une grande variété de méthodes de quantification de couleurs (une partie de ces méthodes peut se retrouver dans les études [119] et [120]) ; actuellement les principales méthodes de quantification de couleurs reposent essentiellement sur le "halftoning" (approximation d'une teinte par la constitution d'un motif composé de deux teintes différentes dont la proximité donne l'illusion de la couleur appropriée), les divisions successives et le clustering. Un problème de ces différents types d'algorithmes est leur objectif de représenter le mieux possible la distribution initiale de couleurs au sens où l'on veut minimiser l'erreur de quantification. Or dans notre problématique d'analyse d'images basée sur la perception humaine, il est important de préserver la dynamique de couleur de l'image. En effet la perception humaine accorde une grande importance aux régions qui sont différentes de leur environnement : ceci se traduirait par la conservation de couleurs qui ne seraient pas présentes en très grand nombre mais qui se démarquent significativement des autres.

Nous passerons rapidement sur les techniques de "halftoning", à l'image de celle proposée dans [121], qui représentent un moyen efficace de représenter une grande quantité de couleurs à partir d'une palette très réduite mais sont tout simplement inadaptées à notre objectif. En effet le halftoning vise une similarité visuelle alors que nous recherchons plutôt à conserver les couleurs de l'image originale.

Les techniques basées sur les divisions successives partent d'un ensemble contenant les couleurs de l'image et le divisent en deux sous ensembles. A chaque itération, on choisit de diviser un ensemble en deux jusqu'à obtention du nombre désiré d'ensembles. L'algorithme Median Cut par Heckbert [122] constitue un des travaux initiaux les plus connus de cette catégorie. Nous utiliserons l'algorithme "Principal Analysis", proposé par Wu [123] et reconnu comme particulièrement performant pour illustrer les résultats de ce type de méthodes dans la section "résultats".

Enfin les techniques basées sur les algorithmes de clustering sont utilisées au même titre que pour la segmentation d'images (cf. la section état de l'art de ce chapitre). Nous avons souligné à cette occasion le volume important de calculs à effectuer et la sensibilité à l'initialisation de cette classe d'algorithmes. Des méthodes récentes utilisant des algorithmes de classification non supervisés (Self Organizing Maps) pour définir les couleurs quantifiées en intégrant des informations spatiales ont vu le jour ([124], [125]) et produisent actuellement les meilleurs résultats selon un critère d'erreur quadratique moyenne entre les couleurs quantifiées et les couleurs originales. Nous utiliserons l'algorithme décrit dans [124] pour illustrer les résultats de ce type de méthodes dans la section "résultats".

C'est cette notion "d'importance de couleur" que nous avons traduite dans notre algorithme en utilisant l'apport d'information (self-information). Nous l'avons ainsi appelé SICR pour "Self Information Color Reduction".

3.1. Réduction perceptuelle du nombre de couleurs

La première étape de notre algorithme est de réduire la complexité de la tâche de choix des couleurs quantifiées par un regroupement rapide des couleurs similaires. Le principe de notre réduction sera donc de garantir que les couleurs regroupées soient perceptuellement similaires et de quantifier les erreurs que l'on peut commettre.

Afin d'agglomérer les couleurs perceptuellement similaires, on calcule un histogramme de couleurs en 3 dimensions et on regroupe les couleurs qui se retrouvent dans la même cellule de l'histogramme en les remplaçant par le centre de cette cellule. Ce simple énoncé fait émerger deux paramètres critiques que sont l'espace de couleurs choisi pour la constitution de l'histogramme et la taille des cellules de l'histogramme 3D.

Concernant l'espace de couleur, on cherche à limiter la différence perceptuelle entre deux éléments au sein d'une même cellule d'une part et d'autre part à maximiser le nombre d'éléments perceptuellement similaires au sein d'une même cellule. Sous ces conditions il apparait que l'espace RGB est clairement insuffisant à cause de sa forte hétérogénéité et particulièrement de ses redondances (un exemple a déjà été exposé Figure 5). Pour les autres espaces de couleur, on part du principe que les images que l'on va devoir traiter seront exprimées, comme toutes les images produites par les appareils photo numériques actuels, en RGB codés avec 8 bits par canal. En conséquence il ne s'agit pas d'étudier directement les espaces de couleurs mais plutôt les couleurs générées par les triplets RGB possibles. La procédure pour déterminer la taille des cellules de l'histogramme se fait par une séquence d'essais: on génère un histogramme avec l'intégralité des couleurs de la palette RGB et on calcule toutes les distances entre les couleurs qui tombent dans une cellule et le centre de celle-ci. L'idée étant tout simplement de borner les distances entre une couleur originale et un élément quantifié. La distance utilisée est sujette à discussion, le problème des discontinuités au voisinage du point de l'axe L des espaces de type TSL rend une distance basée sur cet espace assez délicate à mettre au point (la distance euclidienne n'étant à l'évidence pas appropriée). Ici le temps de calcul n'est pas un problème : il s'agit d'une détermination globale de la précision voulue pour notre algorithme et sera donc effectuée de manière "offline" et une fois pour toutes. De ce fait, l'utilisation d'une distance avancée comme la distance CMC présente l'avantage d'être d'une distance homogène à la perception humaine sans discontinuité majeure avec un seuil perceptuel de 1 pour la différence entre deux couleurs. Nous l'utiliserons donc comme référence. Des tests sur l'espace TSL ont rapidement révélé d'importantes différences dans les cellules du voisinage de l'axe L, nous conduisant à utiliser un espace parmi CIELab et CIELuv. Ceux-ci étant particulièrement proches que ce soit en termes d'évaluations ou en termes de calculs, nous avons choisi l'espace CIELab, la distance CMC étant exprimée par rapport à cet espace, ce qui la rend plus facile à calculer. Le Tableau 3 résume les principales expérimentations faites sur la taille des cellules. Nous sommes partis du principe que la plage de variations de a et b était usuellement 2 fois plus importante que celle de L, la dimension des cellules d'histogramme reflète cet aspect avec des cellules deux fois plus grandes selon a et b que selon L. Il faut aussi noter que nous utilisons le centre de la cellule qui, par rapport à l'utilisation du centre de gravité des couleurs de la cellule, va accroitre la distance moyenne mais limiter la distance maximale.

Taille des cellules L, a, b	0.33, 0.66, 0.66	0.5, 1, 1	1, 2, 2	1.5 ,3 ,3	1.75, 3.5, 3.5	2, 4, 4
Distance CMC(2,1) moyenne	0.14	0.22	0.43	0.64	0.76	1.25
Distance CMC(2,1) maximale	< 1	1.35	2.69	4,04	4.62	7.22
Distances CMC(2,1) inférieures à 1	100%	99.99%	97.82%	88.54%	80,53%	38.56%

Tableau 3: Tests sur la quantification des couleurs

Ces tests d'efficacité sont à mettre en balance avec la quantité de couleurs que nous pouvons agglomérer. Il apparait déraisonnable que deux couleurs puissent être agglomérées avec une distance CMC(2,1) moyenne supérieure à 1. Nous avons finalement opté pour des cellules de taille 1.5 x 3 x 3. Qui, conserve 98% des distances en dessous de 1.5 et qui, après un test sur une base de 600 images hétérogènes, réduit la quantité de couleurs au sein de l'image de 89% en moyenne.

3.2. Procédé de Quantification

L'étape précédente nous permet d'obtenir une réduction significative du nombre de couleurs pour une étape qui peut se dérouler lors de l'extraction de couleurs de l'image et donc pour un coût quasi nul. En partant de cet espace de couleurs réduit, on peut effectuer la quantification. Nous avons expliqué dans l'introduction que nous souhaitions prendre en compte l'importance de la singularité d'une couleur. Pour cela nous allons utiliser la théorie de l'information et exprimer la quantité d'information produite par une couleur à choisir par rapport aux couleurs déjà choisies. D'autre part nous souhaitons garder la possibilité d'éviter la création de couleurs : choisir la moyenne d'un ensemble de couleurs va en effet, dans le cas où l'ensemble est dispersé, aboutir à la création d'une couleur qui n'a rien à voir avec l'image d'origine. Nous allons donc procéder par sélection successive de couleurs quantifiées à partir de l'image. Après quoi nous envisagerons le cas où nous souhaiterions en priorité favoriser la proximité moyenne des couleurs quantifiées aux originales au détriment de la conservation des couleurs originales de l'image.

Pour la sélection des couleurs nous allons mettre en compétition deux facteurs : la population représentée par une couleur d'une part et la quantité d'information qu'elle apporte d'autre part. La quantité d'information I d'une couleur c s'exprime selon (51)

$$I(c) = -\log(P_2(c)) \qquad (51)$$

Où $P_2(c)$ est la probabilité (à un coefficient de normalisation près, celui-ci se traduit par une constante qui n'a pas d'impact sur l'ordre des résultats) d'avoir une couleur similaire à c dans l'ensemble choisi. Cette probabilité peut être estimée selon (52) en supposant que l'ensemble des couleurs similaires à une couleur déjà choisie c_j suit une distribution gaussienne :

$$P_2(c) = \frac{\sum_{j=0}^{S} e^{-\frac{d^2(c,c_j)}{2\sigma^2}}}{S} \qquad (52)$$

Où $d^2(c,c_j)$ représente le carré d'une distance entre la couleur c et la couleur quantifiée c_j, "S" représente le nombre de couleurs quantifiées jusqu'à présent. σ est la variance qui est donc un paramètre définissant la taille du voisinage au sein duquel deux couleurs sont définies comme similaires.

On définit également $P_1(c)$ comme la probabilité d'observer une couleur c dans l'image (population de la couleur quantifiée divisée par le nombre total de pixels dans l'image). On sélectionne ensuite les couleurs itérativement selon une combinaison de ces deux critères, à savoir la population représentée par une couleur et la quantité d'information qu'elle apporte aux couleurs déjà choisies d'autre part. Pour cela, nous avons évalué plusieurs méthodes de combinaison, comme par exemple la combinaison linéaire simple, mais les meilleurs résultats (tant de manière subjective que par le calcul de l'erreur quadratique moyenne) ont été obtenus par application de la formule de combinaison donnée par (53).

$$Eval_{\alpha\tau}(c_i) = \alpha \cdot e^{\frac{-i}{\tau}} \cdot \frac{1}{1 - \sum_{j=0}^{i} P_1(c_j)} \cdot I(c_i) + P_1(c_i) \qquad (53)$$

Où α et τ sont des paramètres qui représentent respectivement l'importance relative de la quantité d'information par rapport à la population et le taux de décroissance de cette importance.

Cette formule s'interprète simplement par une priorité initialement accordée au choix basé sur la quantité d'information et qui décroit avec le temps. Cette décroissance est pondérée par la population déjà représentée par les couleurs quantifiées : plus la quantification représente la population originale, plus on met l'accent sur la quantité d'information. La couleur choisie est celle qui a la meilleure évaluation. Pour la première itération, on choisit simplement la couleur qui présente la plus forte population.

Etant donné un nombre initial de couleurs à choisir, cet algorithme de quantification se termine donc avec une sélection parmi les couleurs de l'image d'origine. Néanmoins, au cas où l'on souhaiterait aussi améliorer l'erreur quadratique moyenne entre les couleurs quantifiées et les couleurs originales, on peut aussi lancer l'algorithme de K-Means avec comme initialisation les couleurs qui viennent d'être choisies. Mais on aboutirait alors à la création de nouvelles couleurs.

L'algorithme global de quantification est donc le suivant :

1. Pré-quantification par agglomération des couleurs similaires
2. Calcul des probabilités d'observation de chaque couleur dans l'image
3. Sélection de la couleur de plus forte population
4. Calculer la distance des couleurs non-choisies à la première couleur sélectionnée

5. Tant qu'il reste des couleurs quantifiées à choisir
    - Pour chaque couleur candidate : calculer son évaluation (52)
    - Sélectionner la couleur candidate ayant la meilleure évaluation
    - Calculer la distance de chaque couleur candidate au nouveau centre choisi
6. Eventuellement, exécution de quelques itérations de K-Means

3.3. Complexité

La première étape consiste à passer tous les pixels de l'image pour placer les couleurs correspondantes dans l'histogramme. Sa complexité est donc linéaire par rapport au nombre de pixels de l'image. A ceci on rajoute la phase de quantification, pour chaque couleur quantifiée on va calculer les évaluations pour chaque couleur qui comprennent une somme sur toutes les couleurs déjà quantifiées, puis les distances à toutes les couleurs. La complexité est donc fonction linéaire du nombre de couleurs de l'image et en n^2 par rapport au nombre de couleurs quantifiées.

Le calcul des gaussiennes à chaque itération est assez lourd en termes de calcul. Ceci peut être amélioré par une discrétisation des distances entre les couleurs et le précalcul des coefficients de chaque gaussienne.

3.4. Résultats expérimentaux

On présente dans un premier temps les résultats après les différentes étapes à ceci près qu'on ne donnera aucune illustration de la phase de pré-quantification qui altère très peu l'image, celle-ci n'ayant un effet perceptible que sur certains dégradés. La Figure 25 illustre la quantification d'une image sans création de couleur (paramètres déterminés empiriquement : $\alpha = 60$, $\sigma = 3$, $\tau = 3.5$), celle-ci est à comparer à la Figure 26 qui illustre l'application de SICR avec $\alpha = 0$ (aucune influence de la quantité d'information) ainsi qu'à la Figure 27 qui montre au contraire l'application de SICR sans influence de la population de couleurs ($\alpha = \infty$). Les mesures CMC données ici sont les mesures CMC(1,1). La distance utilisée dans (51) est la distance euclidienne dans l'espace CIELab, l'utilisation de la distance CMC ayant été testé et n'apportant qu'une faible amélioration de la MSE pour un coût de calcul accru.

Figure 25: Quantification SICR sans création de couleur (108 couleurs)
MSE CMC : 20
MSE Lab : 12

Figure 26: Quantification basée uniquement sur les populations (108 couleurs)
MSE CMC : 38
MSE Lab : 29

Figure 27: Application de SICR avec prise en compte de la "self information" uniquement (108 couleurs)
MSE CMC : 20
MSE Lab : 12

Voici ensuite trois exemples de quantifications de couleurs sur deux images issues de la base de Berkeley [95] et une troisième collectée sur Internet. L'application de SICR se fait avec un jeu de paramètres déterminés de manière empirique : $\alpha = 60$, $\sigma = 3$, $\tau = 3.5$. On calcule cette fois l'erreur quadratique moyenne (MSE) entre les pixels originaux et les pixels quantifiés. Ces exemples ont été choisis car ils étaient particulièrement représentatifs des différentes situations rencontrées. Nous commentons les résultats en comparaison d'algorithmes parmi les plus performants des deux principales catégories de procédés de quantification : l'algorithme de Wu [123] et l'algorithme Adaptive Color Reduction (ACR) [124].

Image Originale


Image quantifiée (Wu, 108 Couleurs)

MSE (CMC) : 24,5 MSE (LAB) : 16,8
Image quantifiée (ACR, 108 Couleurs) :

MSE (CMC) : 21,7 MSE (LAB) : 16,0

Image quantifiée (SICR, 108 Couleurs)

MSE (CMC) : 13,8 MSE (LAB) : 8,9

Ce premier exemple met en lumière la singularité de notre approche : notre but est effectivement de préserver la dynamique de l'image car notre but est de conserver des tons proches de l'image originale en négligeant les tons qui sont voisins. Ceci se reflète par exemple sur le visage du musicien (et d'une manière générale sur tous les dégradés) qui présente plutôt peu de couleurs quantifiées mais qui restent proches de l'original. Les teintes produites par l'algorithme de Wu sont globalement déssaturées alors qu'ACR fait évoluer l'ensemble des teintes vers une teinte dominante de l'image. Ces imprécisions leur permettent par contre de mieux respecter les dégradés.

On relativisera enfin le critère de MSE qui, s'il reflète la proximité objective des pixels des images quantifiée et originale, n'est pas le reflet des sensibilités de la perception humaine. En l'occurrence le fait que le regard se concentre sur le visage du musicien fait que l'algorithme ACR qui produit pourtant de bons résultats au vu de la MSE parait finalement le moins proche de l'image d'origine en raison de l'altération des couleurs qu'il a provoqué dans cette région.

Image Originale

Image quantifiée (Wu, 109 Couleurs)

MSE (CMC) : 11,8 MSE (LAB) : 10,9

Image quantifiée (ACR, 109 Couleurs)

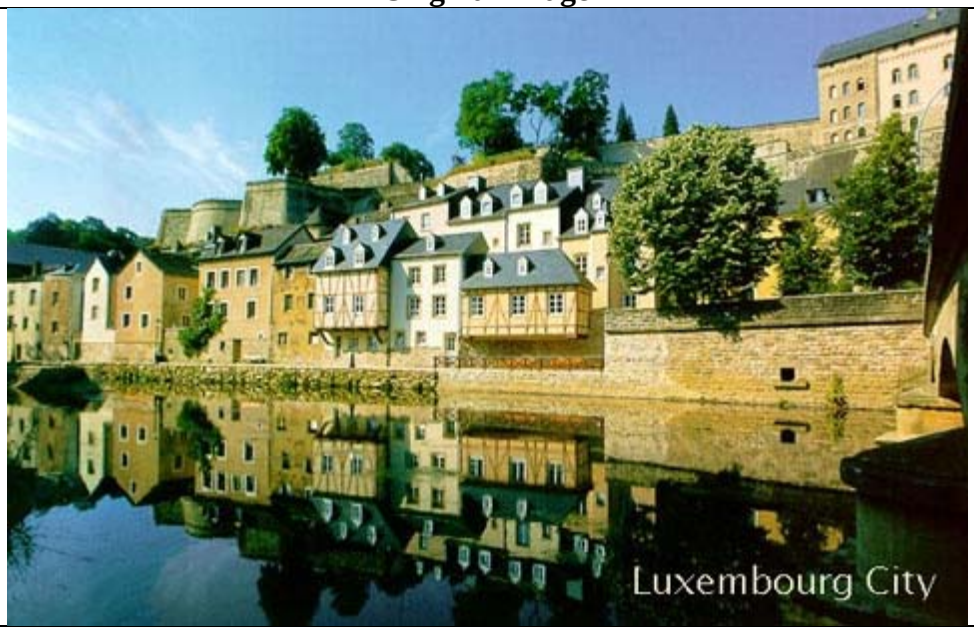
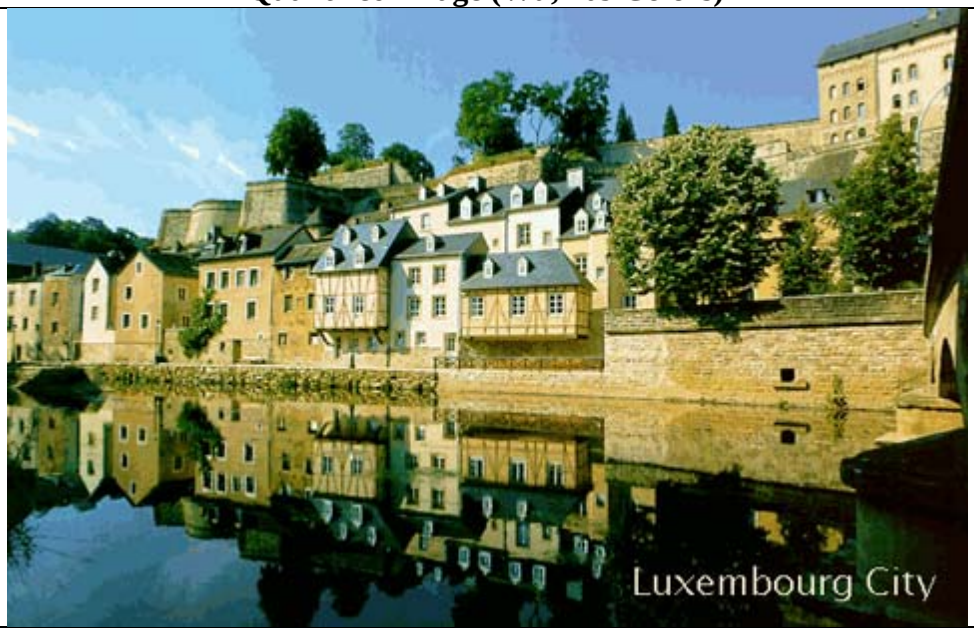
MSE (CMC) : 13,3
MSE (LAB) : 12,1

Image quantifiée (SICR, 109 Couleurs)

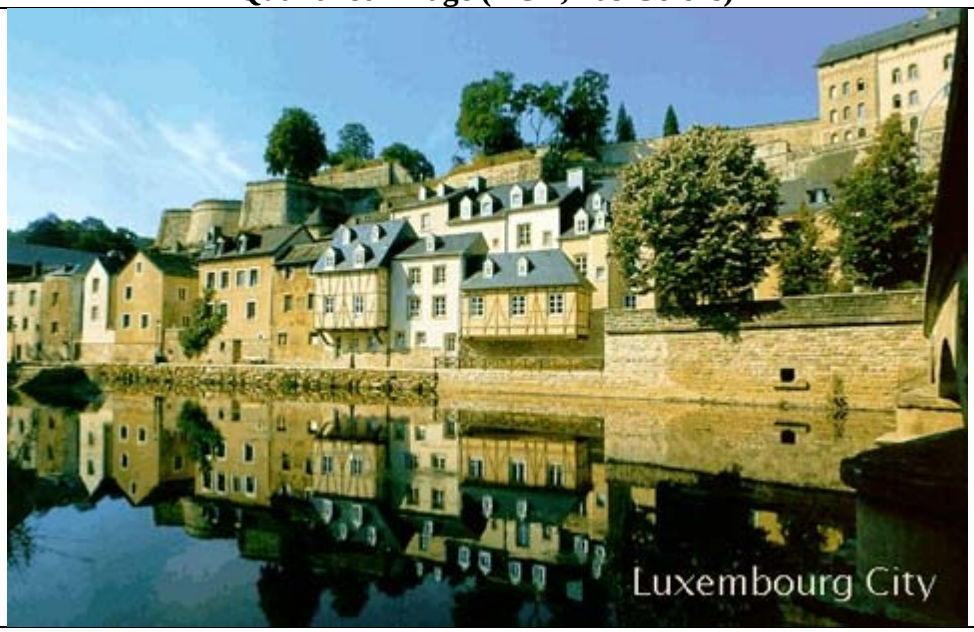
MSE (CMC) : 7,6
MSE (LAB) : 6,9

Cette image est plus délicate que la précédente: la plupart de l'image est constituée de teintes de bleu avec au centre de l'image une personne qui, bien qu'étant l'élément principal de l'image, en représente une faible portion en terme de surface. Dans ces circonstances, les trois algorithmes se conduisent comme nous l'avions observé pour l'image précédente : l'algorithme de Wu éclaircit globalement l'image, notre algorithme choisit de représenter les teintes minoritaires (visage du surfeur, jaune sur la planche, reflet de la combinaison dans la vague) au détriment des teintes de bleu, ce qui a pour effet d'endommager les dégradés, l'algorithme ACM adopte un procédé opposé en décalant les couleurs vers la dominante bleue

ce qui pour cette image est particulièrement néfaste puisque cela a pour effet de rendre le visage du surfeur en bleu et de faire disparaitre son reflet dans la vague.

Original Image

Quantized Image (Wu, 109 Colors)

MSE (CMC) : 28,5 MSE (LAB) : 35,1

Quantized Image (ACR, 108 Colors)



MSE (CMC) : 30,5
MSE (LAB) : 41,5

Quantized Image (SICR, 108 Colors)

MSE (CMC) : 22,9
MSE (LAB) : 28,5

Cette image représente les cas ou l'algorithme SICR est le moins adapté. Le but de notre algorithme est de reproduire la dynamique de couleur de l'image d'origine, son principe étant de choisir des couleurs diversifiées même si elles ne sont pas très représentées dans l'image, ce qui a pour conséquence de diminuer le nombre de couleurs choisies pour représenter les différents tons de couleurs déjà représentées. Ceci est particulièrement adapté aux images précédentes qui présentent des régions singulières. Ici, toutefois, on se retrouve dans le cas où on n'a pas de région avec une couleur distinctive et où les dégradés constituent une très grande partie de l'image. On constate que notre algorithme parvient toutefois à s'adapter avec succès.

Comme pour les images précédentes, l'algorithme de Wu éclaircit l'image et ACR optimise la MSE en adoptant des teintes proches de la couleur dominante marron (particulièrement visible dans les arbres).

Enfin le Tableau 4 présente les résultats de l'évaluation complète sur une base de 600 images comprenant l'ensemble des images de la base de Berkeley ainsi que des images collectées sur internet et comprenant des conditions d'éclairage et des palettes particulièrement variées. Il confirme les bons résultats de notre algorithme.

	Valeurs de la MSE			
	108 Couleurs		135 Couleurs	
	CMC	L_2 Lab	CMC	L_2 Lab
Quantification de Wu	18,58	15,26	16,70	13,64
ACR	17,00	14,73	14,80	12,80
SICR	12,06	9,72	10,64	8,63

Tableau 4: MSE moyennes sur une base de 600 images

3.5. Conclusion

Nous avons proposé ici une méthode de quantification différente des algorithmes existants en mettant l'accent sur la diversification des couleurs quantifiées choisies dans le but de préserver la dynamique de l'image. Les expérimentations ont montré qu'il donnait de meilleurs résultats que les autres algorithmes actuels selon un critère d'erreur quadratique moyenne. Cet algorithme présente aussi la particularité de pouvoir éventuellement éviter la création de couleurs si nécessaire. Enfin on remarquera que l'algorithme est déterministe et produira toujours les mêmes résultats si on l'applique à la même image avec les mêmes paramètres.

4. Segmentation d'images

Notre deuxième contribution est l'algorithme de segmentation complet qui utilise SICR en le complétant par un prétraitement, une détermination automatique du nombre de couleurs et un traitement spatial des régions.

Comme nos expériences préliminaires l'ont montré (voir section 6) et comme nous l'avons dit plus haut en exposant les motivations de nos travaux, l'importance du nombre de seuils à déterminer pour les algorithmes spatiaux et la difficulté de les déterminer automatiquement rentre en conflit avec notre recherche d'un algorithme robuste applicable à une base d'images diversifiée sans avoir à ajuster les paramètres ; nous avons donc décidé d'utiliser une approche mixte qui se base dans un premier temps sur une approche de type "classification".

4.1. Prétraitement

Nous avons déjà évoqué que nous devions mettre l'accent sur la robustesse de notre algorithme. Le cas d'images bruitées et/ou la présence de petites régions a pour conséquence de parasiter la segmentation. En particulier, comme nous l'avons évoqué lors de notre étude

du chapitre 2, notre objectif est de produire des régions de taille significative provenant, conformément à la théorie Gestalt, d'un assemblage de régions moins importantes en termes de proportions. Cet aspect devient prépondérant si on considère aussi que l'extraction de caractéristiques visuelles aura peu de sens si les régions sont trop petites. La première étape que nous effectuons est donc un filtrage. Nous souhaitons toutefois éviter autant que possible d'endommager les frontières des régions ce qui serait le cas avec un filtrage gaussien. Nous choisissons donc la solution d'appliquer un filtre médian vectoriel [126]. Celui-ci nous permet d'homogénéiser les régions sans endommager les contours, il se distingue du filtre médian classique (filtre médian marginal) par la fusion des canaux de couleur. Nous avons préféré cette alternative à l'application d'un filtrage anisotrope [50] certes performant mais beaucoup plus coûteux.

Notre étape de prétraitement est complétée par l'application de notre algorithme de réduction perceptuelle du nombre de couleurs (procédé dynamique également utilisé dans SICR avant l'opération de sélection des couleurs). En effet nous ne l'utiliserons pas seulement pour l'application de SICR mais également pour la détermination du nombre de clusters.

4.2. Détermination automatique du nombre de clusters

Comme nous l'avons vu dans notre bref état de l'art, la détermination automatique du nombre de clusters est le point délicat d'un grand nombre d'algorithmes de clustering. Les essais que nous avons menés avec des algorithmes comportant une détermination automatique du nombre de clusters (implémentation de l'algorithme ARC [89]) n'ont pas donné satisfaction pour l'intégralité des images étudiées. Nous avons donc choisi de développer une approche spécifique pour la détermination du nombre de couleurs quantifié.

4.2.1. Etude de la MSE en fonction du nombre de clusters

Nous avons commencé par observer l'évolution de l'erreur quadratique moyenne entre les centres des clusters et les couleurs originales. Le protocole de l'étude est le suivant : pour un ensemble de 50 images issues de la base Pascal 2007 [3] et aussi diversifiées que possible on effectue une série de calculs de MSE pour différents nombres de clusters cible. Chaque cluster représentant au minimum une région, le nombre maximal de clusters cible est déterminé à partir d'une limite supérieure du nombre de régions voulu dans une image. On peut obtenir une indication de cette limite supérieure du nombre de régions en observant les segmentations humaines proposées pour les images de la base de segmentation de Berkeley [95]: une limite de 200 régions nous a paru comme une borne supérieure qui ne serait raisonnablement pas dépassée. Pour chacune de ces 50 images, nous avons donc mesuré la MSE en effectuant des quantifications correspondant à un nombre de régions variant de 5 à 200 régions avec un incrément de 5 régions. Pour cette quantification, nous avons utilisé un algorithme de clustering classique, en occurrence celui de « neural gas » pour sa faible dépendance aux conditions initiales, un algorithme introduit dans la section « clustering » du chapitre d'état de l'art.

La Figure 28 présente les graphes obtenus pour 5 images représentatives (Figure 29) : globalement les graphes obtenus suivent tous le même modèle avec la MSE qui augmente de façon quasi-linéaire jusqu'à un certain seuil où la croissance de la MSE devient exponentielle. La correspondance avec les couleurs est comme il suit :

Image a) : courbe bleu foncé
Image b) : courbe jaune
Image c) : courbe bleu clair
Image d) : courbe rose
Image e) : courbe mauve

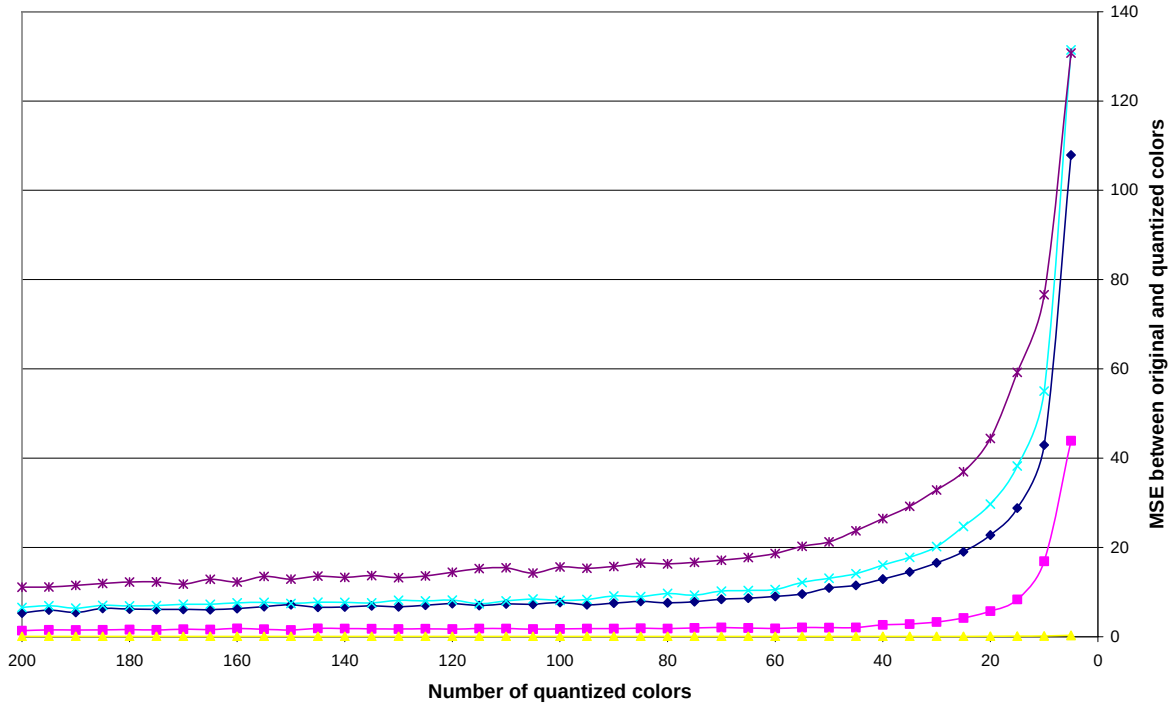


Figure 28: Evolution de la MSE en fonction du nombre de clusters

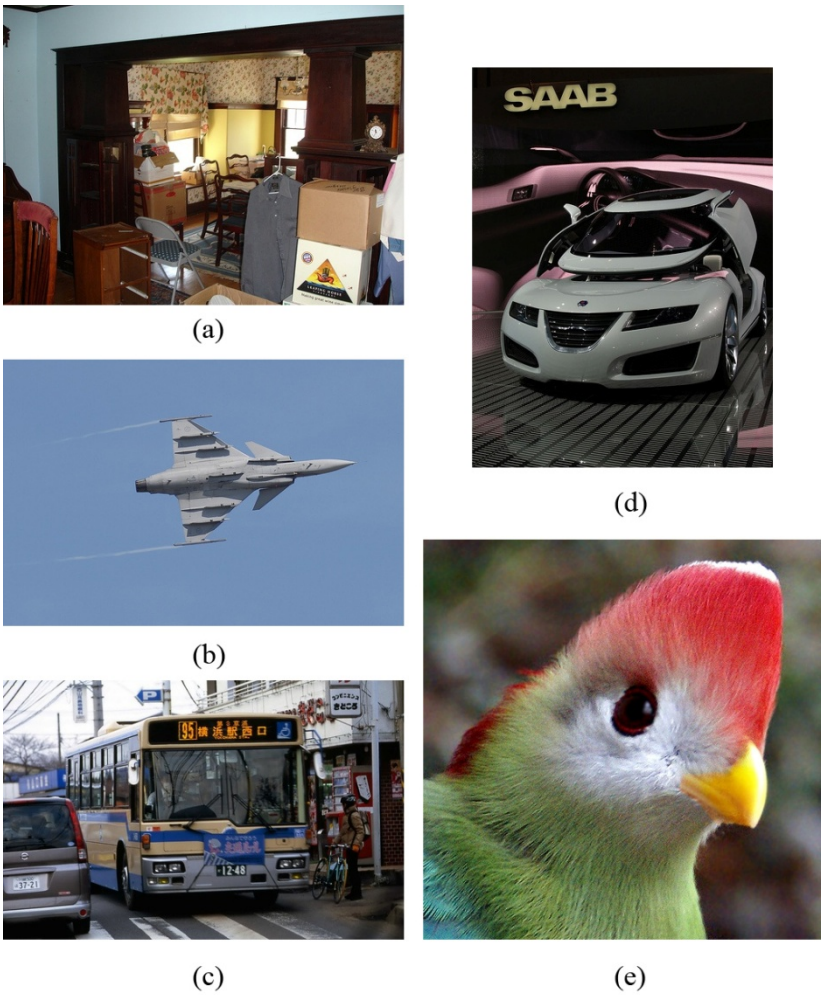


Figure 29: Images correspondant aux courbes (Figure 28)

Comme on peut le constater sur l'évolution des courbes MSE, celles-ci commencent à croitre de manière accélérée à partir d'un seuil qui reflète directement la complexité de l'image en termes de couleur. Comme nous allons le voir maintenant, le principe de la détermination automatique du nombre de régions est donc basé sur l'évaluation de ce seuil.

A noter que le choix de la MSE est tout simplement dicté par le fait que la plupart des évaluations des travaux actuels utilise la MSE. D'autres mesures existent et il s'agit d'une perspective intéressante d'étudier les comportements d'autres indicateurs voire de plusieurs indicateurs simultanément.

4.2.2. Calcul du seuil

Les calculs des différents points de la courbe présentée Figure 28 ont été faits avec un grand nombre d'itérations pour l'exécution des algorithmes "neural gas". Il va de soi qu'il est totalement inenvisageable d'implémenter la détermination de la courbe telle quelle puisque cela produirait des temps de calcul inacceptables. Pour l'application dans l'algorithme nous avons donc initialement choisi de réduire de manière significative le nombre d'exemples envoyés à l'algorithme pour l'entrainer et de le paramétrer pour en accélérer la convergence, au détriment de la précision : la zone d'augmentation que nous cherchons à détecter est en effet suffisamment marquée pour être détectée malgré des centres mal définis.

Initialement calculé par un seuil sur la MSE calculée par les exécutions de "Neural gas" dans nos travaux initiaux [118], nous avons amélioré la détermination du nombre de clusters en la rendant plus adaptable. Le procédé global est le même : on extrait des échantillons de la mesure de MSE pour essayer de localiser la zone où la croissance augmente rapidement.

Cette extraction se fait simplement itérativement par dichotomie, ce qui permet de limiter le nombre de mesures de MSE et donc la complexité de l'évaluation. On commence ainsi les acquisitions de MSE avec pour objectif d'atteindre un endroit où la MSE a été multipliée par γ par rapport à sa valeur initiale, puis on effectue des mesures successives de MSE jusqu'à localisation du nombre de couleurs où la MSE franchit ce seuil. Le facteur γ doit être choisi comme suffisamment grand pour ne pas être atteint prématurément à cause du bruit de la segmentation par "Neural gas". Avec les échantillons ainsi extraits, on peut également avoir une approximation du tracé de la courbe que nous utiliserons pour nos évaluations.

Nous avons ensuite voulu évaluer la sensibilité de la courbe à l'apparition d'une nouvelle couleur dans l'image. L'objectif est double : déterminer si ces courbes sont capables de refléter un changement élémentaire dans la palette et évaluer si le caractère aléatoire des centres déterminés par "neural gas" affecte cette capacité. Une des paires d'images que nous avons utilisées pour ce test est illustrée Figure 30. Pour chaque paire d'images, on effectue une série d'évaluations des deux courbes MSE pour déterminer leurs variations respectives. Les résultats que nous avons obtenus pour la paire d'images de la Figure 30 est présenté Figure 31. Ce graphe représente la sensibilité de la courbe ainsi construite à l'apparition de nouvelles régions, les barres verticales représentent les variations que nous avons obtenues sur les différents calculs de courbes. Sur cet exemple comme pour toutes les paires d'images que nous avons évaluées, on constate que les courbes moyennes sont distinctes et nettement séparables sur la portion précédant l'augmentation brutale de la MSE.

Figure 30: Images test pour vérifier la capacité des courbes MSE à détecter l'apparition d'une nouvelle région dans l'image

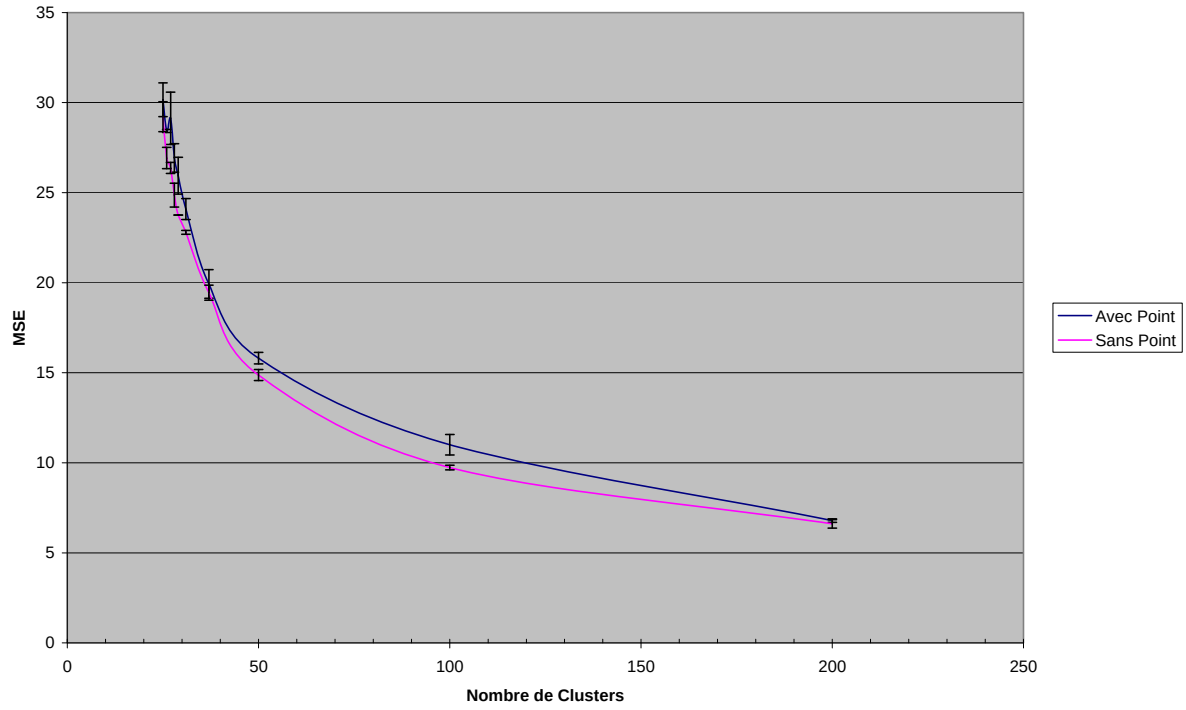


Figure 31: Evaluation de la séparabilité de deux images différant d'une seule région au moyen de courbes MSE échantillonnées par applications rapides de "Neural Gas" (courbe acquise avec $\gamma = 4$)

La zone ou la courbe adopte une forte croissance correspond à une zone ou les centres de couleur deviennent insuffisants pour décrire correctement l'espace de couleurs : c'est donc une zone où les centres sont particulièrement instables puisqu'ils n'ont pas de position satisfaisante vers laquelle ils convergeraient. Ce problème se traduit dans le fait que les segmentations à l'initialisation aléatoire peuvent converger vers des résultats assez différents (d'un point de vue optimisation, ceci correspond à un grand nombre d'extrema locaux). Pour augmenter la stabilité de détermination des premiers centres on les complète par quelques itérations de la variante du K-Means Generalized Lloyd Algorithm (le "neural gas" devient l'algorithme d'initialisation). Cette variante permet de réassigner les centres auxquels aucune donnée n'est associée, ce qui peut se produire avec notre faible nombre d'itérations de "neural gas". Cet ajout permet d'obtenir au final une MSE plus optimale donc de manière plus répétable, ceci n'est toutefois pas suffisant, en particulier parce que cela ne règle pas le problème des extrema locaux. Ce phénomène est particulièrement marqué sur la deuxième

partie de la courbe où il s'avère particulièrement difficile d'obtenir un nombre de clusters cible de manière totalement répétable.

Nous avons par contre remarqué que le nombre de centres devenant relativement faible dans cette zone, l'algorithme SICR dont la vitesse est essentiellement conditionnée par le nombre de centres, donne non seulement de bonnes performances en termes de quantification comme de vitesse, mais est en plus complètement déterministe. Des tests effectués en utilisant SICR pour tracer la courbe d'évolution comparativement à un algorithme de recuit simulé lent et coûteux mais plus susceptible de fournir des optima globaux nous ont permis de constater que la position de la zone où la croissance de la courbe s'accélère ne changeait pas. Nous pouvons donc déterminer rapidement le début de cette zone au moyen d'une série d'exécutions rapides de l'algorithme "neural gas", puis déterminer précisément le nombre de clusters à constituer par SICR.

Le principe de détermination des centres sera donc comme il suit : on recherche par dichotomie le début de l'inflexion de la courbe avec application d'algorithmes "neural gas" avec $\gamma = 2$ (recherche grossière du point d'inflexion – on n'utilise pas SICR pour des raisons de performance) puis à partir de là on applique SICR qui nous permettra d'extraire les centres de manière répétable. L'idéal serait de pouvoir étudier l'évolution de la courbe par sa dérivée mais, malheureusement, si l'évolution de la MSE est globalement comme décrit sur la Figure 28, elle est localement irrégulière, ce qui rendrait l'étude de sa variance particulièrement instable. Les modélisations par des fonctions exponentielles se sont également révélées peu précises. On va donc simplement rechercher par dichotomie un nombre de clusters tel que la MSE dépasse un seuil déterminé par rapport à la croissance de la fonction. On choisit comme référence un point d'ordonnée ε choisi tel que, quelle que soit l'image, la courbe ne soit pas dans sa zone de forte croissance pour $x= \varepsilon$. Le seuil S sur la MSE est donné par $S = \kappa.MSE_\varepsilon$. Afin de calculer S de manière déterministe, on calculera la MSE en ε avec SICR : il est donc souhaitable de prendre ε aussi petit que possible. Expérimentalement, le choix $\varepsilon = 100$, laisse une marge suffisante avant la zone de forte croissance que nous souhaitons explorer. κ sera lui notre paramètre de granularité de la segmentation. En effet κ sert à déterminer le seuil final correspondant au nombre de couleurs cible de la segmentation: on va utiliser l'algorithme SICR à partir de la borne supérieure déterminée grâce à gamma et jusqu'à obtenir une MSE correspondant au seuil obtenu grâce à κ.

L'algorithme de détermination suit donc les étapes suivantes :

- Préquantification de l'espace de couleurs avec l'algorithme de réduction perceptuelle de SICR
- Calcul de MSE_ε, MSE pour ε couleurs quantifiées par SICR
- Détermination de la MSE seuil S telle que $S = \kappa.MSE_\varepsilon$
- Détermination grossière du début de la zone "de forte croissance" de la MSE par exécutions successives de "neural gas" suivies d'itérations de GLA pour stabiliser les centres jusqu'à ce que la MSE calculée pour 200 couleurs quantifiées ait été multipliée par γ.
- On utilise SICR pour effectuer les derniers calculs de MSE jusqu'à encadrer S : $MSE_{t+1} < S < MSE_t$
- On retourne une partition de l'espace de couleurs par SICR appliqué pour t couleurs.

4.3. Traitement spatial de l'image

Une fois les couleurs sélectionnées, on doit assurer la continuité spatiale des régions déterminées on effectue donc un passage sur tous les pixels de l'image quantifiée par SICR. Le découpage spatial s'effectue selon la Figure 32

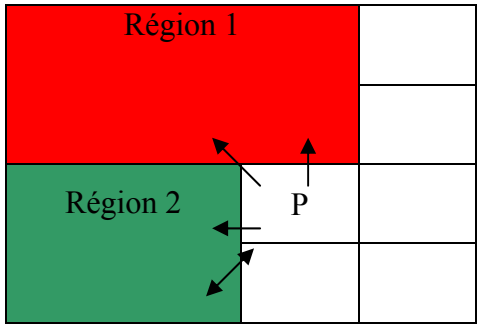


Figure 32: Illustration du procédé de séparation spatiale

L'image est parcourue de haut en bas et de gauche à droite. Les pixels sont associés Le pixel en cours P est comparé aux pixels pointés par des flèches pour savoir s'il se rattache à une région existante. La flèche double représente une direction particulière en effet il est possible que P appartienne à la fois à la région 1 et à la région pointée par la flèche double. Auquel cas il faudra fusionner ces deux régions. Notons que nous travaillons avec les couleurs quantifiées, la comparaison n'implique dont pas un calcul de distance mais une simple égalité : un pixel appartient à une région s'il est exactement de la même couleur que les pixels de cette région. Ce procédé de découpage en régions est également l'occasion d'établir une carte des contours des régions ainsi qu'un graphe d'adjacence : à chaque fois qu'une comparaison échoue, on a présence d'une frontière entre deux régions et on renseigne le graphe d'adjacence. A l'issue de cette étape, on obtient donc une image segmentée, un graphe d'adjacence et, pour chaque région, une carte de ses pixels de contours.

Si l'image reste exploitable à partir de ces premiers résultats nous avons un impératif supplémentaire qui est celui d'avoir des régions de taille suffisante pour en extraire des caractéristiques, il faut donc fusionner les régions de petite taille. Ceci se traduira par deux types de fusion : une fusion impérative basée sur un seuil en terme de proportions par rapport à l'image (les résultats d'une extraction de caractéristique depuis une région de cette taille seraient inexploitables et la région *doit* être fusionnée) et une fusion conditionnelle (la région est peu significative spatialement et ne doit être conservée que si la différence avec ses voisines est très marquée).

La première étape consistera à fusionner les régions de taille inferieure à un seuil donné. Comme pour tous les algorithmes de fusion il convient de prendre des précautions afin d'éviter le phénomène de chaînage. On constitue donc un arbre binaire qui pour chaque région racine tracera l'historique des fusions qui l'ont constituée. On adopte alors un principe hérité de l'algorithme de clustering "complete-link" à savoir que deux régions peuvent fusionner si les plus éloignés de leurs constituants peuvent fusionner. On distinguera deux fusions : la fusion de deux régions très proches et la fusion de régions "forcée" par leur taille. Le critère de proximité, dans le premier cas, est la distance de Fischer (28). Dans le second cas on compare les régions selon une mesure de distance colorimétrique.

Les critères de fusion sont les suivants :

- Pour toute région R de taille inférieure à ε_1, la comparer avec ses voisines en utilisant la distance de Fischer (28). Si une région voisine et/ou R est une région résultant de combinaisons on comparera les composantes élémentaires entre elles et on conserve la distance la plus importante.
- La combiner avec la région valide la plus proche selon cette distance si cette distance est inferieure à un seuil déterminé
- Combiner la région à la région la plus proche selon la distance de Fischer si la région est de taille inférieure à un seuil donné

4.4. Résultats expérimentaux

La Figure 33 illustre les différentes étapes de notre algorithme. En 1 est représentée l'image originale, en 2 on a le résultat de la segmentation après l'exécution de SICR suivie de la séparation spatiale des régions. On peut remarquer que le résultat est proche du résultat final et que l'algorithme ne dépend absolument pas de la phase de fusion. En 3 on a le résultat après la fusion basée sur la distance de Fischer. On remarque la fusion de certaines régions de petite taille, et des éléments comme le reflet dans l'œil, ou des régions dans la crête. Enfin en 4 on a le résultat final, privé des petites régions.

Les Figure 34 et Figure 35 illustrent quelques segmentations avec divers exemples en termes de complexité d'image et de diversité de couleurs On y constate des limitations inhérentes à divers aspects de la segmentation. Ainsi on se rend bien compte que l'avion, qui serait idéalement composé d'une seule région, ne peut pas être "correctement" segmenté étant donnée l'importance de la différence de couleur entre les parties ombragées et les parties éclairées. La photo du cavalier illustre le problème des régions floues qui sont fortement dégradées par le filtre médian. Ainsi le décor se transforme t-il en un amas de régions difficile à reconnaître. La photo du bus montre les problèmes posés par les reflets (sur les vitres du bus, principalement) alors que la dernière photo (issue de la base de Berkeley [95], les autres images étant issues de la base Pascal VOC 2007 [3]) montre les limites d'une segmentation basée sur les couleurs seules : le vêtement de la musicienne présente une texture homogène mais, sur le plan de la couleur, génère un grand nombre de régions parasites.

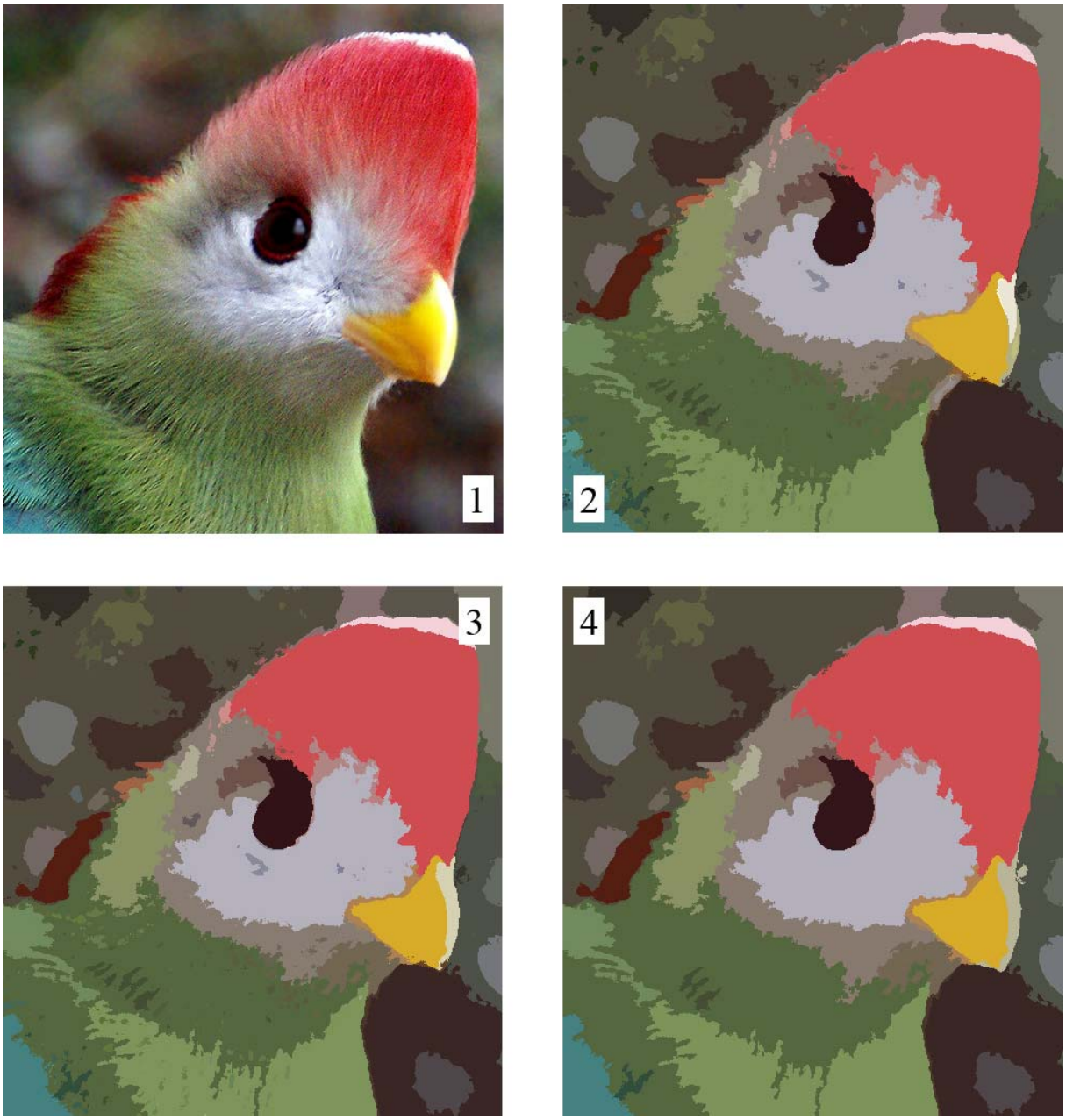


Figure 33: Etapes de la segmentation

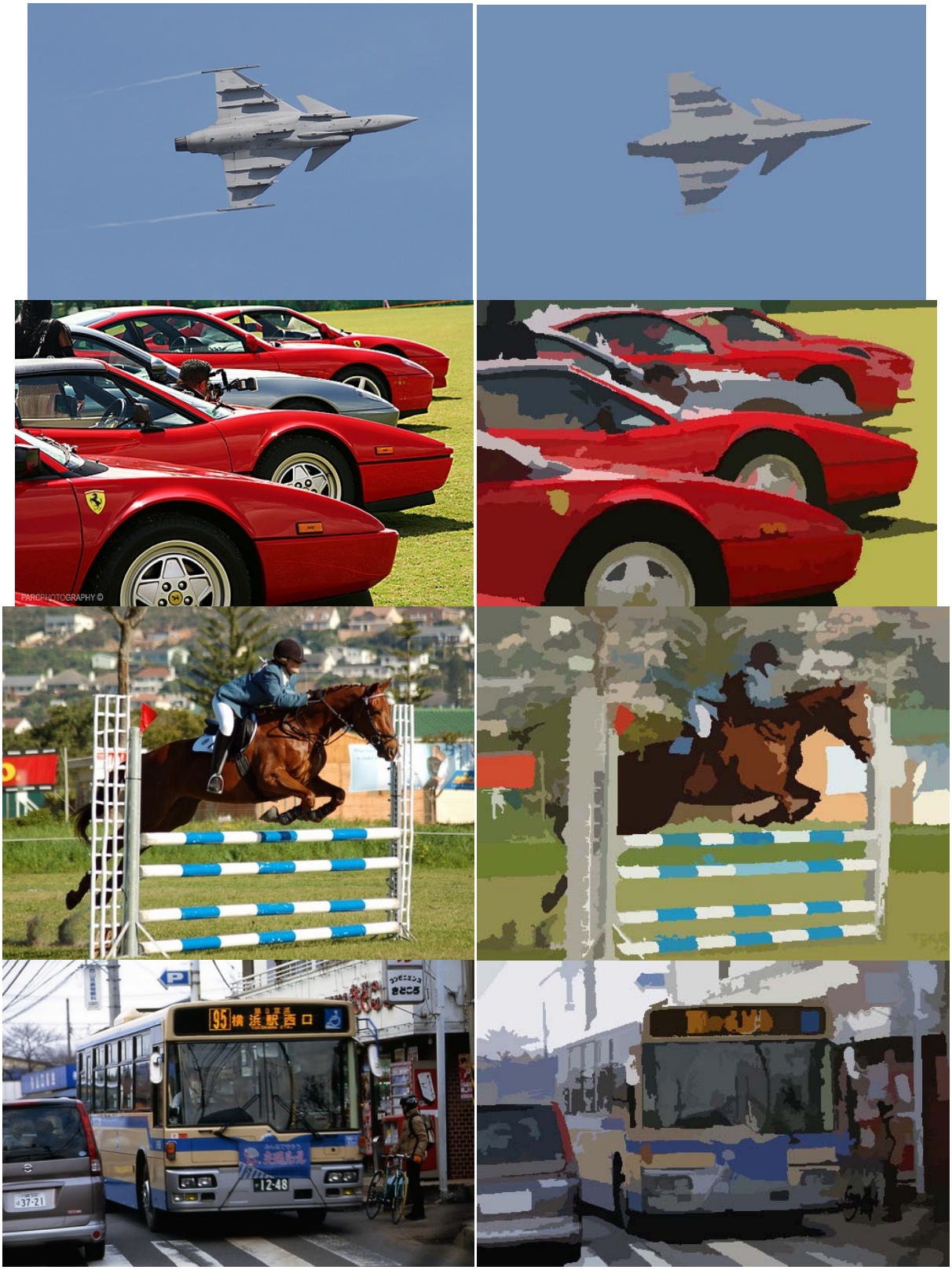


Figure 34: Exemples de segmentation

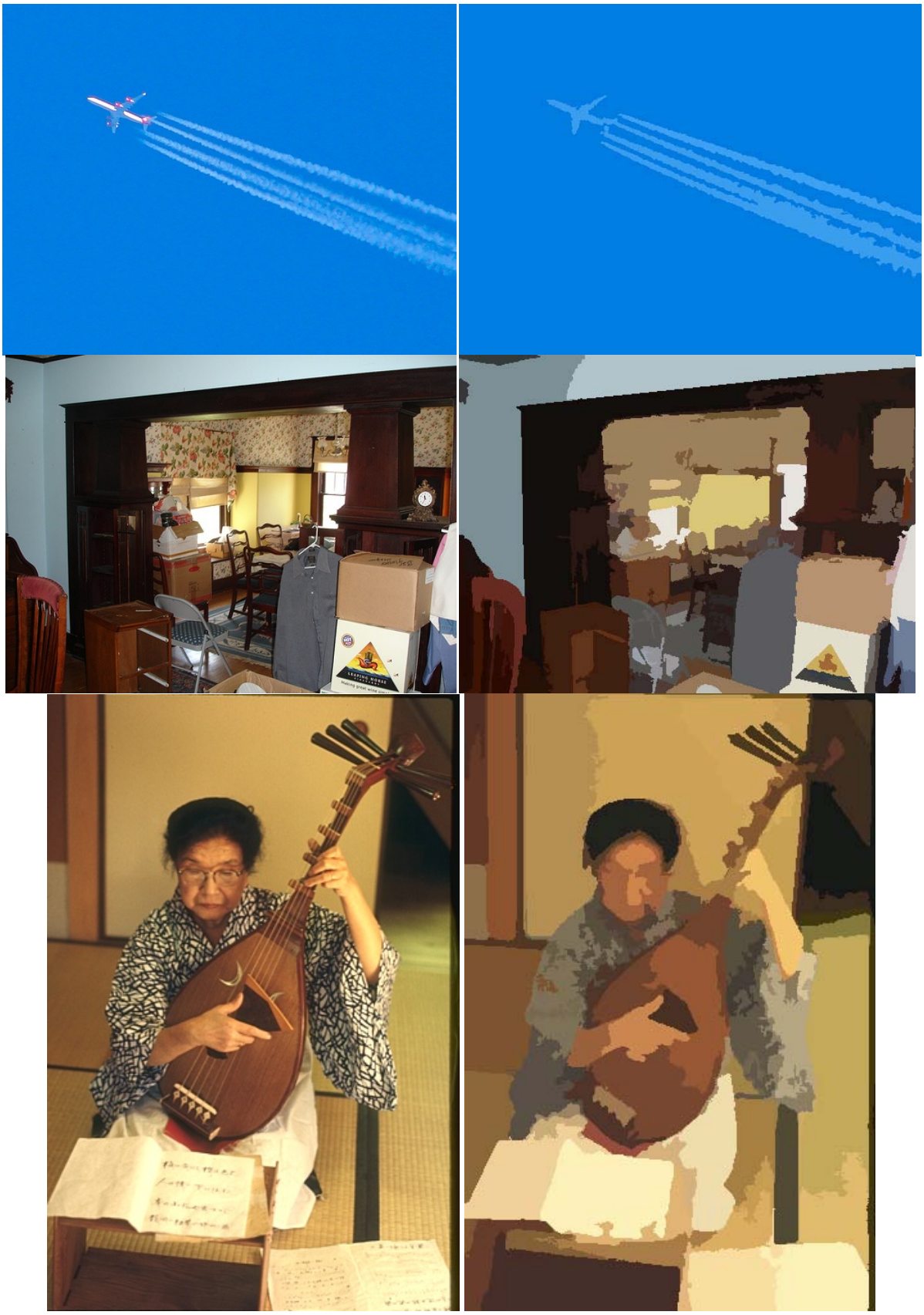


Figure 35: Exemples de segmentation

5. **Conclusion**

Dans cette section nous avons introduit un algorithme de segmentation robuste dont les paramètres affectent l'image de manière relative, permettant ainsi l'application à une base d'images complète sans changement de paramètres. Cette section comprend essentiellement deux contributions : une technique de réduction du nombre de couleurs dans une image qui intègre la notion d'information apportée par une couleur et l'algorithme de segmentation lui-même. Comme nous avons pu le constater sur les exemples proposés, des problèmes fondamentaux de la segmentation demeurent incontournables, il reste que cet algorithme nous permettra tout de même d'extraire des caractéristiques à partir de régions de taille significative sur des parties d'objets. Ceci a pour conséquence d'une part d'obtenir des données cohérentes entre elles et d'autre part de nous permettre d'intégrer l'information spatiale dans nos caractéristiques, grâce au fait que la partition de l'espace que nous réalisons est complète.

Ces deux contributions s'inscrivent dans notre démarche globale d'inspiration des mécanismes d'interprétation humaine. SICR, tout d'abord, en s'inspirant de la quantité d'information apportée par les couleurs, retranscrit le fait que les couleurs singulières soient plus susceptibles d'attirer l'attention. L'algorithme de segmentation, ensuite, applique dans ses principes de construction les lois Gestalt de continuité, de proximité et, par les procédés de filtrage et de fusion, de bonne forme.

Le test de cet algorithme sur la base pascal 2007 (9963 images) [3] nous a montré que l'algorithme s'adaptait à la diversité des images traitées sans produire d'images "aberrantes" (images composées d'une seule région ou au contraire d'une mosaïque de régions de petite taille. Nous allons maintenant évoquer notre contribution sur les caractéristiques que nous extrairons de cette région.

Chapitre 5: Extraction de caractéristiques visuelles basées sur les segments

Comme nous avons pu le voir dans le chapitre 3, il existe dans la littérature un nombre très important de descripteurs en vue de caractériser le contenu visuel d'une image. Dans ce chapitre, nous apportons aussi notre pierre de contribution et proposons quelques descripteurs basés sur les segments de droite que l'on peut extraire d'une image. Notre motivation est double. D'une part, nous souhaitons réaliser un compromis entre le "holisme ontologique" de la perception humaine (un élément est entièrement ou fortement déterminé par le tout dont il fait partie, un "tout" n'est pas un simple agrégat) et l'information locale que l'on peut extraire d'une manière au-delà du pixel ; D'autre part, nous souhaitons disposer d'un descripteur du contenu visuel qui reflète à la fois une information géométrique d'objets visuels et des propriétés de texture. Notre choix s'est orienté naturellement vers les segments de droite que l'on peut extraire d'une image ainsi que les divers descripteurs qui peuvent en dériver pour avoir un certain nombre de propriétés d'invariance (la rotation, la translation, l'échelle) ou de robustesse au bruit.

Le chapitre est organisé comme suit. Nous décrivons d'abord dans la section 1 notre méthode pour l'extraction de segments basée sur la technique de Fast Connective Hough Transform (FCHT) puis dans la section 2 quelques descripteurs basés sur les segments d'une image.

1. Extraction de segments par Fast Connective Hough Transform

Dans nos travaux, l'extraction de segments de droite est basée sur le travail d'Ardebilian et al. [15] développée au sein de l'équipe à l'origine pour la détection de lignes de fuite pour l'analyse de la vidéo. La méthode d'Ardebilian, appelée Fast Connective Hough Transform (FCHT), est une variante de la transformée de Hough qui exploite la dualité entre la droite dans l'espace d'image et le point dans l'espace des paramètres. En effet, la technique originale de Hough [Hough 62] représente une droite par l'équation $y = ax + b$, en lui associant le point (a,b) dans l'espace des paramètres. L'ensemble des droites passant par un point de l'espace d'image est alors représenté par une droite dans l'espace des paramètres. Aussi, la droite passant par un ensemble de points colinéaires dans l'espace d'image, est l'intersection dans l'espace des paramètres, des droites représentant ces points colinéaires. Si l'espace discrétisé des paramètres est représenté par des cellules d'accumulateur, la transformation des points de contours dans l'espace des paramètres se fait par l'incrémentation des cellules d'accumulateur, à chaque fois que l'équation $y = ax + b$ est validée pour un point (x,y) du contour, et une cellule (a,b) de l'accumulateur. Par la suite, un simple seuillage de l'accumulateur permet d'extraire les paramètres des droites présentes dans l'image.

Dans la suite de cette section, nous présentons d'abord le principe de FCHT et notre implémentation. Ensuite, nous illustrons la performance de FCHT sur quelques images et analysons rapidement la complexité de l'algorithme. Rappelons que notre objectif est de caractériser le contenu visuel d'une image, que ce soit de manière globale ou au sein d'une région.

1.1. Principe de "Fast Connective Hough transform"

Dans la transformée de Hough [127] de base, la paramétrisation d'une droite se fait par l'équation $y = ax + b$. Malheureusement, une telle paramétrisation n'est pas optimale car a priori l'espace des paramètres [a,b] peut ne pas être borné. Aussi, de nombreuses améliorations à ce principe de base sont proposées dans la littérature : décomposition de l'image en blocs, choix de points de contours aléatoires ont par exemple été proposés pour améliorer les performances comme le rapportent les études de Leavers [128] et Illingworth [129]. Ceci étant dit les algorithmes produits ne donnent toujours pas de performances satisfaisantes que ce soit en termes de robustesse ou de vitesse de calcul.

Proposé dans [15] par M. Ardebilian sur lequel nous nous basons, l'algorithme "Fast Connective Hough Transform" part du principe d'une exploration locale de l'espace et propose d'utiliser une paramétrisation polaire de droites par $r = x\cos\theta + y\sin\theta,\ 0 < \theta \leq \pi$. Ainsi on va utiliser une représentation paramétrique de chaque segment de droite de manière locale et centrée sur les points de contour. On recherche ces points de contour selon ses coordonnées P(x,y), l'image est parcourue selon les valeurs de x croissantes puis selon les valeurs de y croissantes. Les points servant à des segments détectés étant effacés au fur et à mesure on sait que l'angle θ que formera un segment de droite avec l'horizontale vérifiera nécessairement $0 \leq \theta < \pi$. On rentre alors dans une phase de détection locale où l'on parcourt l'image dans chaque direction θ tant que l'on a des points à explorer le long du segment. Le segment le plus long est conservé puis effacé alors que les autres sont simplement effacés. Une des optimisations de l'algorithme est de considérer des orientations discrètes et par conséquent de précalculer les valeurs de cos(θ) et sin(θ).

Les étapes de l'algorithme sont donc le parcours de l'image, la détection de segments de droite dans toutes les directions et leur suppression de manière séquentielle.

1.2. Notre implémentation : EFHT

Nous implémentons directement l'algorithme décrit ci-dessus avec quelques adaptations. L'algorithme produira comme résultat une liste de segments ainsi qu'une carte des segments de l'image qui à un point P(x,y) de l'image fait correspondre un segment de la liste. Lors de nos premières expériences faites pour valider l'utilisation de segments en tant que descripteurs de forme [130], l'utilisation de filtres de Sobel pour extraire les segments de droite a parfois posé des problèmes dans le cas de lignes épaisses qui généraient des faisceaux de segments. Nous choisissons donc d'utiliser le Canny Edge Detector [49] qui nous permet d'extraire des contours d'épaisseur 1 (voir chapitre 3). Nous choisissons également de retranscrire l'information de couleur en tant que vecteur plutôt que de manière décorellée [47], [48].

Un inconvénient de l'algorithme FCHT original est qu'il efface les segments au fur et à mesure : ceci est indispensable pour éviter de détecter plusieurs fois le même segment mais produit des "trous" dans tous les segments qui coupent le segment effacé. Ceci est compensé par une tolérance de l'algorithme qui autorise la détection de segments même interrompus par des trous de taille inférieure à un seuil donné. Ceci rend toutefois l'algorithme plus sensible au bruit. On choisit donc de conserver deux cartes des contours de l'image, une carte D(x,y) pour détecter le départ des segments et une carte C(x,y) pour détecter la continuité des segments.

Le principe de l'algorithme tel qu'il est implémenté est le suivant :

- Application d'un filtrage de Canny [49] sur des contours extraits par un gradient vectoriel [47], [48]
- Précalcul des valeurs de cos(q) et sin(θ) pour toute valeur entière de θ
- Pour chaque point de contour $P(x_p,y_p)$ détecté sur D(x,y) :
    - Pour chaque orientation θ
        - r=1
        - Calcul des coordonnées absolues $(x_{r\theta},y_{r\theta})$ du point défini par ses coordonnées relatives (r,θ) exprimées par rapport à p
        - Tant que $C(x_{r\theta},y_{r\theta})$ est un point de contours
            - Incrémenter r
            - Calcul des coordonnées absolues $(x_{r\theta},y_{r\theta})$ du point défini par ses coordonnées relatives (r,θ) exprimées par rapport à p
        - Stocker le segment S_θ de longueur r
    - Soit S le segment S_θ de longueur maximale
    - Si $S > S_{min}$ conserver S dans la liste de segments finale et rapporter les points de ce segment sur la carte de segments
- Effacer tous les segments détectés S_θ de la carte D

Le fait de se baser sur une carte complète C pour détecter les segments et de ne supprimer les segments que de la carte de détection D nous permet d'éviter les "trous" dans les segments. On peut toutefois toujours s'autoriser quelques "trous" dans les segments ou encore de petites variations d'angle θ et ainsi chercher des points contigus pas seulement en $(x_{r\theta},y_{r\theta})$ mais dans un voisinage autour de ce point. Cette implémentation sera référencée comme EFHT (Enhanced Fast Hough Transform) par la suite. Nous illustrons ses possibilités dans la section suivante.

1.3. Exemples

Nous allons tout d'abord appliquer la EFHT sur une image simple pour montrer son fonctionnement et le paramètre de robustesse que nous venons d'évoquer : la Figure 36 représente deux applications de la EFHT sur un rectangle tracé à la main.

Le premier exemple est appliqué avec aucune tolérance sur les variations de θ et aucune tolérance sur l'espace entre deux points d'un même segment. Comme nous le voyons les segments ainsi extraits sont morcelés par les imperfections du tracé : le trait intérieur du bas étant parfaitement droit il produit un seul segment, le trait extérieur du haut est par contre irrégulier et est décomposé en plusieurs segments de petite taille. On remarque également quelques discontinuités qui correspondent à des segments détectés mais qui ont été filtrés comme bruit à cause d'une longueur inférieure à 3 pixels (cette limite étant définie en paramètre).

Le second exemple est appliqué avec une tolérance de +/- 2 degrés sur les variations de θ et de "trous" dans les segments jusqu'à 1 pixel. On constate que les trous disparaissent et laissent place à des segments correspondant mieux aux côtés du rectangle. On a par contre d'une part une imprécision sur l'orientation des segments (ex. le segment intérieur gauche) et la présence de quelques segments "parasites" qui doublent des contours irréguliers.

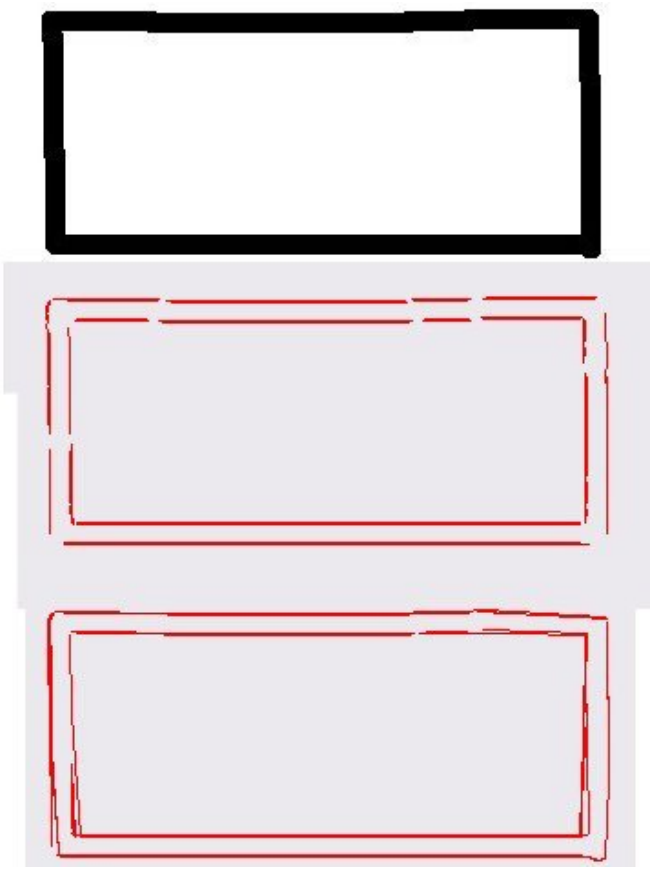


Figure 36: Exemple d'extraction de segments appliquée sur un rectangle tracé à la main

On utilise ensuite notre algorithme pour extraire des segments de droite à partir d'une photographie numérique issue de la base d'images de Berkeley. Le résultat est présenté sur la Figure 37. On remarque d'une part que les segments permettent de bien décrire l'image (on reconnait les formes de l'image d'origine) on remarque aussi l'importance de leurs caractéristiques de base que sont leur orientation et leur longueur. Ainsi les bâtiments sont constitués de segments horizontaux et verticaux, de plus on remarque que les segments longs ont tendance à être rattachés à des formes alors que les segments plus courts peuvent aussi constituer des informations de texture. Ces informations font de cette transformation un outil intéressant pour l'analyse d'image dès lors que les segments de droite dans une image ont tendance à caractériser à la fois la forme géométrique ainsi que la texture d'objets visuels dans les images.

Figure 37: Exemple de détection de segments sur une photographie numérique

1.4. Complexité

La complexité de l'algorithme est, de prime abord assez délicate à exprimer puisqu'elle dépend de la concentration de points de contour. En effet comme nous l'avons dit, pour chaque point de contour on va explorer son environnement dans 180 directions. Si on élimine toute tolérance sur les variations de θ et sur les "trous" au sein d'un segment on se contentera de rechercher les pixels de contour un par un, le seul cas où un même pixel sera évalué deux fois sera le rare cas de l'intersection de deux segments et on peut donc dire que la complexité de l'algorithme sera linéaire en fonction du nombre de points de contours au sein de l'image. Il convient toutefois de noter que rajouter des tolérances selon θ ou r augmente le temps de calcul de manière significative.

2. Extraction de descripteurs basés sur les segments

Un segment de droite peut être complètement décrit par 3 informations : sa localisation spatiale, son orientation et sa longueur. L'extraction de segments par EFHT met à notre disposition d'une part la liste des segments au sein de notre image et d'autre part la carte des segments associés à chaque pixel de l'image. A partir de là on peut extraire une variété de caractéristiques utiles pour la description du contenu visuel que nous allons décrire ci-après. Ces caractéristiques ont été comparées entre elles et par rapport à leur équivalent basé sur le gradient pour la classification d'images dans [131]. Nous reviendrons sur ces travaux dans le chapitre suivant sur la classification d'images.

Néanmoins, se pose d'abord la question de normalisation afin de doter les descripteurs des segments, respectivement les orientations et les longueurs, de propriétés d'une certaine invariance, comme par exemple par rapport à la rotation ou au changement d'échelle.

2.1. Normalisation

Comme nous l'avons vu dans la section sur les caractéristiques, le choix de la normalisation des caractéristiques de longueur et d'orientation se pose préalablement à l'étude de nos descripteurs. On considèrera donc ici la normalisation en orientation qui nous permettra de rester ou non invariant par rapport à la rotation de l'image en calculant ou non des orientations relatives à une référence locale. Ce choix est particulièrement important ; en effet une photographie numérique se lit selon une orientation définie et on peut donc considérer que les orientations absolues contiennent plus d'information que des orientations relatives. Il est ainsi bien plus pertinent de chercher à identifier une catégorie sémantique "ville" à partir d'orientations verticales et horizontales qu'à partir d'orientations relatives qu'on identifierait comme plus ou moins perpendiculaires. Cette normalisation apparait en revanche plus pertinente si on se place dans le contexte de reconnaissance d'objets visuels qui peuvent changer d'orientation d'une photo à l'autre, sur ce plan là on peut distinguer des caractéristiques qui seront plus adaptées à la détection d'objets visuels et d'autres qui seront peut être plus adaptées à la détection de catégories sémantiques.

La normalisation en orientation se fait par rapport à une orientation de référence qui se doit d'être choisie de manière stable et répétable. Dans notre cas faire une simple moyenne arithmétique des orientations ne suffit pas. D'abord parce que cette moyenne n'en est pas une : on voit bien que si on cherche à faire la moyenne de la série d'angles {90°, 300°, 3°, 12°} on est confrontés à toute une série de problèmes selon si on fait la division par 4 en appliquant ou pas le modulo 180° on trouve une "moyenne" de 11.25° ou de 101.25° respectivement. De plus la valeur obtenue change encore si on fait la moyenne des moyennes calculées séparément entre 90° et 300° d'une part et 3° et 12° d'autre part (avec à chaque fois des résultats différents). On remarquera enfin qu'avec ce système la "moyenne" sans modulo entre deux orientations horizontales comme 350° et 4° est de 177° (sens opposé) ; Le même problème apparait pour une opération avec modulo pour une moyenne entre 179° et 182° qui donne 0.5° (là encore sens opposé). On utilisera donc préférentiellement des statistiques circulaires [132], plus robustes par rapport au calcul de moyennes précédentes. Ainsi on utilisera par exemple la formule de moyenne angulaire suivante qui permet de choisir une référence d'orientation et permet des calculs répétables :

$$\bar{\theta} = \arctan\left(\frac{\sum_{i=0}^{N}\sin(\theta_i)}{\sum_{i=0}^{N}\cos(\theta_i)}\right) \quad (54)$$

Pour un ensemble de N angles θ_i, $i \in [0,n]$

La normalisation en longueur pose moins de problèmes : en effet les composants d'une même image ne sont pas nécessairement tous à la même échelle et il n'y a pas de bénéfice particulier à exprimer des longueurs de segments de manière absolue. Ceci est particulièrement vrai pour les objets visuels qui seront rarement représentés à une même échelle d'une image à l'autre. De plus comme nous le verrons dans le chapitre 6, les expériences de classification ont montré que la normalisation de la longueur des segments n'affectait pas les performances d'un classificateur. On pourra donc ainsi procéder à une normalisation en longueur "classique" en divisant les longueurs des segments par la longueur du segment le plus long.

2.2. Histogramme de segments

Il s'agit à l'évidence d'un descripteur le plus simple que l'on puisse imaginer : on constitue un histogramme à partir des données produites par l'ensemble des segments présents dans l'image ou dans une région. Un segment étant défini par sa longueur et son orientation, nous proposons ci-dessus deux types d'histogrammes explorant ces deux informations.

2.2.1. Histogramme d'orientation

Le premier histogramme est un simple histogramme d'orientations privilégiant les longs segments à l'image de celui proposé par Jain et Vailaya dans [61] et décrit dans le chapitre 2. En effet, pour tout segment d'un angle donné, on propose que la cellule correspondante de l'histogramme soit incrémentée par le carré de la longueur du segment. L'histogramme sera donc plus fortement affecté par les segments longs, plus rares, ce qui correspond bien à ce que nous avons évoqué au sujet de la perception humaine dans la section 2, à savoir qu'on accorde plus d'importance aux caractéristiques les plus rares. Notons enfin que pour une cellule représentant un intervalle de α degrés, la première cellule couvrira les angles entre 0 et $\alpha/2$ degrés et la dernière entre 180 - $\alpha/2$ et 180 degrés. Ceci est un choix que nous avons fait afin de mieux regrouper les orientations horizontales qui sont particulièrement significatives, que ce soit en absolu ou en relatif (colinéarité avec l'orientation de référence choisie). Les résultats donnés par un histogramme d'orientations absolues sur deux images exemple sont présentés sur la Figure 38. Les abscisses représentent l'indice de la cellule de l'histogramme, ce qui signifie qu'on a 36 cellules soit un intervalle de 5 degrés par cellule.

On constate que l'image de ville est bien retranscrite par deux pics très nets selon les orientations horizontales et verticales. L'impact des segments générés par les arbres et la silhouette du musicien est très faible étant donnée la longueur des segments qui définissent les bâtiments. L'image de montagne est, quant à elle, composée d'orientations plus réparties, avec toutefois une domination des orientations diagonales (ces orientations correspondent essentiellement à la partie gauche de la ligne de crête). Ce descripteur a produit d'intéressants résultats lors d'une tâche de classification globale (voir chapitre 6).

S'il n'intègre qu'indirectement l'information de longueur des segments, on constate que ce descripteur permet de bien caractériser le contenu de l'image. Ses limitations sont celles d'un histogramme classique, à savoir les effets de seuils inhérents à la quantification des données et l'absence d'information spatiale.

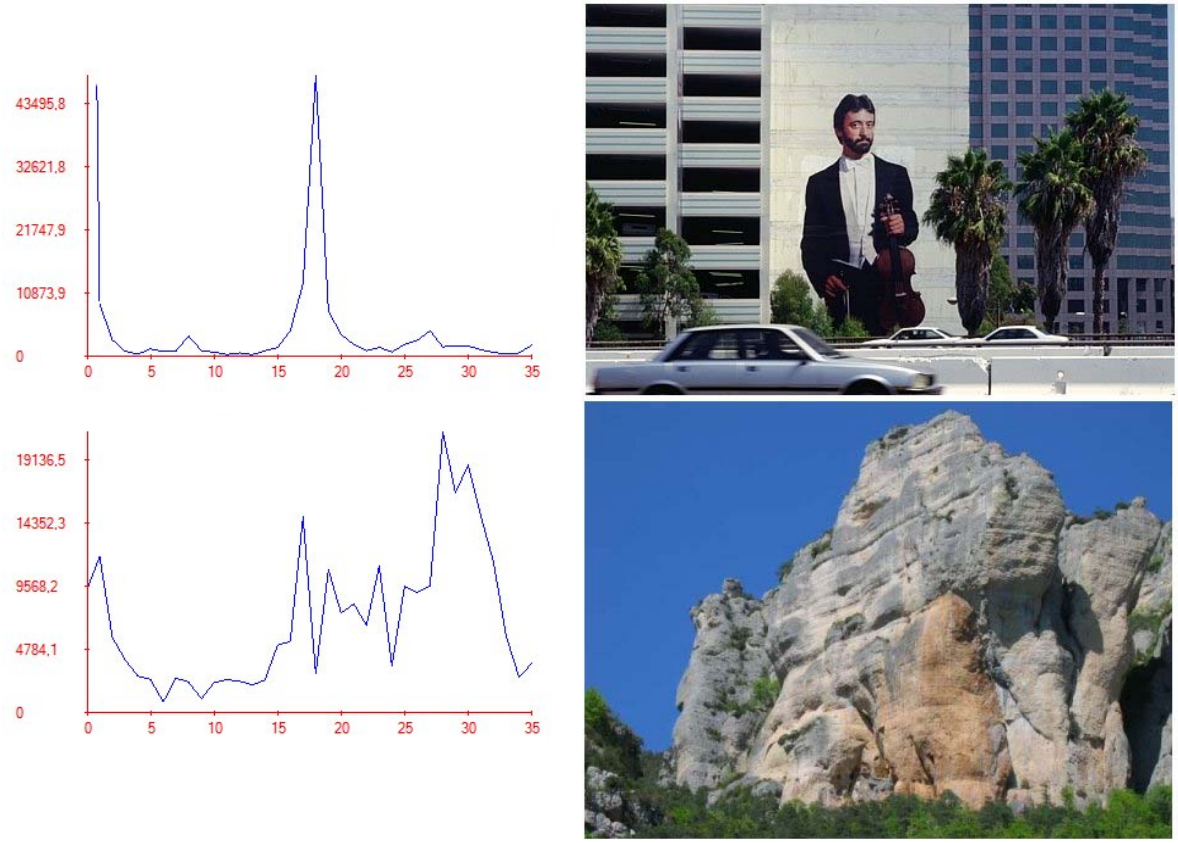


Figure 38: Illustration des histogrammes de contours EFHT

2.2.2. Histogramme de longueur et d'orientation

Le second histogramme est le plus intuitif : il représente directement les combinaisons orientation + longueur au sein d'un histogramme en trois dimensions où tout couple longueur-orientation (l,θ) sera associé à la population de segments $H(l,\theta)$ correspondant à l'intervalle de longueur et l'intervalle d'orientations défini par l'histogramme. L'avantage de ce type de structure est qu'il permet de relier explicitement la longueur des segments avec leur orientation. Avec le descripteur que nous avons présenté dans le paragraphe précédent, il n'est pas simple d'étudier les répartitions entre les différentes longueurs de segments pour une même orientation. Ce descripteur exprime donc mieux les informations fournies par les segments, même s'il conserve les limitations propres à un histogramme i.e. perte d'information spatiale. On rajoutera également un inconvénient inhérent à cette nouvelle structure qui est celui de sa taille par rapport au nombre de segments : en effet pour les segments de longueur les plus importantes il n'y aura à l'évidence que peu de représentants, ce qui signifie que les cellules de l'histogramme $H(l,\theta)$ seront majoritairement vides pour de grandes valeurs de l.

Ce descripteur représentant directement les informations produites par les segments, nous l'avons utilisé pour valider globalement l'approche orientée segments par rapport au descripteur SIFT dans une tâche de classification [133]. Le principe de cette expérience était

d'étudier le comportement séparé puis combiné de différents descripteurs (moments de couleur, histogramme de segments et SIFT) dans une tâche de reconnaissance d'objets visuels au sein de la base Pascal VOC 2007 [3] en utilisant une méthode de classification développée par notre équipe. Les résultats de cette expérience sont présentés en annexe 2, ils montrent clairement que le descripteur est efficace pour cette méthode de classification (nous verrons ultérieurement qu'il est également performant avec d'autres) mais il est particulièrement intéressant de constater que la combinaison avec SIFT améliore les performances. Nos informations extraites des segments seraient donc d'une autre nature et par conséquent complémentaires de celles extraites avec les descripteurs SIFT.

2.3. Matrice de cooccurrence

Une autre façon de représenter l'information des segments est d'utiliser les matrices de cooccurrence ce qui permettent d'extraire une notion de structure locale qui peut s'apparenter à une information de texture.

D'une manière similaire à la matrice de cooccurrence existant pour le gradient ([63], voir section 3) on quantifie les orientations θ en n cellules d'histogramme pour construire une matrice M de taille (n x n). La présence d'une valeur en M(i,j) désignant qu'un segment orienté selon j a été trouvé dans un voisinage défini d'un segment orienté selon i. Les valeurs étant collectées pour chaque pixel de tous les segments i de l'image. Ceci induit une redondance qui renforce l'importance des segments longs (ce que nous trouvons souhaitable étant donné leur rareté et leur importance visuelle). Par ailleurs, cette définition du voisinage peut être symétrique ("dans un rayon de k pixels") ou pas ("en dessous, à droite et à une distance inferieure à k pixels"). Le choix d'un voisinage symétrique revient à diviser la taille du descripteur par deux (la matrice devenant symétrique), alors que le choix contraire permet d'intégrer une information de position relative entre les éléments. Enfin pour un même pixel d'origine de i, chaque segment j ne peut être pris en compte qu'une seule et unique fois (le segment i étant lui-même pris en compte, la diagonale sera donc nécessairement incrémentée, même s'il n'y aucun segment dans le voisinage).

Pour ce qui est de l'intégration de l'information de longueur du segment, il apparait cette fois difficilement envisageable d'ajouter des dimensions à la matrice (faire correspondre des couples l,θ à d'autres couples l,θ aboutirait à un descripteur de dimension 4). En effet ceci alourdirait considérablement le descripteur. Même en quantifiant les données d'une façon très imprécise, par exemple en utilisant seulement 4 cellules d'orientation pour 3 longueurs, on produirait une matrice de 144 éléments (où on trouverait, là encore, beaucoup de zéros). D'une manière similaire à notre descripteur d'histogrammes d'orientation, nous allons donc représenter indirectement l'influence de la longueur du segment en reportant, pour tout couple de segments (i,j), le carré de la longueur du segment j. La encore on retranscrit l'importance prépondérante des segments longs. Un exemple de matrices extraites est donné Figure 39. Le vecteur de caractéristiques final est représenté par la concaténation des lignes de la matrice.

Le descripteur utilisé Figure 39 est une matrice de cooccurrence non symétrique (on recherche les segments situés à droite du point d'origine à une distance de 5 pixels), les orientations sont quantifiées de la même manière que dans la section précédente. L'orientation "1" représente donc une direction horizontale et l'orientation "4" une direction verticale. Afin de rendre le graphe lisible, les valeurs des diagonales ont été divisées par un fort coefficient ainsi que celles des valeurs des orientations 1 (on retiendra que ces valeurs sont très importantes sans considérer leur valeur absolue) sur le premier graphe.

Sur la première matrice, on remarque de manière assez caractéristique la forte présence de combinaisons des orientations 1 et 4. D'une manière plus générale on peut ainsi

caractériser une perspective par la forte domination de deux orientations, en l'occurrence 1 et 4. On retrouve ici la caractérisation des images urbaines par la forte présence d'orientations horizontales et verticales.

Sur la seconde matrice, on trouve une diagonale particulièrement marquée qui correspond à des segments colinéaires qui peuvent s'interpréter à la fois par des petites inflexions d'orientations voisines le long de la ligne de crête et également les lignes présentes sur les roches le long de la falaise. En dehors de cela on retrouve la forte présence des régions 4,5 et 6 qui correspond aux orientations "en biais" très marquées sur cette image.

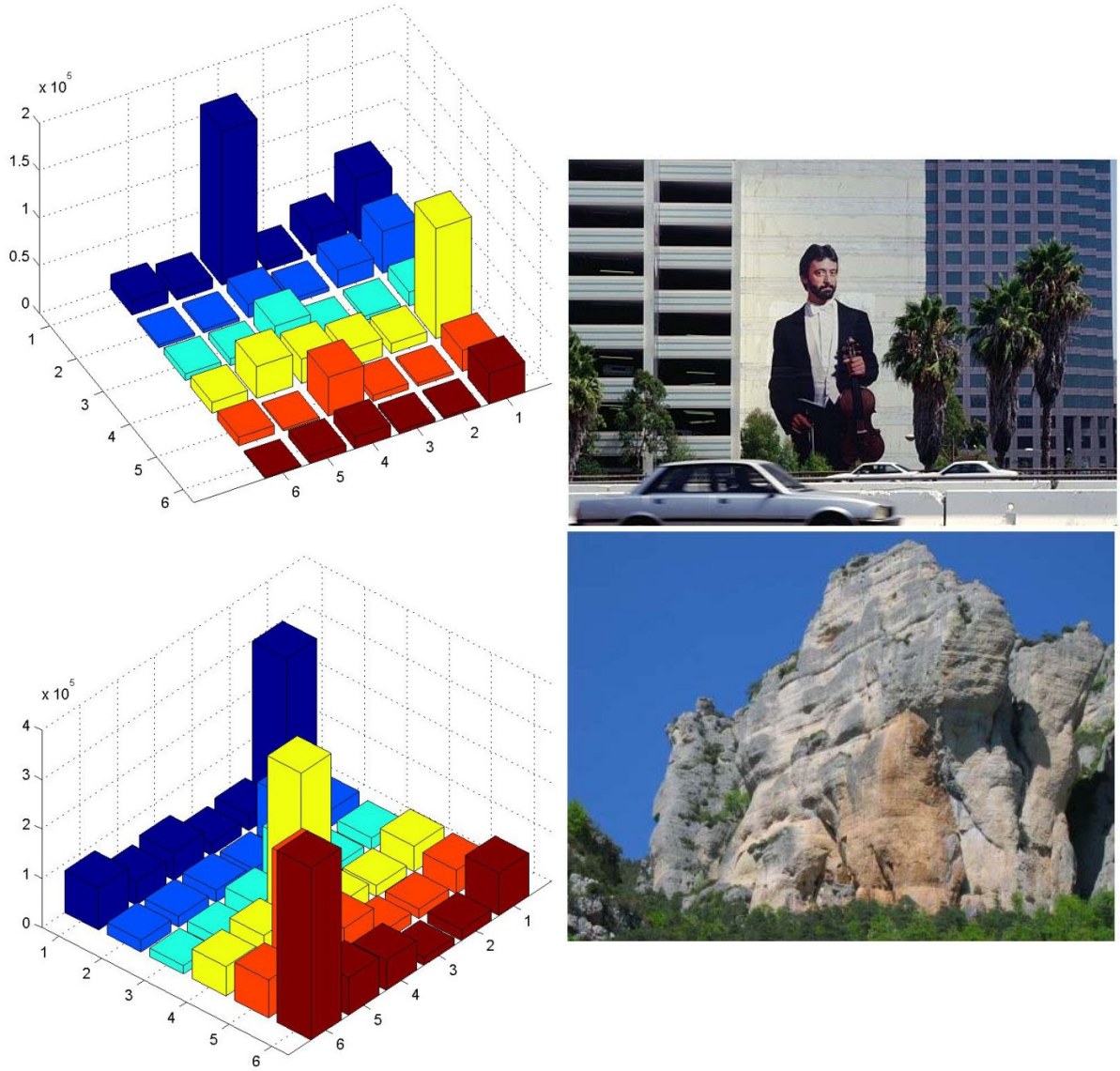


Figure 39: Illustration des matrices de cooccurrence EFHT

Nous présentons donc ici un descripteur qui va avoir la capacité de retranscrire des motifs locaux. On remarque également que la diagonale comprend les informations d'un histogramme classique ce qui permet de retranscrire une information plus complète et, comme nous les verrons dans les expériences de la section 6, plus performante.

3. **Descripteur spécifique "ville/non-ville" et résultats expérimentaux**

Afin de valider la pertinence des descripteurs du contenu visuel basées sur les segments, nous les avons testés en proposant un descripteur spécifique pour la classification ville/non-ville qui se basait sur l'identification d'orientations principales horizontales et verticales [130]. L'intuition est que les images de ville, ayant beaucoup de structures artificielles (route, bâtiments, etc.), devraient avoir une structure plus ou moins verticales et/ou horizontale. Le principe de ce descripteur est le suivant : détection de groupes de segments, évaluation selon leur orientation puis obtention d'un descripteur final composé de trois caractéristiques décrivant les groupes de segments horizontaux, verticaux et les autres.

Le regroupement des s'effectue selon un clustering de type "complete link" sur les orientations (voir l'état de l'art), puis on re-décompose chaque cluster obtenu selon le même algorithme pour les regrouper spatialement selon la distance de leur origine au coin supérieur gauche de l'image. Le but est ici de filtrer les groupes de segments similaires générés par une structure qui occupe une faible portion de l'image en regroupant les segments de même origine spatiale. Par ailleurs, seuls les segments de longueur significative (supérieure à un seuil déterminé – ici 5 pixels) sont pris en compte. La Figure 40 illustre la représentation des segments regroupés par orientation.

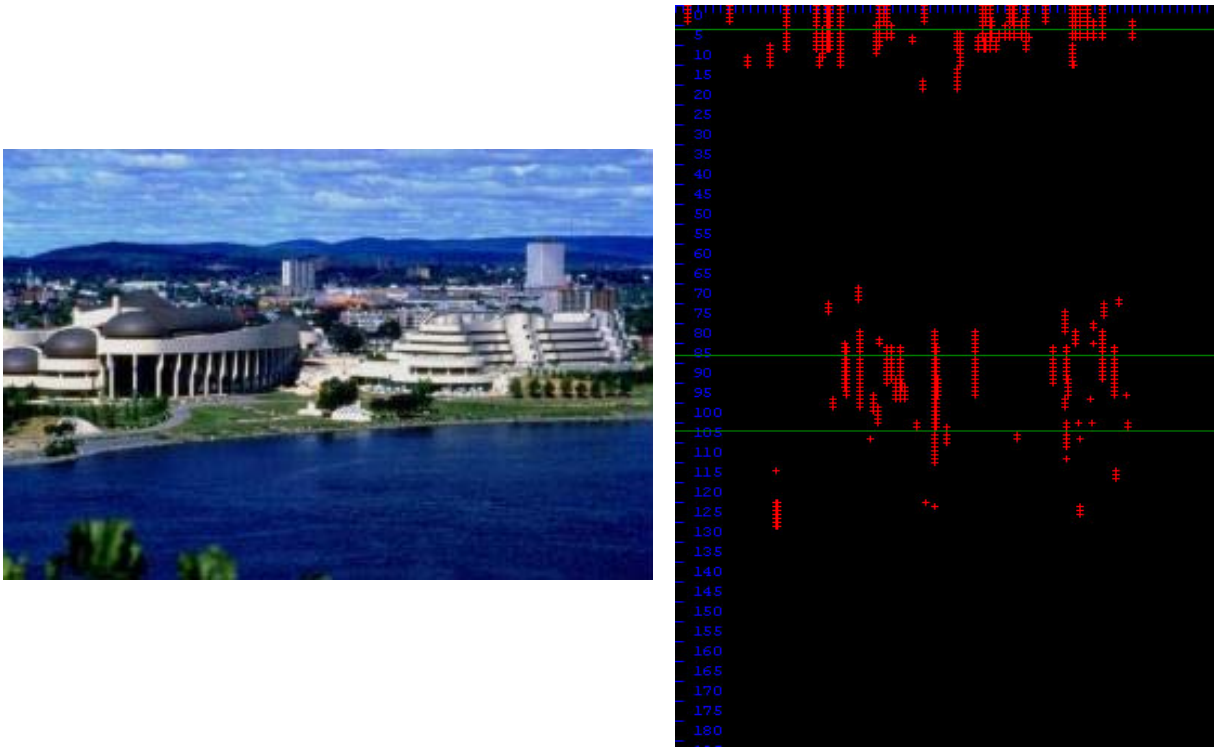


Figure 40: Représentation des segments sur une image exemple avec en abscisse la distance des segments au coin supérieur gauche de l'image et en ordonnée l'orientation du segment. Les lignes vertes indiquent les orientations principales détectées (groupes d'orientations)

Pour chaque groupe l'orientation moyenne des segments produit deux scores (pour les orientations horizontales et verticales) déterminés par deux gaussiennes centrées respectivement sur 0° et 90°, leurs variances σ_1 et σ_2 exprimant la tolérance qu'on autorise par rapport aux variations d'orientation provoquées notamment par la perspective. Pour une région donnée contenant N groupes de segments, on produit trois caractéristiques : la somme

H des scores des orientations horizontales H_i, la somme V des scores des orientations verticales V_i, et la somme P des scores des groupes "parasites" P_i qui correspond aux segments dont les scores horizontaux et verticaux sont trop faibles (55).

$$
\begin{aligned}
H &= \sum_{i=0}^{N} H_i \quad avec \quad H_i = e^{-\frac{\theta_i \bmod 180}{2\sigma_2^2}} \\
V &= \sum_{i=0}^{N} V_i \quad avec \quad V_i = e^{-\frac{\theta_i - 90}{2\sigma_1^2}} \\
P &= \sum_{i=0}^{N} P_i \quad avec \quad P_i = \begin{cases} 0 \;\; si \;\; \max(H_i, V_i) < \lambda \\ 1 - \max(H_i, V_i) \;\; autrement \end{cases}
\end{aligned}
\tag{55}
$$

Malgré sa faible dimensionnalité, ce descripteur très spécifique s'est révélé efficace pour cette tâche de classification spécifique "ville/non-ville", comme nous allons le voir.

3.1. Etude préliminaire : classification Ville/Non Ville

L'objectif de ce premier travail est de valider les descripteurs de segments sur un problème de classification spécifique. Il s'agit d'un travail préliminaire qui nous servira également à déterminer des axes d'améliorations pour la réalisation d'un algorithme de classification plus générique.

3.1.1. Caractéristiques et mode d'extraction

Les caractéristiques utilisées ont pour seul but de caractériser les orientations horizontales et verticales. L'intuition ici est qu'il existe plus de structures géométriques dans une image de ville que celle de non ville. Une caractéristique basée sur les orientations de segments devrait donc être discriminante pour ce problème. On utilise donc le descripteur simplifié que nous venons de présenter qui nous produit 3 simples caractéristiques (orientations verticales, horizontales, "bruit"). Bien que nous désirions dans un premier temps construire une approche "canonique", des premiers tests ont révélé un mauvais taux de classification sur certaines images qui comprenaient des forêts, les arbres produisant de forts contours verticaux. Notre descripteur d'orientation est ainsi complété par un descripteur de couleur aussi simple que possible qui considère l'application d'un noyau gaussien à un vert de référence (on choisit la valeur maximale de y dans l'espace CIExyz, voir Figure 6) qui décroit donc en fonction de la distance à cette couleur. Cette caractéristique, volontairement minimale, suffit à écarter la majorité des faux positifs générés par des images de forêt. Ceci produit un vecteur de caractéristiques à simplement 4 composantes.

L'image en entrée est décomposée en régions via l'algorithme CSC utilisé avec un seuil suffisamment important pour produire des régions grossières. Un exemple de régions segmentées est montré Figure 41.

Figure 41: Exemple d'image segmentée par CSC

La classification d'une région en ville/non-ville s'effectue au moyen d'un perceptron multicouche avec une couche cachée. Une taille produisant des résultats convenables a été établie à 10 neurones. Le verdict final de classification est donné sur un seuil fixé sur le pourcentage total de l'image couvert par les régions étiquetées comme "ville" (empiriquement fixé à 25%).

3.1.2. Résultats et conclusion

Le test de cette étude préliminaire est effectué sur une base de 300 images dont 50 sont utilisées pour l'entrainement et 250 pour le test puis on effectue une validation croisée. Les images ont été sélectionnées afin de représenter un corpus varié et difficile (voir quelques exemples en annexe 4). Nous avons toutefois volontairement évité de faire figurer des images d'intérieur qui, à l'évidence se confondent avec les images de ville selon notre critère de reconnaissance. Les régions des images d'entrainement ont été manuellement annotées. Les performances obtenues sont les suivantes (résultats moyens sur 5 validations croisées) :

	Rappel	Précision
Ville	0,81	0,87
Non-ville	0,88	0,82

Tableau 5: Performances du classificateur préliminaire ville/non-ville

Notre première conclusion sur ces résultats est l'efficacité des segments en tant que primitive de base sur ce problème de classification, ce qui motive l'extension du test à la classification générale et plus particulièrement sa comparaison directe avec des descripteurs basés sur le gradient. On note aussi que les résultats sont bons si l'on considère la taille du vecteur de caractéristiques, ce qui permet de montrer qu'un faible nombre de caractéristiques adaptées peut être suffisant pour caractériser des catégories.

L'étude des erreurs de classification dégage de nombreux axes d'amélioration. On remarque d'abord les erreurs de segmentations dues à l'utilisation d'une méthode dépendant d'un seuil. De plus, si les régions produites sont satisfaisantes et que l'algorithme ne tombe pas dans les pièges classiques posés par les dégradés, l'imprécision des frontières entre régions (induite par la tendance de l'algorithme à produire des régions hexagonales) est rédhibitoire. Ensuite la confusion entre des bâtiments et des structures artificielles peut s'attribuer du manque de précision de la description de la structure locale ainsi qu'au manque de finesse de prise en compte de l'environnement : le contenu des régions "non ville" n'est pas pris en compte et les caractéristiques proposées n'ont que très peu de finesse pour caractériser autre chose que les lignes horizontales ou verticales.

On peut conclure sur le fait que la méthode, bien que très probablement inférieure à celle utilisée par Vailaya et al. dans [62] montre clairement la validité de notre approche de descripteurs basés sur des segments extraits à partir de régions segmentées (dans l'étude de Vailaya, des résultats exceptionnels ont été obtenus mais sur une base plutôt facile et avec une proportion très importante de données dédiées à l'apprentissage ; l'utilisation d'une méthode de classification avancé et de vecteurs de caractéristiques plus complets laisse toutefois présager une très probable meilleure efficacité). On notera également l'idée, pour des travaux futurs, d'utiliser, éventuellement en nombre, des descripteurs compacts et très spécialisés.

4. **Conclusion**

Dans ce chapitre nous avons utilisé une adaptation de la Fast Connective Hough Transform [15] afin de proposer des descripteurs basés sur des segments de droite. Nous avons évoqué différentes représentations de ces descripteurs avec leurs qualités et leurs défauts respectifs. Nos descripteurs tendent à caractériser d'une part les informations géométriques d'objets visuels d'une scène mais également les informations locales en termes de texture. Ce qui est confirmé par l'expérience que nous avons décrit dans la section 2.2. et dont les résultats sont présentés dans l'annexe 2 : nos descripteurs basés sur les segments apparaissent être complémentaires avec les descripteurs SIFT et permettraient donc de compléter les approches classiques basées sur ces caractéristiques. Nous avons aussi évoqué la notion de normalisation qui nous incite à la distinction entre les descripteurs basés région et les descripteurs globaux. Enfin nos descripteurs correspondent à notre ligne directrice de nous inspirer de la perception humaine par la mise en relief des informations singulières ; le principe même d'utiliser des segments est en accordance avec les lois Gestalt de "bonne continuation", de proximité et de continuité.

Par ailleurs, si on considère des descripteurs basés sur les régions, il faut noter que la notion d'échelle ne se limite pas à l'invariance des descripteurs : il faut aussi prévoir qu'à une échelle différente, des régions peuvent fusionner voire éventuellement disparaitre. Ces problématiques seront abordées dans le prochain chapitre où nous allons intégrer les caractéristiques que nous venons de présenter pour constituer des plateformes de classification.

Chapitre 6: Classification sémantique d'images

Nous venons de proposer, respectivement aux chapitres 4 et 5, notre méthode de segmentation d'images ainsi que des caractéristiques basées sur les segments capturant des propriétés visuelles en géométrie et texture d'une image. Nous allons aborder dans ce chapitre le but ultime de nos travaux, à savoir la classification sémantique d'images. Nous allons voir que cet objectif de classification sémantique d'images se divise en deux problèmes différents : d'une part la catégorisation globale d'images et d'autre part la classification d'objets visuels.

Le chapitre est organisé comme suit. Nous décrivons d'abord précisément le problème de la classification sémantique d'images. Nous introduisons ensuite notre approche après un bref état de l'art. Nous décrivons ensuite nos travaux et expérimentations sur la catégorisation globale d'images d'une part et la classification d'objets visuels d'autre part.

1. Position du problème

Une classification sémantique d'images suppose tout d'abord que les classes cibles soient clairement définies. Or, étant donné une image de photo, on est tenté soit de mettre une vignette globale à cette image (ville, paysage, etc.) et l'on est dans le cas d'une catégorisation globale d'images, ou alors de décrire les objets tels que l'on voit à l'intérieur de celle-ci (bus, piétons, etc.) et l'on est dans ce cas dans la classification d'objets visuels au sein d'une image. Cependant, les deux problèmes ne sont pas mutuellement exclusifs dès lors que la catégorisation globale d'une image peut dépendre dans une large mesure d'objets visuels que l'on peut détecter.

Pour le procédé de décision lui-même, dans le chapitre 2 nous avons mentionné une approche qui nous était suggérée par la théorie des Schemas introduite par Piaget et la notion de vocabulaire visuel où on décrirait une image par une combinaison de caractéristiques visuelles. Ce principe a déjà été considéré pour la classification de texture avec le principe des "Textons" introduit par Leung et Malik [134], puis utilisé pour la classification notamment dans les travaux menés dans [70] ; nous étudierons ces approches un peu plus loin, avant de détailler notre version de ces principes de classification.

1.1. Catégorisation globale d'images

Au sujet du choix des catégories globales, on notera en particulier une étude réalisée par Vailaya et al. en prélude à la réalisation de leur plateforme de classification [136]: en demandant à un groupe de personnes de classer dans des catégories non prédéfinies un jeu de 171 images, il a obtenu une première série de catégories. Les exemples cités sont: bâtiments, rues, villes, ponts, monuments, personnes, paysages naturels, montagnes, fermes, campagne, forêts, couchers/levers de soleil. On voit ici des catégories qui sont d'une part non mutuellement exclusives et parfois très profondément ancrées dans la sémantique voire dans une culture qui serait difficilement à la portée d'une machine (tout particulièrement la classe monument). En recoupant ces résultats, les auteurs obtiennent les 11 catégories suivantes : Forêts et fermes, paysages naturels et montagnes, plages et scènes d'eau, chemins (routes et fleuves), couchers/levers de soleil, villes, ponts et scènes de ville avec de l'eau, monuments, tours, portrait et une catégories "autres" dans laquelle rentreront notablement les photos prises de trop loin pour que la machine puisse efficacement déterminer le sujet.

On imagine la difficulté d'un tel problème de classification. En effet, les classes précédemment proposés correspondent à des concepts hautement sémantiques dont l'apparence n'est pas clairement définie à la différence d'objets visuels dans les images qui

peuvent avoir une variation extrêmement riche d'apparences à cause des conditions d'éclairage, de variations pose, d'échelle ou encore de problèmes d'occlusion. Prenons l'exemple du concept paysage naturel. On peut imaginer qu'une image de paysage naturel contiendrait probablement des arbres, des prés, éventuellement quelques maisons lointaines, des collines, etc. mais ne possède pas une apparence proprement définie.

1.1. Classification d'objets visuels

S'il est difficile de catégoriser une image selon des concepts globaux comme on l'a évoqué dans la section précédente, on peut tenter au moins de classifier une image selon les objets présents au sein de celle-ci. La classification d'objets visuels vise à étiqueter une image de tout objet visible au sein de celle-ci. Ce problème, tout aussi difficile du domaine de la vision par ordinateur, a été popularisé ces dernières années par le Challenge Pascal [3] qui propose chaque année une compétition d'algorithmes pour la classification de 20 catégories non mutuellement exclusives :

1. Avion	6. Bus	11. Table	16. Plante en pot
2. Vélo	7. Voiture	12. Chien	17. Mouton
3. Oiseau	8. Chat	13. Cheval	18. Sofa
4. Bateau	9. Chaise	14. Moto	19. Train
5. Bouteille	10. Vache	15. Personne	20. Télévision

Des exemples de catégories de la base Pascal VOC 2007 [3] sont présentés en annexe 3. On peut constater la présence de certaines classes très difficiles à séparer deux à deux si on considère les caractéristiques que l'on peut extraire (ex. : le couple vélo/moto). Au vu des images présentées en exemple, on peut aussi remarquer que certaines sont en noir et blanc, et que d'une manière plus générale, les conditions de prise de vue sont particulièrement variées. Enfin les objets visuels sont présentés sous toutes les variétés possibles, que ce soit conceptuellement ou visuellement. En effet, si on considère par exemple la classe "train", d'une part on voit différents types de train comme des trains modernes, des locomotives à vapeur, etc. dans différentes situations (en circulation, à l'abandon, en exposition…) ; mais, de plus, les trains dans ces images apparaissent dans une grande variété de situations (de face, de côté, de loin, de près déformés par la perspective, …) et éventuellement affectés par des phénomènes d'occlusion. Il s'agit donc d'une base particulièrement difficile.

2. Bref état de l'art et notre approche

Un problème de classification comme celui de catégorisation d'images en concepts ou en objets visuels se distingue par la façon de former le vecteur de caractéristiques et le classifieur utilisé pour discriminer les différentes classes. Dans notre cas de classification d'images en concepts ou objets visuels, alors que nous sommes face à une extrême variation d'apparences en raison de conditions d'éclairage, de pose, d'échelle ou d'occlusion, nous disposons paradoxalement aussi très peu de données d'apprentissage en général. Une autre difficulté dans la classification d'images vient du fait que le nombre de vecteur de caractéristiques que l'on peut extraire d'une image est très variable en fonction du contenu visuel de celle-ci alors que les méthodes de classification actuelles exigent que le vecteur de caractéristiques soit de même dimension d'une image à l'autre pour leur catégorisation.

Dans cette section, nous faisons d'abord un survol rapide des travaux de la littérature. Dans la mesure où les différentes caractéristiques visuelles que l'on peut extraire d'une image ont été étudiées au chapitre 3 sur l'état de l'art, nous allons nous intéresser ici plus sur la modélisation d'image pour aboutir à un vecteur de caractéristique décrivant le contenu visuel

d'une image et utilisé en tant que tel pour la classification. Ensuite, nous introduisons notre approche à nos deux problèmes de classification sémantique d'images, à savoir la catégorisation globale d'images et la classification d'objets visuels. Nous commençons ce survol des travaux de la littérature par une comparaison des deux approches classiques de la classification : les approches génératives et les approches discriminatives.

2.1. Approche générative et approche discriminative

On distingue deux grandes façons d'effectuer la classification : le modèle discriminatif et le modèle génératif. Le modèle discriminatif va représenter les éléments que l'on souhaite classer par des points dans un espace à n-dimensions (voir plus haut). L'apprentissage aura pour objectif de tracer des frontières précises entre des ensembles de points représentant les catégories. Le modèle génératif, lui, va étudier d'abord la distribution des caractéristiques et la probabilité a priori $P(C_k)$ pour chacune des classes C_k à classifier pour pouvoir inférer la probabilité a posteriori $P(C_k/x)$ qui est utilisée dans la décision. Nous résumons ici les avantages et défauts de chacune de ces deux approches qui ont été comparées dans [135]

Pour les approches génératives :

1. *Elles peuvent prendre en compte des données manquantes ou partiellement annotées, elles peuvent également augmenter de petites quantités de données difficiles à annoter avec de grandes quantités de données générées*
2. *De nouvelles classes sont faciles à ajouter, il suffit d'en apprendre le modèle*
3. *Les modèles génératifs peuvent simplement prendre en compte la composition (rajout de lunettes, d'un chapeau ou de moustaches à un visage) alors que les modèles discriminatifs doivent recevoir tous les exemples possibles durant l'entrainement*

Et pour les approches discriminatives :

1. *Le modèle génératif observe uniquement la façon dont les exemples d'une classe sont construits alors que les modèles discriminatifs se concentrent sur les aspects qui distinguent cette classe des autres ce qui est potentiellement plus efficace pour prendre une décision.*
2. *Les modèles discriminatifs sont le plus souvent très rapides pour classer un nouvel élément alors que les modèles génératifs ont souvent besoin d'un certain nombre d'itérations*
3. *Un modèle discriminatif étant spécifiquement entraîné pour prendre une décision entre plusieurs classes, il devrait être plus efficace pour un modèle créé pour déterminer seulement la présence conjointe d'éléments constituants d'une classe*

Ce choix pilote aussi celui de l'algorithme utilisé par la décision. Les approches discriminatives vont ainsi utiliser le plus souvent des algorithmes d'apprentissage supervisés comme les très populaires SVM, des réseaux de neurones (Perceptrons, réseaux RBF, réseaux construits avec AdaBoost…) ; les méthodes génératives sont quant à elles probabilistes et se basent donc sur des modèles comme les "Mixtures de Gaussiennes" (GMM) ou les chaines de Markov cachées (HMM).

Si on considère l'approche par "vocabulaire visuel" envisagée, il s'agit d'une approche hybride au sens où l'on pourrait générer des images à partir du vocabulaire appris (modèle

génératif) mais la classification finale, elle, se fait par apprentissage supervisé sur les images apprises et détermine une frontière de décision (modèle discriminatif).

2.2. Catégorisation globale d'images

Finalement peu de travaux de la littérature traitent le problème de la catégorisation globale d'images tel que l'on a défini précédemment. Les méthodes de décision existantes de la littérature suivent un schéma classique de la classification par un apprentissage supervisé : un jeu de caractéristiques globales est extrait d'un jeu d'images exemples puis un apprentissage est effectué sur ces caractéristiques.

Une première approche ([136] et [137]) consiste à bâtir des systèmes hiérarchiques qui prennent des décisions de classification successives en se basant respectivement sur des classificateurs respectivement multi-classes et binaires. Dans les travaux de [136], les concepts ont été hiérarchisées pour produire un arbre de décision puis simplifiées pour obtenir des classes plus accessibles à une indexation automatisée. Wu et al. [137] propose un système de classification dans des catégories sémantiques en se basant sur un système hiérarchique. Des classificateurs propres à chaque catégorie sont constitué par des SVM entraînées pour discriminer sur le mode « concept A/ non-concept A », la hiérarchie du système définit ensuite des relations entre les concepts : un concept parent – extérieur – va favoriser la détection de concepts fils – nature et végétation, nature non-végétation –. Respectivement : des concepts qui ne peuvent pas cohabiter – intérieur/extérieur – vont se pénaliser mutuellement. D'autres approches, comme celle de Forsyth et al. dans [138], mettent directement toutes les classes en compétition dans un seul algorithme de classification (ici un arbre de décision).

On mentionnera quelques travaux qui proposent des approches originales. D'abord le travail de Vailaya [139], qui constitue par quantification vectorielle l'équivalent du "vocabulaire visuel" que nous décrirons dans la section suivante. Le "vocabulaire" ici est une modélisation de la distribution des caractéristiques de type GMM afin de calculer les probabilités d'apparition de chaque vecteur. La taille du vocabulaire est déterminée par une adaptation du critère MDL (Minimum Description Length) ; ce dernier a été introduit en 1978 par Jorma Rissanen (décrit dans [140]), et son objectif est de déterminer un codage idéal pour un ensemble de données selon le principe que toute régularité dans les données à coder est utilisable pour les compresser. Cet algorithme va comparer différents modèles en mettant en balance leur simplicité et leur capacité à décrire les données. Une image est classifiée grâce à la formule de bayes calculée au moyen du modèle GMM sélectionné.

Enfin Torralba et Olivia [141] ont proposé une méthode qui opère une analyse sur deux axes sémantiques. L'analyse globale se fait par l'étude des fréquences spatiales et des orientations (spatiales également). Une analyse en composantes principales sur un jeu d'exemples permet de déterminer des axes qualifiant le degré de naturel, le degré d'ouverture et la proportion de lignes horizontales par rapport aux lignes verticales. Une image peut ainsi être définie par ses coordonnées dans cet espace en trois dimensions comme une combinaison de ces trois caractéristiques de base.

Ces approches nous permettent à la fois de mieux cerner la problématique et nous donnent des pistes de résolution. Tous tentent de répondre au problème majeur du passage à la sémantique en l'abordant par une décomposition soit en étapes (succession de classifications binaires) soit en un ensemble de composantes dont la combinaison représenterait un motif qui correspondrait à une catégorie. Les classificateurs binaires ont l'avantage de présenter une problématique claire avec une seule frontière de décision, mais leur accumulation entraîne

nécessairement l'accumulation des erreurs de chaque classificateur. La logique A/non-A ne permet pas non plus de s'adapter directement à tous les problèmes, et en particulier ceux ou on introduit la non-exclusivité mutuelle des classes.

2.3. Détection d'objets visuels

D'une manière générale les approches dominantes (et les plus performantes) sont les approches de types "bag of features". Ces approches visent à caractériser un objet visuel par des statistiques sur des caractéristiques locales.

Un des premiers travaux sur ce principe est celui d'Ullman et al. dans [142] avec l'utilisation de blocs (ou fragments) dont la combinaison va caractériser une image. Les fragments en question sont choisis par entraînement sur un jeu d'images d'une classe. Leur méthode de sélection initiale n'est pas précisée mais au vu d'exemples donnés, ils semblent déterminés, au moins en partie, manuellement. Ensuite l'entropie relative des images par rapport aux fragments candidats est utilisée ; ainsi pour chaque classe, les fragments associés sont sélectionnés par maximisation de l'information mutuelle (calculée au moyen de deux jeux de tests : un représentant la classe à caractériser et un autre représentant des images d'autre nature). Deux évaluations sont effectuées pour cette mesure : son caractère informatif (la présence de ce fragment permet-elle de déduire à coup sûr la classe de l'image ?) et sa probabilité d'apparition dans la classe. Ces mesures sont également utilisées dans la phase de détection : elle permet de déterminer à quel point un critère est représentatif d'une classe lorsqu'il est présent. On notera la présence de fragments de tailles très diverses (les plus petits étant de 5x5 pixels) afin de pouvoir détecter à la fois des éléments rares mais informatifs (ex : un coffre de voiture), ou des éléments plus sujets à une erreur de détection mais très souvent présents (ex : détection d'un œil pour trouver une personne).

Une fois les critères déterminés les fragments sont recherchés dans l'image à classer. La recherche est faite en niveaux de gris et à différentes échelles, les mesures utilisées tentent d'être aussi invariantes aux conditions d'éclairage que possible (les méthodes utilisées permettent en fait surtout des variations d'intensité plus que des variations de sources). Finalement la décision est prise sur des critères probabilistes en se basant sur les évaluations faites lors de l'entraînement en combinant les informations apportées par les fragments présents.

Cette première approche pose les jalons des méthodes aujourd'hui dominantes dans la classification d'objets visuels basées sur une modélisation d'image selon une approche dite « bag-of-features ». Le terme "bag of features" provient de l'approche "bag of words" introduite pour l'analyse des documents textuels ([143], [144]). Ces méthodes modélisent une image comme étant une simple distribution de caractéristiques locales, typiquement les caractéristiques populaires SIFT, qui sont extraites de régions saillantes, appelées « points d'intérêt », ou encore à partir de points issues simplement à partir d'une grille. Cet ensemble désordonné de caractéristiques locales est alors caractérisé par un histogramme de « visual keywords » d'un vocabulaire visuel, ce dernier étant appris sur les caractéristiques locales extraites des images d'apprentissage soit par un regroupement dur (clustering) à travers une quantification ou alors un regroupement plus souple à travers des GMMs.

Actuellement les méthodes les plus performantes sur le Pascal challenge en 2007 étaient des méthodes de ce type. On citera les approches proposées par l'INRIA [70], [6], [7] et Xerox [5], [145]. Ces approches sont essentiellement basées sur les descripteurs SIFT même si dans [70] la complémentarité avec d'autres descripteurs du même type (RIFT et SPIN) est aussi étudiée.

Les travaux développés au centre de recherche Xerox (XRCE) sont parmi les premiers à exploiter cette approche de "bag-of-features". Le vocabulaire visuel, dans leurs travaux, correspond à des composantes gaussiennes de la distribution de caractéristiques locales extraites d'images d'apprentissage. Pour la décision de classification, ils intègrent un nouveau mode de classification en utilisant les "Fischer kernels", une sorte de SVM.

L'approche de l'équipe LEAR, à l'INRIA Rhone Alpes, propose deux mécanismes dont l'évaluation révèle des performances équivalentes. Le premier mécanisme est une approche de type "vocabulaire visuel" : un clustering est effectué sur l'ensemble des vecteurs de caractéristiques de la base d'entrainement via un algorithme de type K-Means avec un nombre de clusters déterminé empiriquement. Une image est alors caractérisée par un histogramme de "mots visuels" qui sont en fait les barycentres de clusters de descripteurs précédemment déterminés. Le second mécanisme proposé est de quantifier l'ensemble des vecteurs de chaque image toujours en utilisant un K-Means vers un nombre fixe de clusters. Le mécanisme de classification supervisée est alors appliqué sur cet ensemble de vecteurs quantifié. Comme on compare des ensembles de vecteurs, le mécanisme de classification est adapté afin de pouvoir manipuler ce type de données : il s'agit de SVM utilisant la distance $\chi 2$ ou EMD (voir chapitre 3) dans sa fonction noyau pour comparer deux histogrammes ayant les mêmes indexes ou non.

Plus généralement, les GMM et l'utilisation de l'algorithme K-Means sont les moyens les plus utilisés pour constituer le vocabulaire ; on y rajoutera l'algorithme "mean shift" utilisé dans [146] et l'essentiel des travaux dans ce domaine se fait sur l'algorithme de classification.

Bien que cette approche « bag-of-features » affiche la meilleure performance au challenge Pascal, elle souffre de plusieurs défauts qui sont inhérents à celle-ci. D'abord, l'idée de « visual keywords » inspirée des méthodes statistiques « bag-of-keywords » d'analyse textuelle trouve rapidement ses limites. En effet les "mots visuels", déterminés ici soit à travers une quantification ou des GMMs, n'ont pas une évidence contrairement aux vocabulaires textuels qui sont connus et définis. Dans le cas des images, il n'y a pas de correspondance directe entre les mots du vocabulaire et ce que l'on souhaite caractériser. Ce qui pose en conséquence un problème sur la taille du vocabulaire qui est actuellement fixée empiriquement dans les travaux de la littérature. Or la connaissance précise sur la taille du vocabulaire est fondamentale sur la performance des méthodes dans la classification d'objets visuels.

Un deuxième défaut de cette approche est lié à la modélisation du contenu visuel d'une image par une distribution de caractéristiques locales, perdant donc par la même occasion toute notion de relations spatiales sur ces caractéristiques locales. Or, on sait intuitivement que ces relations spatiales sont importantes pour les objets visuels.

Enfin, les caractéristiques locales étant extraites à de multiples endroits de l'image, celles-ci peuvent donc relever d'objets visuels recherchés ou simplement du décor. Il n'est pas simple de trouver un mécanisme pour les combiner entre elles tout en permettant de mettre en valeur les caractéristiques discriminantes extraites des objets recherchés par rapport aux caractéristiques extraites de l'arrière plan.

Plusieurs travaux de la littérature tentent de gommer ces défauts. On notera tout d'abord l'approche de Barnard et al. basée sur l'extraction et l'annotation de régions [147]. Cette méthode présente l'intérêt de s'affranchir du modèle classique du "bag of features" en recherchant l'information apportée par des grosses régions de l'image, ce qui correspond mieux aux principes de la perception humaine que nous avons énoncé dans le chapitre 2. Plusieurs principes d'annotation de chaque région sont proposés [148], par exemple un modèle basé sur un vocabulaire déterminé par clustering ou encore un modèle probabiliste

généré par l'algorithme EM. Cette annotation n'est pas définitive et est en fait constitué de l'ensemble des probabilités d'appartenance de la région à chaque classe. Le contenu finalement associé à une image est déduit des probabilités produites par les N régions les plus grandes [147]. Cette méthode suppose toutefois que les régions extraites correspondront effectivement à un ensemble identifiable de concepts et l'aspect de l'échelle d'observation n'est pas explicitement traité. On peut noter également le problème d'annotation de la base d'exemples qui peut s'avérer particulièrement fastidieuse. La nécessité d'annoter individuellement chaque région s'accompagne également de la pertinence d'une telle annotation : la segmentation peut potentiellement rassembler deux objets sémantiquement distincts ou au contraire produire une part non-identifiable d'un objet.

Des approches de la littérature proposent des modèles déformables de classes pour intégrer les relations spatiales entre objets ([149], [150], …). Il s'agit d'adaptations des modèles indéformables précédemment utilisés pour la détection d'objets (par exemple, le détecteur de piétons de Papageorgiou et Poggio dans [151]). Comme noté dans [152] ce type d'approches, bien que conceptuellement séduisant, obtient de moins bon résultats que les approches de type "bag of features" voire même que les approches basées sur des modèles indéformables, plus simples, mais plus robustes. Des améliorations à ces modèles, présentées dans [152] et [153] ont été appliqués au challenge pascal. Ce type de solution est intéressant car il permet non seulement d'identifier la présence d'un objet mais de le localiser. Ainsi dans [152] les auteurs reprennent des caractéristiques de contour locales qui définissent une partie. Un objet à reconnaitre est défini par une modélisation de la disposition des parties qui le composent à l'intérieur d'une fenêtre de détection, cette fenêtre étant appliquée à diverses échelles. Un tel modèle peut par contre souffrir de la nécessité de connaitre toutes les configurations possibles et d'un travail d'annotation délicat. Dans [153] on garde le concept de "mots visuels" dont on constitue un histogramme spatial hiérarchique à l'intérieur d'une fenêtre de détection. Cet histogramme s'obtient tout simplement en effectuant la détection sur une représentation pyramidale de l'image. Les mots détectés sont comparés à des modèles appris via une mesure de distance qui inclut la distance spatiale entre les mots. On retrouve par conséquent les difficultés de construction du vocabulaire inhérentes à la méthode des "mots visuels". L'intégration d'information spatiale via les modèles va également être bien moins efficace sur des objets déformables (dans [153], les tests sont effectués sur les classes "voiture" et "vélo" mais les résultats sur le challenge pascal VOC 2007 semblent confirmer ceci).

2.4. Fusion des informations pour la classification : fusion précoce contre fusion tardive

La plupart du temps on extrait une information multimodale de l'image en combinant diverses caractéristiques parmi celles mentionnées dans le chapitre 2 ; nous devons donc choisir une manière de combiner ces informations. Dans [154] Snoek et al. ont abordé ce problème de la stratégie de fusion dans le domaine de la vidéo (combinaison des caractéristiques audio et des caractéristiques extraites de l'image). On distinguera deux stratégies : la fusion précoce et la fusion tardive.

La fusion précoce est l'intégration la plus répandue et la plus intuitive : on effectue la concaténation des différents types de vecteurs de caractéristique extraits de l'image. La fusion tardive utilise un classificateur séparé pour chaque vecteur et fusionne les résultats en utilisant un autre classificateur. Cette méthode permet de limiter l'impact de la "curse of dimensionality" par rapport à une concaténation. Ces deux stratégies de fusion sont

représentées Figure 42 à partir d'un exemple extrait de [133] où on fusionne des descripteurs SIFT, des moments de couleur (RCM) et un histogramme de segments (RHS).

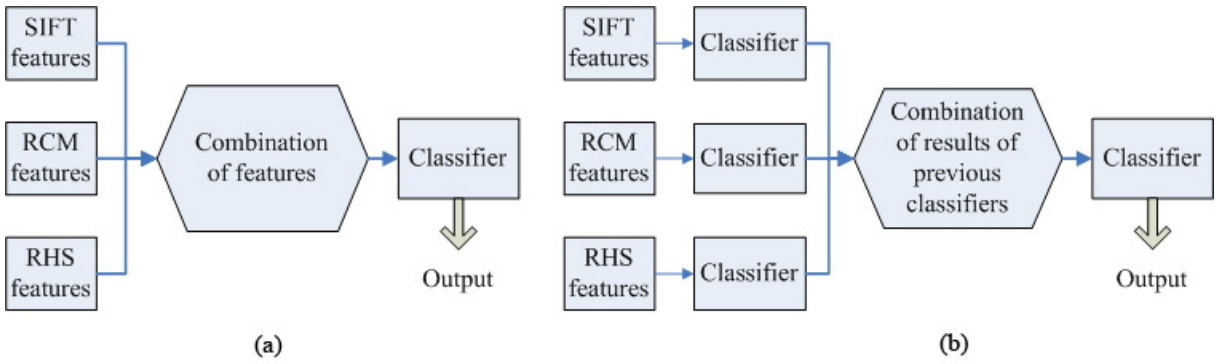


Figure 42: Deux modes de fusion de l'information: précoce (a) et tardive (b)

2.5. Notre approche

Par rapport à ces méthodes, nous proposons une approche basée sur des régions considérées comme parties non nécessairement identifiables d'un objet visuel (il s'agirait d'un ensemble Gestalt constitué par une ou plusieurs lois). Nous nous attacherons également à retranscrire la possible fusion de régions voisines, notamment pour traduire leur possible combinaison sémantique.

Dans nos travaux, nous suivrons aussi cette approche qui consiste à construire un vocabulaire visuel car c'est un moyen simple finalement de modéliser le contenu visuel d'une image par un histogramme basé sur un tel vocabulaire malgré la variation du contenu visuel d'une image à l'autre. Ici encore nous recherchons à nous inspirer des modèles proposés de la perception humaine par la théorie Gestalt [11], les travaux que nous avons étudiés comme [16] ainsi que nos propres études [17]. Rappelons que nous nous sommes déjà largement inspirés de ces principes dans nos travaux décrits dans les chapitres 4 et 5, respectivement sur la construction de descripteurs basés sur les segments ainsi que la segmentation d'image en régions. Nous nous plaçons ici au niveau de l'interprétation d'une image et notre but sera donc de tenter de mettre en œuvre l'approche holiste suggérée par ces travaux. L'agglomération de parties de l'image pour obtenir une structure en régions est une première étape dans cette direction. L'étape suivante est de retranscrire les relations entre les régions. Il s'agit d'un travail particulièrement difficile au sens ou les relations entre les différentes régions sont de natures variées et conditionnées par des caractéristiques tout aussi variables. Par exemple, l'inclusion d'une région dans une autre peut traduire une relation d'appartenance ou tout simplement deux éléments distincts placés l'un devant l'autre. Un humain distingue ces cas à partir d'indices complexes : présence d'ombres portées, voire interprétation du contenu des deux régions. Nous ne pouvons pas prétendre que notre segmentation isolera des objets reconnaissables indépendamment et même si c'était le cas, il n'est pas envisageable d'apprendre tous les objets que l'on pourrait rencontrer.

A défaut, notre approche pour modéliser les interactions spatiales sera donc de considérer différents niveaux de voisinage. Ainsi le vecteurs de caractéristiques d'une région sera obtenu avec les caractéristiques extraites de la région elle-même, concaténée avec le vecteur issu de la région augmentée des voisines et ainsi de suite : à défaut de pouvoir déterminer quels sont les regroupements pertinents on essaiera de représenter tous les

regroupements. Cette approche présente aussi l'avantage de fournir un début de solution au problème d'échelle qui se matérialise au niveau des régions par la disparition ou l'apparition de petites régions.
Pour résumer, inspiré des lois de Gestalt, nous souhaitons proposer des descripteurs basés sur les régions et ses voisines, comprenant nos propres descripteurs basés sur les segments, pour la classification sémantique d'images (catégorisation globale et classification d'objets visuels). Ces descripteurs seront comparés et combinés aux autres descripteurs de la littérature pour améliorer les performances de classification.

3. Notre démarche de classification

Notre approche consiste donc à appliquer le principe du « vocabulaire visuel » aux descripteurs extraits des régions d'images. Rappelons que cette notion de vocabulaire visuel était suggérée par la théorie des Schémas introduite par Piaget. Ce principe a déjà été considéré pour la classification de texture avec le principe des "Textons" introduit par Leung et Malik [134], puis utilisé pour la classification notamment dans les travaux menés dans [70]. Néanmoins, nous apportons plusieurs améliorations aux défauts d'une telle approche que nous avons mentionnée dans la section sur le bref état de l'art, notamment sur la construction du vocabulaire visuel qui est critique pour une telle approche.

3.1. Construction du vocabulaire visuel

Les caractéristiques sont donc extraites de chaque région et nous donnent un ensemble de vecteurs pour chaque image. La taille de cet ensemble (nombre de régions) varie évidemment selon la complexité de l'image. Ce nombre reste toutefois relativement faible comparé à la quantité de descripteurs de type SIFT que l'on obtiendrait avec les méthodes classiques. En conséquence, une modélisation de ces vecteurs de caractéristiques par des GMM apparait comme délicate faute de données suffisantes. Nous choisissons donc de modéliser notre vocabulaire visuel par un clustering sur l'ensemble des vecteurs de caractéristiques. Sur cette phase de clustering, comme nous l'avons mentionné dans la section précédente, deux problèmes se posent : la dimension des vecteurs de caractéristiques et la taille du vocabulaire.

Nous avons déjà parlé du problème de la "curse of dimensionnality" [14] et celle-ci affecte tout particulièrement les algorithmes de clustering, les distances entre deux points ayant tendance à être égales. Par ailleurs nos expériences dans [133] (cf. annexe 2) suggèrent que la fusion tardive génère globalement de meilleurs résultats. Nous proposons donc de générer un vocabulaire visuel propre associé à chaque type de descripteur (couleur, forme, texture, etc). L'effet est double : non seulement nous retardons la fusion des données (que nous désignerons par "fusion intermédiaire") mais en plus nous réduisons les dimensions de l'espace de clustering ce qui nous permet d'effectuer un clustering plus efficace et ainsi d'obtenir un vocabulaire plus précis.

Le problème de la taille du vocabulaire, lui, est assez rarement abordé dans la littérature et, comme nous l'avons vu, il est le plus souvent défini empiriquement. En réalisant une étude similaire à celle effectuée pour la segmentation dans le chapitre 4, nous avons découvert que le comportement de l'erreur quadratique moyenne évoluait de manière similaire aux couleurs à savoir une augmentation de plus en plus importante de la croissance de la MSE lorsque l'on décroit le nombre de clusters. Le nombre de mots (clusters) du vocabulaire peut donc être déterminé par observation de cette courbe (par exemple lorsque la pente de la courbe dépasse un certain angle). Dans un espace à forte dimensions, l'initialisation est particulièrement difficile (les vecteurs de caractéristique étant mal répartis dans l'espace beaucoup de centres risquent d'être mal positionnés) on utilisera donc un

algorithme neural gas pour une première détermination des centres peu sensible à l'initialisation affinée par quelques itérations de Generalized Lloyd Algorithm.

Notre approche nous permet donc d'une part de déterminer un vocabulaire avec une précision contrôlée et d'autre part de limiter l'impact de la "curse of dimensionnality".

3.2. Modélisation du contenu visuel d'images par une caractérisation floue

Une fois les vocabulaires visuels déterminés, il s'agit maintenant de caractériser le contenu visuel d'une image par un histogramme sur les mots visuels du vocabulaire. Dans la littérature, on utilise le plus souvent des algorithmes basés sur le clustering qui se contentent de rattacher les vecteurs de caractéristiques extraits d'une image au mot visuel le plus proche, ce qui pose des problèmes pour des vecteurs qui sont dans des zones ambigües. Dans nos travaux, nous proposons de ne pas définir une attribution définitive d'un vecteur à un mot mais au contraire une appartenance floue $u_{\sigma ij}$ du vecteur $x_{\varphi,\sigma i}$ à tous les mots $V_{\sigma j}$ du vocabulaire du type de caractéristiques σ (56) :

$$H_{\varphi,\sigma j} = \sum_{i=0}^{N_v} u_{\sigma ij} = \frac{1}{N} \cdot \sum_{i=0}^{N_v} \frac{1}{\sum_{k=0}^{N_\sigma} \left(\frac{d^2\left(x_{\varphi,\sigma i}, V_{\sigma j}\right)}{d^2\left(x_{\varphi,\sigma i}, V_{\sigma k}\right)} \right)} \tag{56}$$

Cette équation se base sur la formule d'appartenance floue du fuzzy C-Means définie par (37) et son comportement y est identique. Une image φ comprenant N régions et N_v vecteurs sera donc définie par ses histogrammes $H_{\varphi,\sigma}$ comprenant N_σ cellules, une par mot de vocabulaire du type σ. La valeur d'une cellule $H_{\varphi,\sigma j}$ est calculée grâce à la somme des appartenances floues de tous les vecteurs de l'image, l'histogramme étant normalisé par rapport au nombre de régions présentes dans l'image.

Cette caractérisation floue des caractéristiques visuelles vise à répondre à notre critique à l'encontre des vocabulaires visuels dont les "mots" ne correspondent à rien de précis, contrairement aux approches textuelles. Elle a pour but d'atténuer l'impact de choix d'un mot par rapport à un autre et constitue notre troisième contribution.

Si une évaluation poussée de cette méthode de classification n'a pas été conduite, nous avons, lors de l'utilisation de cette méthode pour la classification des images du challenge Pascal, tenté une classification en utilisant un histogramme classique (un vecteur est rattaché à un mot et un seul) : les performances ont été très médiocres comparées à celles décrites en fin de chapitre, étant, selon la catégorie, comprises entre 55% (proche de l'aléatoire) et 65% de classifications correctes.

3.3. Le procédé de classification d'images

On peut donc résumer notre procédé de classification d'images par les étapes suivantes :

- **Constitution du vocabulaire sur la base d'entrainement :**
  - Pour chaque image d'entrainement
    - Décomposition de l'image en régions
    - Extraction des caractéristiques de chaque type (couleur, forme, texture, etc.) pour chaque région
  - Pour chaque type de caractéristiques σ :
    - Effectuer une série de clusterings entre $N\sigma_{max}$ et $N\sigma_{min}$ clusters cible N_σ sur l'ensemble des images d'entrainement
    - Etudier la courbe MSE = $f(N_\sigma)$ et déterminer un nombre de clusters idéal avec $f'(N_\sigma)$ = seuil. On obtient un ensemble de centres V_σ qui correspond au vocabulaire de la caractéristique σ
- **Construction du classificateur :**
  - Pour chaque image d'entrainement φ
    - Pour les caractéristiques de chaque type, pour chaque vecteur, on détermine les histogrammes $H_{\varphi,\sigma}$ par (56)
    - Le vecteur de caractéristiques V_φ de l'image φ est la concaténation des $H_{\varphi,\sigma}$ (cas de la fusion intermédiaire, pour une fusion tardive, on classera séparément chaque $H_{\varphi,\sigma}$)
  - Entrainement d'un classificateur à partir de tous les vecteurs V_φ
- **Pour toute image d'entrée φ**
  - Décomposition de l'image en régions
  - Extraction des caractéristiques de chaque type (couleur, forme, texture, etc.) pour chaque région
  - Calcul des appartenances floues de chaque vecteur à chaque mot pour chacun des vocabulaires (37)
  - Calcul des histogrammes $H_{\varphi,\sigma}$ et construction du vecteur de caractéristiques V_φ
  - Classement de l'image en fonction du classificateur utilisé avec V_φ

Dans les deux sections suivantes, nous allons maintenant tester et comparer nos descripteurs orientés région et notre schéma de classification précédent sur les deux aspects d'une classification sémantique d'images, à savoir la catégorisation globale d'images et la classification d'objets visuels.

4. Catégorisation globale d'images

Rappelons que la catégorisation globale d'images a pour objectif de classifier une image en des concepts qui ne correspondent pas à des objets visuels dont l'apparence, bien que très variable, reste définissable. Dans cette section, nous allons étudier de manière séparée la pertinence de différentes approches (approche globale, approche basée régions) tout en étudiant les apports de nos descripteurs sous leurs différentes formes.

Cette section est organisée comme suit. Nous présentons d'abord la base Concept ECL qui est utilisée comme vérité terrain dans nos travaux. Nous proposons d'étudier deux approches dans la catégorisation globale d'images : d'une part une approche de classification globale qui consiste à utiliser dans un classifieur des descripteurs extraits directement d'une image sans que celle-ci soit segmentée et, d'autre part une approche de classification « locale » qui exploite les descripteurs extraits des régions et de ses voisins comme le préconise notre procédé de classification.

4.1. La base d'Images Concept ECL

La base Concept ECL a été construite en complétant les images de test pour la classification ville/non-ville par une série d'images collectées sur le même principe. Nous étendons le problème à un ensemble de 6 catégories extraites des catégories préconisées par l'étude de Vailaya [136]; ces catégories sont les suivantes :

- Lever/coucher de soleil
- Plage/désert
- Mer/Paysages maritimes
- Montagnes
- Verdure/Forêt
- Ville

Nous constituons ainsi une base de 600 images à raison de 100 images par catégorie. Des exemples sont présentés en annexe 5.

Les images ont été collectées en variant les conditions d'éclairage, le type de prises de vues et en incluant des images ambigües qui comprennent des éléments de deux catégories (villes entourées de forêts, plages lors d'un coucher de soleil, etc.). Le but est d'amener le classificateur à s'adapter aux conditions puis à "trancher" à la manière d'un humain si on lui imposait de faire un choix exclusif. On remarquera également qu'intuitivement les différents concepts choisis ont intuitivement des critères de discrimination évidents : par exemple si on considère la classe "Coucher de soleil" il apparait comme évident que c'est essentiellement la couleur qui va caractériser cette catégorie et que d'autres descripteurs vont générer du bruit. Au contraire des classes comme "Ville" seront a priori caractérisées plus efficacement par les contours et la couleur revêtira une importance secondaire.

4.2. Approche « globale » de catégorisation d'images

Notre approche dite « globale » pour la catégorisation globale d'images en concepts consiste à utiliser dans un classifieur des descripteurs extraits directement d'une image sans que celle-ci soit segmentée. Dans un premier temps, comme il n'existe pas de base commune dans la littérature à l'image de Pascal VOC, nous proposons d'établir une expérience de référence dans laquelle seule la couleur est utilisée pour la classification et à laquelle on peut mesurer les améliorations en utilisant d'autres descripteurs que la couleur. Ensuite, nous proposons d'évaluer la pertinence de nos descripteurs basés sur les segments en comparaison avec les descripteurs du type gradient en les combinant avec les descripteurs de couleur; Enfin, nous effectuons un test des différentes variantes du descripteur le plus performant afin d'étudier sa sensibilité aux variations de ses paramètres.

4.2.1. Expérience de référence par une classification basée sur les couleurs

Maintenant que nous avons établi la pertinence de nos descripteurs nous allons évaluer plus précisément leur efficacité. Nous allons dans un premier temps construire un classificateur simpliste qui nous servira de référence. Le principe de réalisation est le suivant : on utilise comme caractéristique les moments de couleur pour chaque canal de l'espace $CIELch_{Lab}$ (qui, comparé à CIELab, RGB et HSV a donné les meilleures performances) soit un vecteur de caractéristiques de dimension 9 (3 moments par canal, voir chapitre 2). Ce vecteur est extrait de manière globale à partir de l'ensemble des pixels de l'image. On utilise alors un perceptron multicouche (avec une couche cachée, de taille fixée expérimentalement à 20). Les valeurs des descripteurs sont normalisées (centrées et réduites).

Le Tableau 6 et le Tableau 7 représentent les performances du classificateur ainsi obtenu sur 5 jeux différents de cross-validation (sur deux ensembles) :

	Rappel	Précision
Montagne	0,392	0,4066
Verdure foret	0,71	0,655
Mer	0,58	0,5524
Ville	0,414	0,4836
Coucher de soleil	0,786	0,7081
Plage désert	0,66	0,7051

Tableau 6: Précision/Rappel pour le classificateur basé sur les moments de couleur

	Montagne	Verdure foret	Mer	Ville	Coucher de soleil	Plage desert	Total
Montagne	196	59	128	47	12	58	500
Verdure foret	39	355	9	51	25	21	500
Mer	110	22	290	32	19	27	500
Ville	78	76	57	207	68	14	500
Coucher de soleil	7	12	14	56	393	18	500
Plage désert	52	18	27	35	38	330	500
Total	482	542	525	428	555	468	3000
Taux d'erreur	**0,4097**						

Tableau 7: Matrice de confusion du classificateur basé sur la couleur (vérité terrain en première colonne)

4.2.2. Comparaison des descripteurs orientés segments/gradients

Par rapport à notre expérience de référence, nous allons d'abord à évaluer la performance des descripteurs basés sur les segments par rapport à des descripteurs basés simplement sur le gradient introduits dans l'état de l'art. Nous allons tout simplement utiliser nos descripteurs de matrice de cooccurrence et d'histogramme de segments décrits dans le chapitre 5 et les comparer à leurs équivalents basés sur le gradient (décrits dans l'état de l'art). On notera que même si les deux types de caractéristiques partent du même type d'informations (gradient) les modes de calculs sont très distincts. La comparaison à donc lieu d'être. Les caractéristiques sont conçues pour avoir à peu près la même taille (36 éléments). Ces caractéristiques sont combinées avec un simple histogramme de couleur (de taille 36 également, 12 cellules par canal de CIELch, les 3 canaux étant décorellés). On notera que nous avons cherché à pondérer également le nombre de caractéristiques de chaque type, nous nous retrouvons donc avec un descripteur de couleurs légèrement moins performant que celui de l'expérience de référence.

Le procédé expérimental est le suivant : les caractéristiques sont extraites de manière globale sur toute l'image puis sont centrées sur 1 et normalisées à une variance de 1. Avec les vecteurs de caractéristique ainsi obtenus, on effectue une série de 5 tests de validation croisée sur 10 ensembles toujours avec le perceptron multicouches dont l'architecture a été présentée pour l'expérience de référence. Les résultats sont présentés sur le Tableau 8.

	Histogramme				**Cooccurrence**			
	Gradient		**Segments**		**Gradient**		**Segments**	
	précision	**rappel**	**précision**	**rappel**	**précision**	**rappel**	**précision**	**rappel**
Lever/coucher de soleil	87.31	70.2	76.96	72.8	74.95	68.8	83.47	79.8
Paysages maritimes	72.26	67.2	71.81	64.2	63.53	55.4	72.75	64.6
Montagnes	53.86	62.8	56.17	62.8	54.63	60.2	55.2	64.8
Plage/Désert	65.7	63.2	64.57	59.4	57.97	53.8	67.63	60.6
Forêt/verdure	61.79	65.0	70.66	76.6	71.77	78.8	74.85	77.4
Paysages urbains	59.67	64.8	68.02	70.6	78.93	85.4	80.99	85.2

Tableau 8: Comparaison de caractéristiques basées sur le gradient et sur les segments

Notons que nous avons mené, parallèlement à cela, une étude afin de déterminer l'importance de l'algorithme dans la classification. Sans surprise les résultats montrent que si les taux de classification correcte diffèrent, la comparaison des différentes techniques donne les mêmes résultats. Les détails de cette étude se trouvent en annexe 6. De plus, nous nous attachons seulement à comparer les deux types de descripteurs, l'étude plus précise des résultats selon chaque catégorie se fera dans les sections suivantes.

Le résultat montre que nos caractéristiques améliorent significativement les taux de classification pour des catégories où les segments ont intuitivement une forte importance. Ainsi les classes "paysages urbains", "forêt verdure" et, dans une moindre mesure, la classe "montagnes". On remarque également que les caractéristiques de cooccurrence obtiennent des résultats nettement supérieurs aux histogrammes. Ceci s'explique par le fait qu'ils capturent également l'information de structure locale ce qui correspond à une information de type

texture supplémentaire. On constate par ailleurs une baisse de performance notable sur des classes où l'information de contours apparait comme peu importante. Il est possible que cela soit du à l'exclusivité mutuelle des classes qui poussent certaines images à basculer dans une classe qui, bien que présente, n'est pas la leur. Il est aussi tout simplement possible que le descripteur génère du bruit dans le classificateur

4.2.3. Catégorisation globale d'images basées sur les matrices de cooccurence de segments

Nous venons de voir la pertinence de nos descripteurs basés sur les segments ainsi que l'avantage de la matrice de cooccurence sur le simple histogramme. On va réaliser maintenant un classificateur en intégrant diverses formes du descripteur de matrice de cooccurrence afin d'en déterminer la forme la plus performante. Nous avons tout d'abord testé l'impact de la forme de la matrice de cooccurrence elle même. Nous avons vu dans le chapitre 5 que l'on cherchait les segments dans un voisinage sous deux formes :

- Une forme localisée pour tout point P(x,y) appartenant à un segment, on examine tous les points $P_n(x_n,y_n)$ à une distance inférieure à D et tels que $x>x_n$ et $y>y_n$.
- Une forme invariante en rotation : pour tout point P(x,y) appartenant à un segment, on examine tous les points $P_n(x_n,y_n)$ à une distance inférieure à D, quelle que soit leur direction. Ceci aboutit à un descripteur plus compact puisque la matrice de cooccurrence devient symétrique et on peut donc se contenter de la matrice diagonale supérieure pour exprimer les caractéristiques.

Par la suite, les paramètres évalués sont essentiellement l'influence de l'invariance (doit on rendre les segments invariants en échelle et en rotation) et la taille de la matrice (nombre de cellules pour représenter les angles). Nous avons fait varier ces paramètres et nous entrainons un classificateur (MLP, comme dans l'expérience de référence) composé du descripteur de couleurs basé sur les moments (présenté au chapitre 3) et d'un descripteur basé sur les matrices de cooccurrence de segments (voir chapitre 5). Comme nous allons le voir, différents modes de normalisation comme de calculs de la matrice sont explorés.

Notons que pour toutes ces expériences nous avons fixé D à 5 qui s'est révélé être le meilleur compromis entre temps de calcul et performances de détection. Pour le reste, les conditions expérimentales sont exactement les mêmes que pour l'expérience de référence (5 jeux de cross validation sur deux ensembles). Les résultats de ces expériences sont présentés dans le Tableau 9.

Les résultats des expérimentations sur ces deux types de caractéristiques sont représentés en (a) et en (b) montrent que les deux formes de voisinage envisagées sont équivalentes si on considère la variance des résultats entre les différents tirages de cross validation que nous avons effectués.

	Rappel	Précision
Montagne	0,538	0,5295
Verdure foret	0,72	0,695
Mer	0,624	0,6753
Ville	0,79	0,7932
Coucher de soleil	0,826	0,7749
Plage désert	0,678	0,7048
Taux d'erreur	**0,30**	

(a)

	Rappel	Précision
Montagne	0,532	0,5385
Verdure foret	0,744	0,7209
Mer	0,616	0,6652
Ville	0,802	0,7609
Coucher de soleil	0,822	0,7555
Plage désert	0,644	0,7061
Taux d'erreur	**0,31**	

(b)

	Rappel	Précision
Montagne	0,552	0,5811
Verdure foret	0,734	0,7355
Mer	0,626	0,6674
Ville	0,792	0,7689
Coucher de soleil	0,83	0,795
Plage désert	0,698	0,6712
Taux d'erreur	**0,29**	

(c)

	Rappel	Précision
Montagne	0,4952	0,5
Verdure foret	0,8	0,6391
Mer	0,6471	0,5093
Ville	0,4429	0,5345
Coucher de soleil	0,4381	0,5349
Plage désert	0,2818	0,3647
Taux d'erreur	**0,47**	

(d)

	Rappel	Précision
Montagne	0,486	0,5084
Verdure foret	0,756	0,7146
Mer	0,556	0,6123
Ville	0,69	0,666
Coucher de soleil	0,796	0,7524
Plage désert	0,682	0,6931
Taux d'erreur	**0,34**	

(e)

	Rappel	Précision
Montagne	0,556	0,5594
Verdure foret	0,748	0,7083
Mer	0,628	0,6528
Ville	0,804	0,7643
Coucher de soleil	0,744	0,7932
Plage désert	0,67	0,6713
Taux d'erreur	**0,31**	

(f)

Tableau 9: Résultats des expérimentations sur les caractéristiques de cooccurrence
(a) Matrice de cooccurrence brute, sans normalisation, recherche des segments "en bas à droite"
(b) Matrice de cooccurrence , recherche des segments "dans un rayon r"
(c) Matrice de cooccurrence avec normalisation en longueur, recherche des segments "dans un rayon r"
(d) Matrice de cooccurrence avec normalisation en orientation, recherche des segments "dans un rayon r"
(e) Matrice de cooccurrence avec 4 cellules d'angles, recherche des segments "dans un rayon r"
(f) Matrice de cooccurrence avec 7 cellules d'angles, recherche des segments "dans un rayon r"

L'effet de la normalisation en longueur ne semble pas lui non plus significatif comme le montrent les résultats en (c). En revanche, la normalisation en orientation évaluée en (d) semble avoir un impact négatif significatif : si on compare les résultats par rapport à l'expérience de référence, on constate que si les effets du descripteur sont positifs sur les classes faisant intervenir des segments (ville, foret/verdure et montagne) mais que les effets sont très négatifs sur les classes "coucher de soleil" et "plage/désert" : il semblerait que ce descripteur provoque donc un bruit conséquent. Plus généralement on peut douter des bénéfices de l'invariance en rotation que ce soit au vu de ces résultats ou d'une manière intuitive (la position verticale d'un objet est en elle-même une information), mais également au vu des expériences de [70] qui montrent que dans les problèmes de classification, l'algorithme RIFT [72], (version invariante en rotation de SIFT [21], voir chapitre 3) n'apporte pas d'information supplémentaire par rapport à SIFT.

Les résultats présentés en (e) et (f) nous permettent enfin de déterminer un nombre de cellules optimal en tant que compromis entre la taille du vecteur de caractéristiques et les performances de classification : ils montrent que le nombre de cellules le plus efficace est 5, le taux d'erreur commençant à croitre à partir de 4 et n'évoluant plus pour 6 ou 7 cellules.

En terme de classification par utilisation globale de l'image, les meilleurs résultats sont obtenus par les méthodes (a) (b) et (c). Notre classificateur le plus efficace devient donc le classificateur (c) dont les résultats sont présentés dans le Tableau 10. Notons encore une fois que ces résultats sont obtenus à partir d'un vecteur de dimension assez faible (9 dimensions pour la couleur et 15 pour la matrice de cooccurrence)

	Montagne	**Verdure foret**	**Mer**	**Ville**	**Coucher de soleil**	**Plage désert**	**Total**
Montagne	276	63	81	6	19	55	500
Verdure foret	50	367	10	50	6	17	500
Mer	72	13	313	17	21	64	500
Ville	21	33	17	396	26	7	500
Coucher de soleil	8	8	17	24	415	28	500
Plage désert	48	15	31	22	35	349	500
Total	475	499	469	515	522	520	3000
Taux d'erreur	**0,29**						

Tableau 10: Classification globale d'images au moyen d'une matrice de cooccurrence de segments (vérité terrain en première colonne)

On retrouve un fort taux d'erreurs entre la classe montagne et les mêmes classes (verdure forêt et mer) que montré précédemment. Les classes mer et plage-désert sont assez naturellement confondues, ce qui est d'ailleurs plutôt compréhensible étant donnée la proximité sémantique de ces deux catégories. La confusion plage-désert/montagnes peut être à la fois attribuée à la présence d'image mixtes et à de nombreux points communs sur les structures des photos (grandes zones de ciel bleu, structures vallonnées qui se découpent, etc.).

A partir de là, on peut, à priori, penser qu'une une meilleure différenciation sur les catégories les moins "performantes" pourrait provenir de deux améliorations : l'intégration de caractéristiques de texture (utiles pour une meilleure identification de la classe montagne) et

l'utilisation de caractéristiques locales qui nous permettraient en effet de juger plus finement les éléments présents dans l'image et en particulier leurs proportions respectives.

Si la taille du vecteur de caractéristiques par rapport au nombre d'images d'entrainement disponibles dans notre base ne nous permet pas d'intégrer des informations de texture, nous pouvons en revanche explorer la piste de la classification locale et éventuellement la fusion des informations produites avec des informations globales.

4.3. Approche « locale » de catégorisation d'images

Ayant évalué les performances d'une approche « globale » utilisant les descripteurs extraits directement d'une image sans segmentation, nous choisissons maintenant de traiter le problème avec notre approche orienté régions évoquée tout au long de cette thèse. Si l'approche "vocabulaire visuel" a été largement utilisée dans la littérature pour la détection d'objets visuels, quel serait son comportement pour le problème de catégorisation globale d'images en concepts car a priori les composantes d'une image peuvent autant perturber l'interprétation globale que l'assister.

Le procédé de classification est donc le suivant : sur une image segmentée en région, on extrait pour chaque région les mêmes caractéristiques que pour la classification globale (matrice de coocurrence, moments de couleur) avec en plus des descripteurs de forme de régions (moments de Hu). Le vocabulaire est, quant à lui, déterminé comme indiqué dans la section 3.3 de ce chapitre. Les figures 43 à 45 représentent les courbes de MSE obtenues pour les vocabulaires :

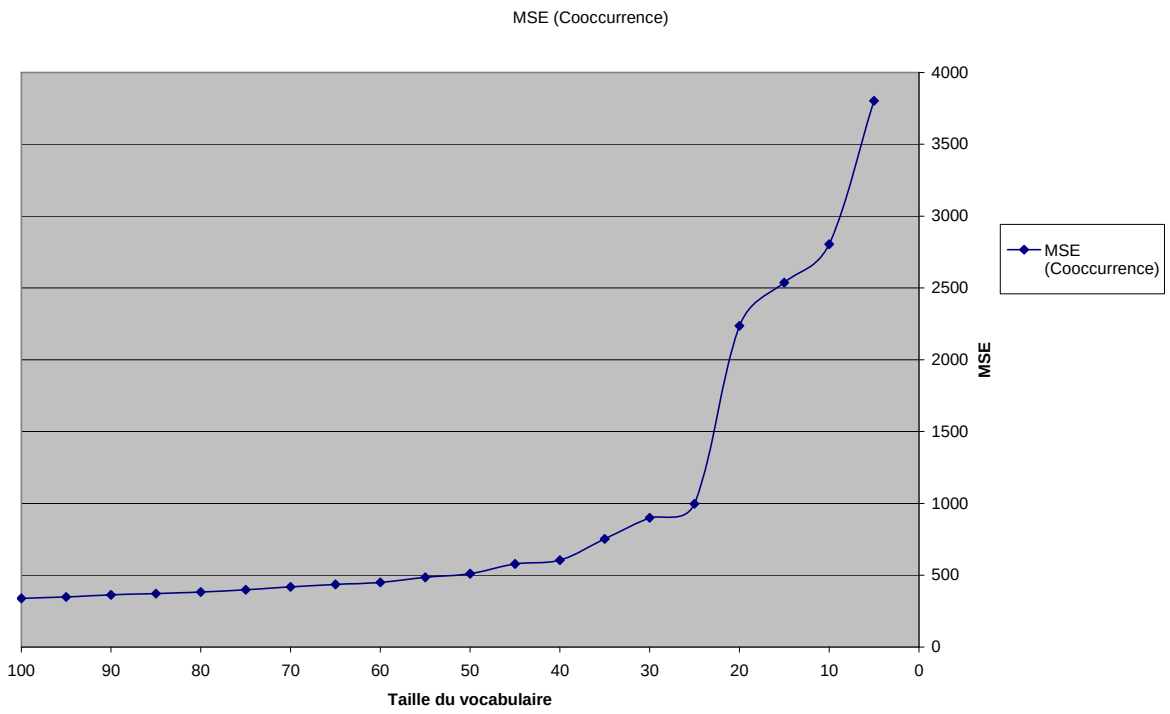


Figure 43: Evolution de la MSE pour la quantification des informations de cooccurrence (normalisées en longueur et non en angle – meilleurs résultats)

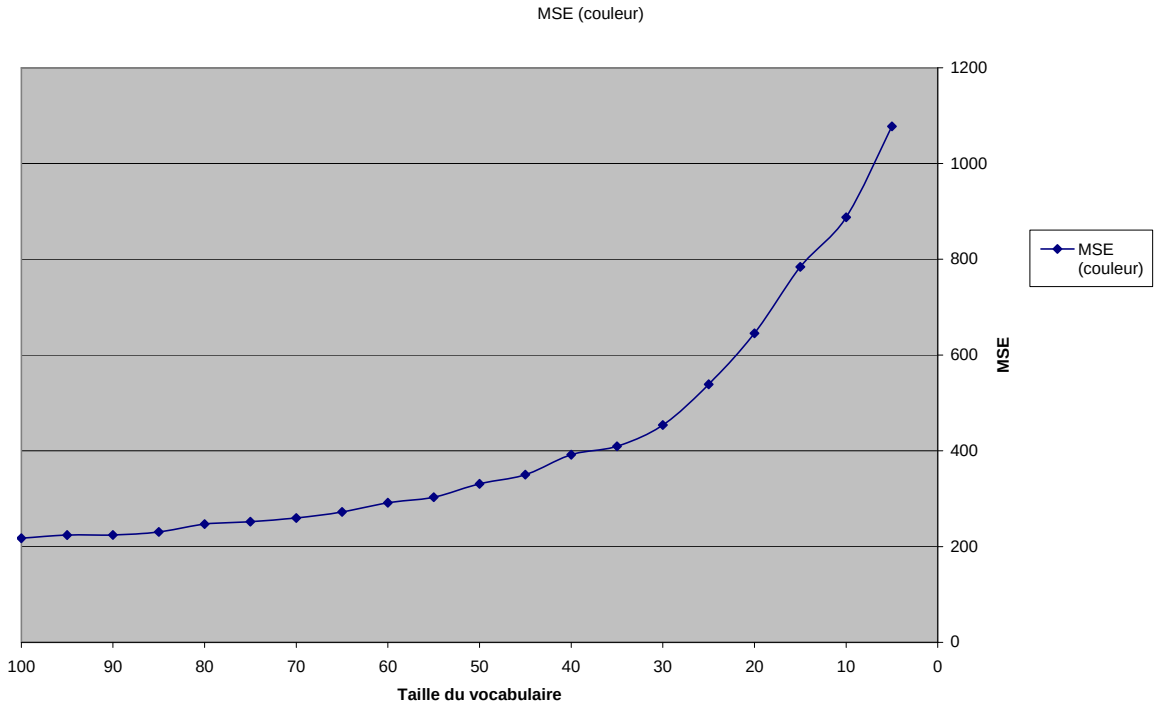


Figure 44: Evolution de la MSE pour la quantification des informations de couleur

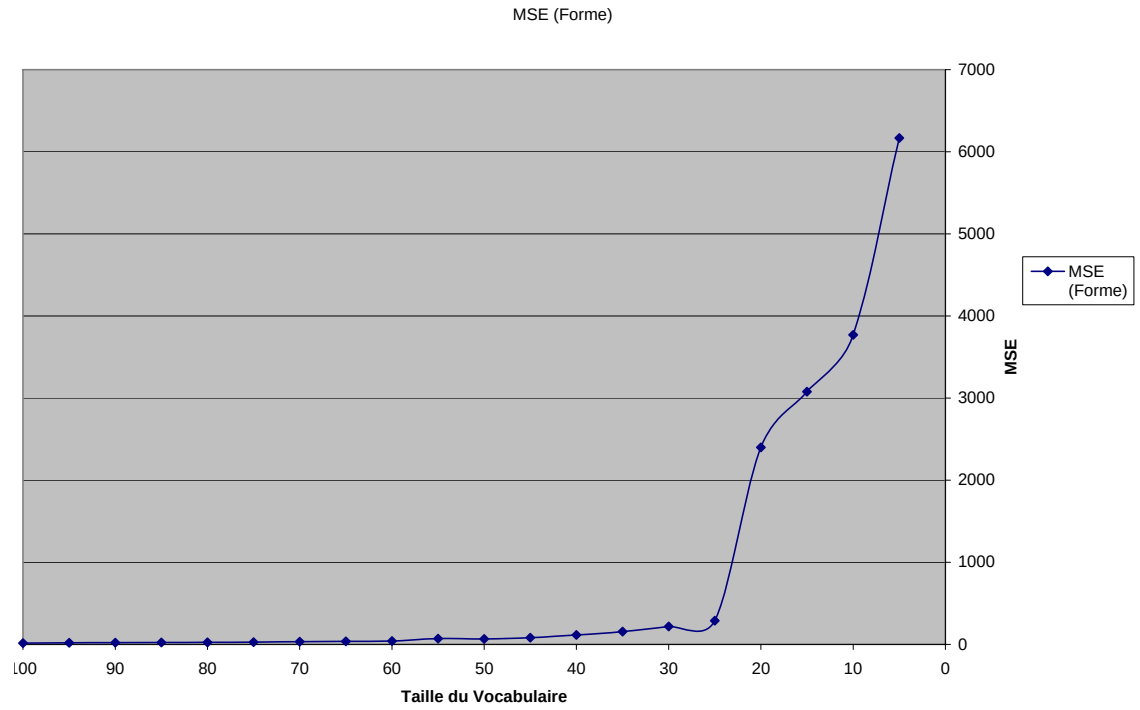


Figure 45: Evolution de la MSE pour la quantification des informations de Forme (moments de HU)

Ces courbes nous permettent de choisir des vocabulaires de tailles adaptées. On considère des vocabulaires de tailles respectives 25, 20 et 25. A titre de comparaison nous donnerons les résultats obtenus avec des vocabulaires de tailles 40, 30 et 30. Les résultats obtenus en fusion intermédiaire sont résumés dans le Tableau 11 et le Tableau 12. Les valeurs des descripteurs sont, comme toujours, normalisées (centrées et réduites). A noter qu'il a été nécessaire de réapprendre le vocabulaire pour chaque étape de chaque validation croisée afin de ne pas biaiser les résultats.

	Rappel	Précision
Montagne	0,362	0,3679
Verdure foret	0,468	0,4766
Mer	0,57	0,5588
Ville	0,594	0,5332
Coucher de soleil	0,562	0,5916
Plage désert	0,34	0,3579
Taux d'erreur	**0,5173**	

	Montagne	Verdure foret	Mer	Ville	Coucher de soleil	Plage désert	Total
Montagne	181	71	121	58	21	48	500
Verdure foret	82	234	15	83	22	64	500
Mer	105	28	285	23	21	38	500
Ville	59	86	18	297	8	32	500
Coucher de soleil	21	14	24	37	281	123	500
Plage désert	44	58	47	59	122	170	500
Total	492	491	510	557	475	475	3000

Tableau 11: Classification globale par vocabulaire visuel (vocabulaire réduit, vérité terrain en première colonne)

	Rappel	Précision
Montagne	0,368	0,3907
Verdure foret	0,478	0,4705
Mer	0,59	0,5684
Ville	0,534	0,4846
Coucher de soleil	0,6	0,6085
Plage désert	0,322	0,3515
Taux d'erreur	**0,518**	

	Montagne	Verdure foret	Mer	Ville	Coucher de soleil	Plage désert	Total
Montagne	184	92	103	73	9	39	500
Verdure foret	73	239	12	74	27	75	500
Mer	87	34	295	25	26	33	500
Ville	80	77	17	267	12	47	500
Coucher de soleil	13	13	38	33	300	103	500
Plage désert	34	53	54	79	119	161	500
Total	471	508	519	551	493	458	3000

Tableau 12: Classification globale par vocabulaire visuel (vocabulaire étendu, vérité terrain en première colonne)

Notre première conclusion est qu'il semble que cette méthode de classification n'est pas adaptée à l'interprétation globale d'image. En effet là où les caractéristiques globales vont déterminer une plage plus par sa couleur, les caractéristiques locales vont chercher à caractériser l'image par sa composition, ce qui est ici inadapté. La matrice de confusion est particulièrement instructive à cet égard : la classe montagne présente en fait de nombreux points communs avec la classe mer (présences de roches, étendues bleues, etc.…), on a également une forte quantité d'images de la catégorie "coucher de soleil" qui comprennent des éléments de plage ce qui implique d'assez fortes confusions. D'une manière générale, l'hétérogénéité du contenu pose problème et la transposition directe d'un algorithme plutôt pensé pour la classification d'objets visuels n'est pas adaptée

On notera également que le second vocabulaire a présenté une très forte tendance à l'overfitting ce qui traduit une dimension excessive par rapport à la quantité d'exemples présents. D'une manière plus générale, les deux types de caractérisation souffrent de la trop grande dimension de leurs vecteurs de caractéristiques par rapport au nombre d'exemples disponibles. Ainsi les performances des algorithmes traditionnellement les plus efficaces (Réseaux de neurones, SVM) deviennent-elles inférieures à des algorithmes plus simples comme les arbres de décision. Les résultats décrits plus hauts ont été obtenus avec l'utilisation d'arbres de décision C4.5 combinés par Adaboost. L'utilisation d'un algorithme de sélection pourrait ici s'avérer intéressante.

Enfin on remarquera que les performances avec les deux vocabulaires sont assez similaires ; on se gardera néanmoins de conclure quant à la performance de notre mode de

sélection de la taille du vocabulaire car le faible nombre d'exemples limite les capacités d'apprentissage.

4.4. Discussion

Cette partie sur la classification globale nous a permis d'une part de valider nos descripteurs issus de segments et d'autre part de donner une indication sur la complémentarité entre les caractéristiques globales et les caractéristiques locales. Il serait intéressant de compléter ceci par l'étude de résultats locaux et globaux fusionnés. L'expérience basée sur le vocabulaire visuel montre toutefois une limite de notre mécanisme d'expression des relations entre régions.

Concernant les procédés de classification eux-mêmes, on notera qu'étant donné la taille réduite de la base, une sélection des caractéristiques (par exemple par ACP) serait à envisager, tout particulièrement pour la classification basée sur des régions. Notre stratégie de fusion intermédiaire permet déjà de limiter la taille du vecteur de caractéristiques. Sa pertinence sera étudiée plus en détail dans la section suivante.

5. Catégorisation d'objets visuels

Dans cette section nous allons aborder la deuxième facette du problème de classification sémantique qui concerne la détection d'objets visuels dans le cadre du challenge pascal VOC 2007 [3].

5.1. La base Pascal VOC 2007

La base Pascal VOC 2007 [3] est un ensemble de 9963 images destinées à poser un problème de classification difficile afin d'évaluer les performances des algorithmes de classification d'objets visuels (VOC). Cette base est décomposée en 20 catégories non-mutuellement exclusives qui sont les suivantes :

1. Avion
2. Vélo
3. Oiseau
4. Bateau
5. Bouteille
6. Bus
7. Voiture
8. Chat
9. Chaise
10. Vache
11. Table
12. Chien
13. Cheval
14. Moto
15. Personne
16. Plante en pot
17. Mouton
18. Sofa
19. Train
20. Télévision

Sur le plan de l'évaluation elle-même la base est décomposée en 5011 images d'entrainement et de validation ainsi que 4952 images de test. Si les catégories sont représentées équitablement entre ces deux sous-ensembles, elles ne sont pas, en revanche, équitablement réparties dans la base. De plus on notera qu'évaluer la reconnaissance d'une catégorie sur toute la base biaise le résultat en faveur d'une classification négative (les images négatives étant largement plus représentées que les autres).

5.2. Protocole expérimental

Il s'agit donc mettre de en application notre paradigme de classification, décrit dans la section 3, inspiré des principes d'agglomération des lois Gestalt et basé sur la décomposition de l'image en régions. Néanmoins, la mise en application de notre procédé de classification

nécessite encore de préciser plusieurs paramètres, à commencer par la structure de classificateurs ainsi que les descripteurs utilisés pour la construction de vocabulaire visuel et la modélisation du contenu visuel d'images.

5.2.1. Architecture des classificateurs

En ce qui concerne la structure de classificateurs, notre solution s'articule autour d'un ensemble de classificateurs binaires qui sont individuellement responsables d'une classe d'objets. Cette structure nous est imposée par la possibilité de voir cohabiter deux classes ou plus au sein d'une même image. L'hétérogénéité du contenu d'une même image, la variété de la base et des situations qu'elle présente suggèrent fortement une approche locale.

Lors de cette première approche nous utiliserons des SVM pour la classification. Là encore, après quelques essais, le faible nombre d'exemples disponibles pour chaque classe ne nous a pas permis d'avoir de grosses variations de performances en variant le type de noyaux ; certains étant pourtant réputés plus performants (par exemple des noyaux basés sur des RBF – fonctions à base radiale – ou polynomiales de haute dimension). Les meilleurs résultats ont donc été obtenus avec un noyau basé sur les RBF mais ceux-ci étaient très voisins (surtout pour les classes faiblement représentées) à ceux obtenus avec un noyau linéaire.

5.2.2. Caractéristiques d'image

Dans la section précédente nous avons établi la pertinence de certaines caractéristiques pour la tâche d'interprétation globale de l'image mais il est délicat de transposer ces résultats sur le problème de classification présenté par le challenge Pascal tant ils sont de nature différente (nous avons d'ailleurs pu constater que l'approche basée régions n'avait pas beaucoup de succès). On va donc tout simplement extraire un ensemble d'informations de l'image et on travaillera sur leur éventuelle combinaison ultérieure en utilisant notre mécanisme de vocabulaires visuels :

- Caractéristiques SIFT (nous utilisons l'implémentation de S. Nowozin [155])
- Moments de couleur d'une région
- Histogramme de segments d'une région (histogramme 3D : population en fonction d'une longueur normalisée et d'une orientation normalisée)
- Matrice de cooccurrence de segments d'une région (voir chapitre 5)
- Matrice de cooccurrence de segments invariante en rotation d'une région (voir chapitre 5)
- Moments de Hu d'une région

Les caractéristiques SIFT ont été choisies parce qu'elles ont été à la base des classificateurs ayant produit la meilleure performance au challenge Pascal et de ce fait très populaires dans la littérature. Elles reflètent des caractéristiques extraites des points d'intérêt censées à être invariantes au changement d'échelle. Ensuite, nous introduisons des descripteurs permettant de caractériser les régions d'une image segmentée dans la lignée de l'inspiration perceptuelle : les moments de couleur d'une région, histogramme de segments d'une région, etc.

Nous avons évoqué également lors de l'état de l'art la possibilité de fusionner les Gestalts partiels représentés par les régions. Le problème étant que les modes de fusion possibles sont trop complexes pour être modélisés (on ne sait pas vraiment s'il faut fusionner

par continuité de texture, de forme, etc.) on prendra donc en compte différents niveaux de fusion pour chaque région, ainsi le vecteur de caractéristiques représentant une région sera constitué d'une partie contenant les caractéristiques de la région elle-même, puis d'une seconde partie contenant les caractéristiques de la région mélangée à ses voisines, et ainsi de suite, sur trois niveaux. Ceci nous permet également dans une certaine mesure de modéliser des changements d'échelle avec la fusion d'une région dans son environnement. Les caractéristiques sont extraites sur tous les pixels constituant la région et les segments sont comptés complètement s'ils sont présents (même partiellement) dans une région.

On notera que les caractéristiques SIFT ne peuvent pas se combiner avec les autres en utilisant un mécanisme de fusion précoce et qu'ils seront donc testés et utilisés uniquement en fusion intermédiaire (leur mode d'extraction est différent).

5.2.3. Conditions expérimentales

La base Pascal VOC 2007 étant déséquilibrée entre les exemples positifs et les négatifs pour chacune des catégories, on s'attachera donc pour nos évaluations plutôt à équilibrer les images positives et négatives et à répéter les expériences sur des sous ensembles différents afin d'une part d'évaluer précisément la capacité discriminante de notre classificateur et d'autre part de prendre en compte la diversité des images négatives.

Nos résultats sont obtenus avec le jeu d'entrainement de la base pascal et testés sur le jeu de test. Il en résulte que les jeux d'entrainement et de test sont plus fournis et de taille égale. Les résultats présentés sont la moyenne sur des séries de 5 expériences, chacune suivant le protocole suivant : on entraine le classificateur sur toutes les données positives d'entrainement combinées à un nombre égal de données négatives tirées aléatoirement mais en faisant en sorte que les 5 expériences présentent des jeux distincts, les données de test sont tirées aléatoirement en équilibrant les données positives et négatives. Ceci évite un biais d'apprentissage important qui serait obtenu en entrainant des classes peu représentées (moins de 200 occurrences dans la base) : on obtiendrait des algorithmes favorisant la non-détection de classe, ce cas étant largement majoritaire.

5.3. Résultats expérimentaux

D'après notre démarche de classification décrite dans la section 3, nous procédons d'abord à la constitution d'un vocabulaire par type de caractéristiques en étudiant l'évolution des courbes MSE. Ensuite, nous allons étudier et comparer la différence et la complémentarité de nos descripteurs sur quelques catégories typiques de la base Pascal VOC 2007. Enfin, nous présentons nos résultats finaux sur l'ensemble de la base Pascal VOC 2007.

5.3.1. Constitution des Vocabulaires

Nous débutons là encore par une étape d'étude puis de génération des vocabulaires pour chacun de nos descripteurs. Les figures 46 à 51 illustrent les courbes de MSE pour chacune des caractéristiques. Ces courbes nous ont servi de base pour déterminer la taille de nos vocabulaires et ont été établies sur des moyennes effectués sur toutes les images de la base d'entrainement.

A partir de ces résultats les tailles de vocabulaire choisies sont les suivantes :

- Caractéristiques SIFT : 35
- Moments de couleur : 25
- Histogramme de segments (invariant en rotation) : 20
- Matrice de cooccurrence de segments : 25
- Matrice de cooccurrence de segments invariante en rotation : 25
- Moments de Hu : 25

Ces tailles correspondent à des positions estimées de changement prononcé de la courbure de la courbe de MSE.

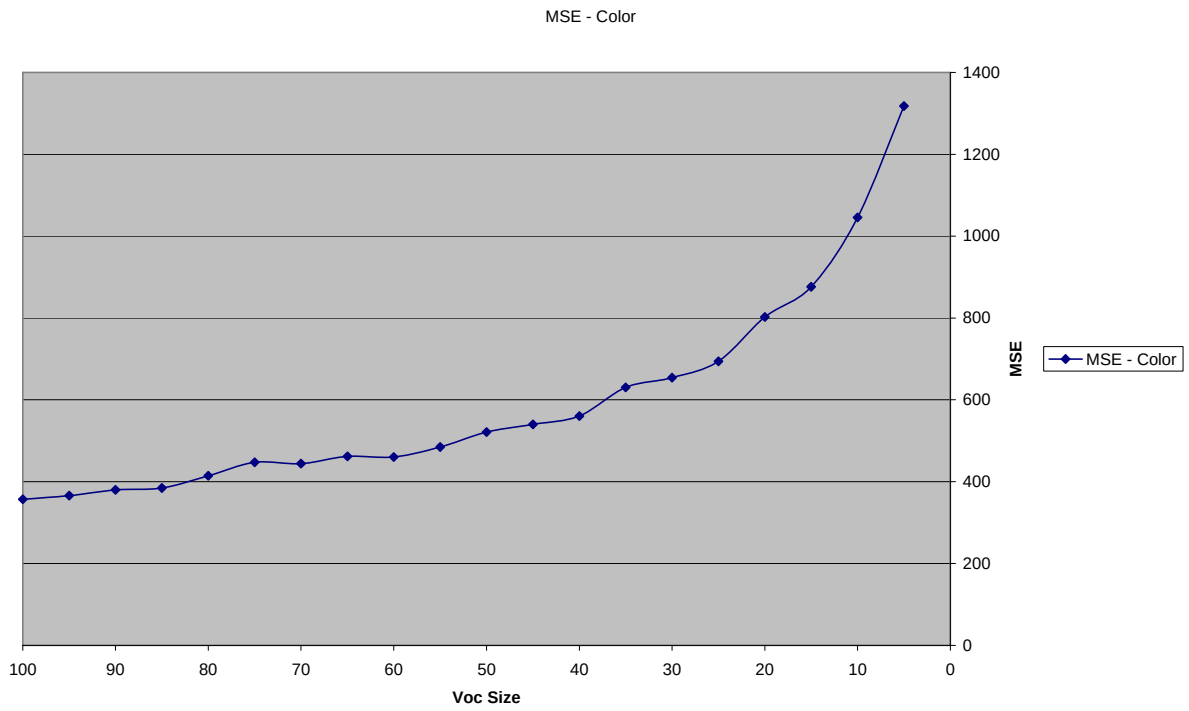


Figure 46: Evolution de la MSE ; Couleur (Moments)

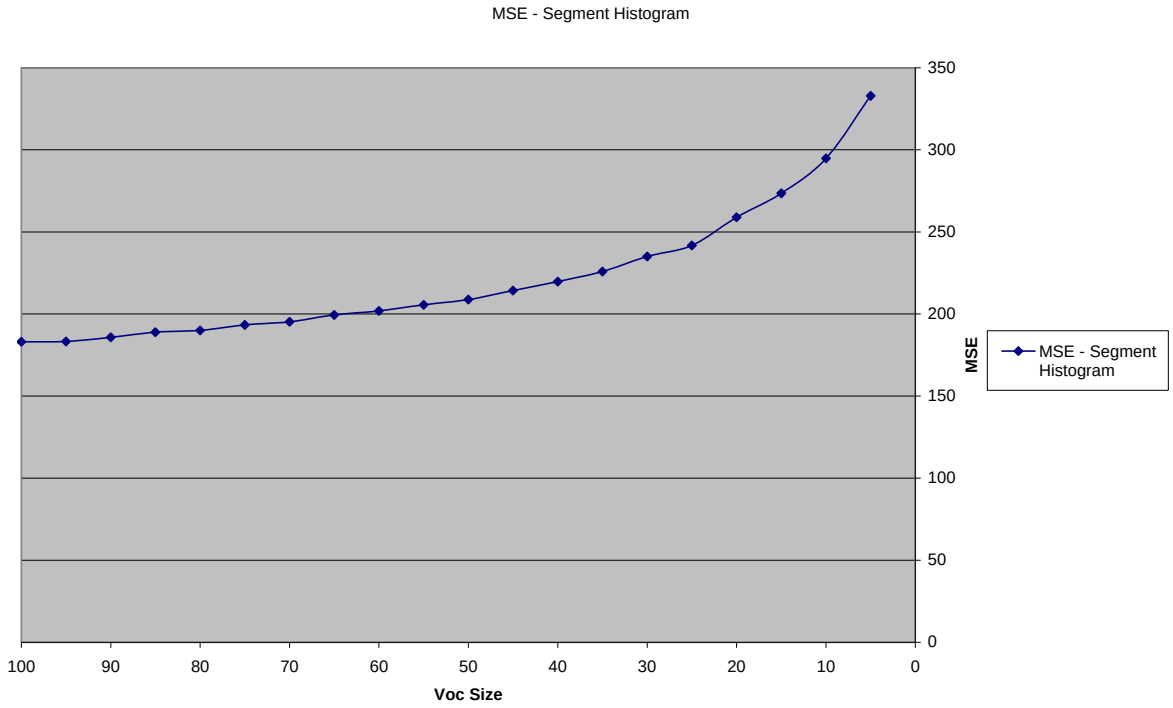


Figure 47: Evolution de la MSE ; Histogramme de Segments

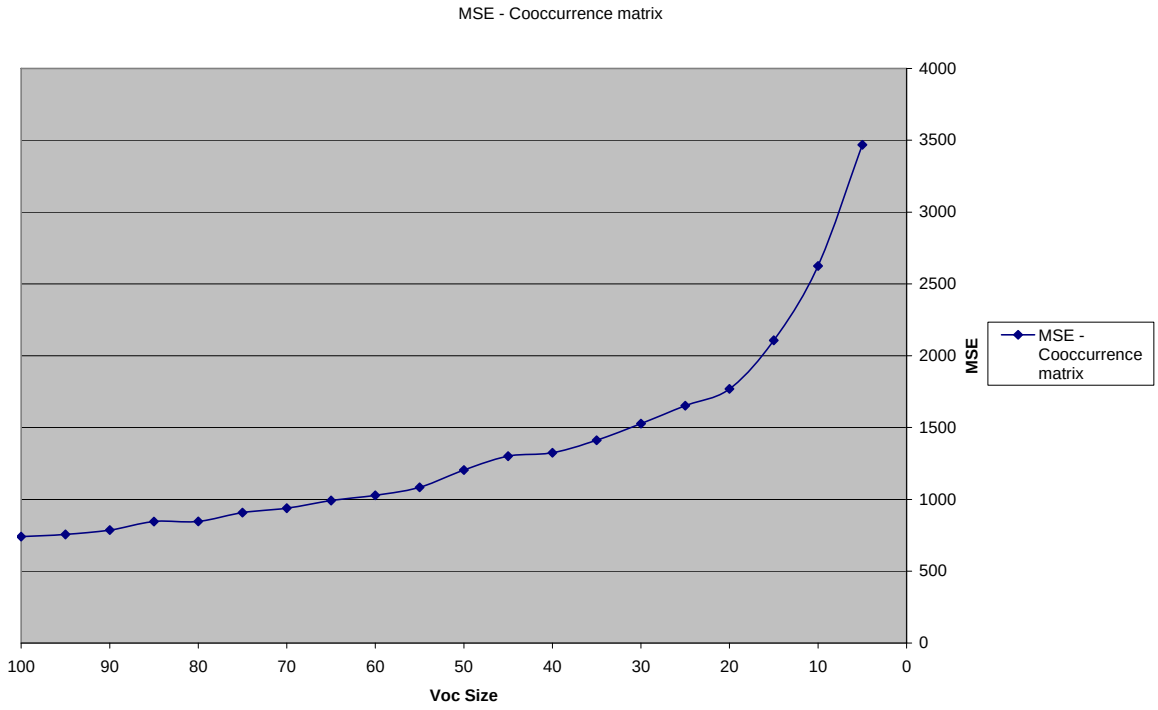


Figure 48: Evolution de la MSE ; Matrice de cooccurrence de Segments

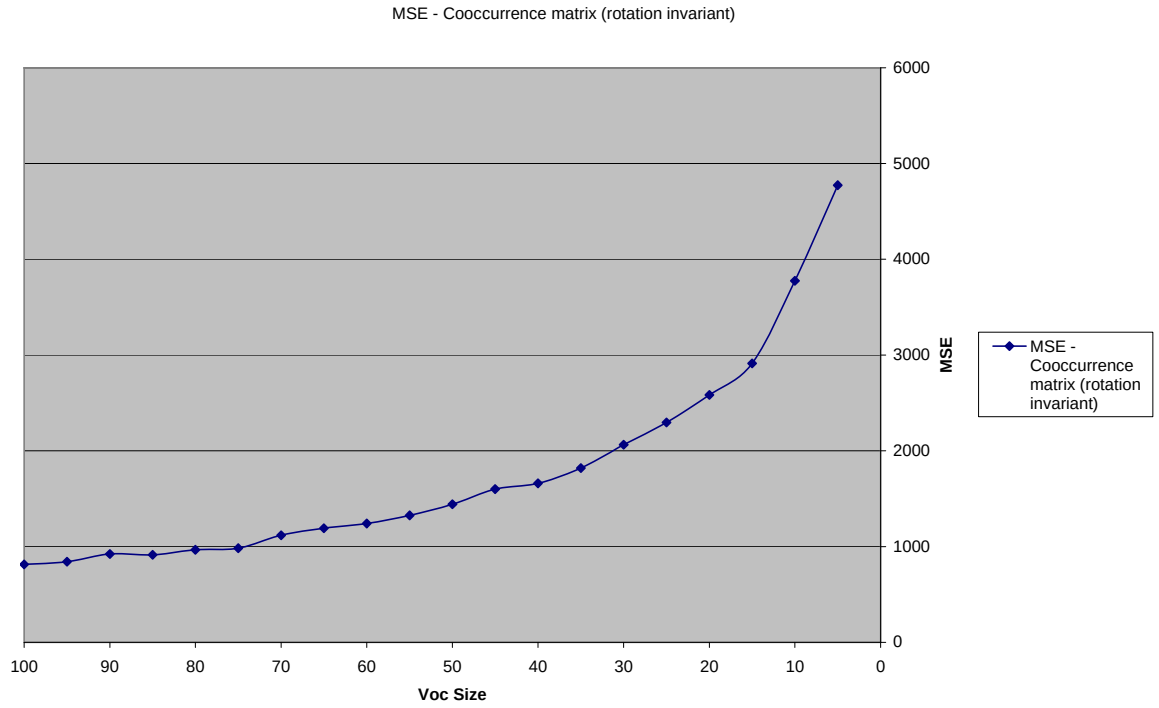


Figure 49: Evolution de la MSE ; Matrice de cooccurrence de Segments (invariance en rotation)

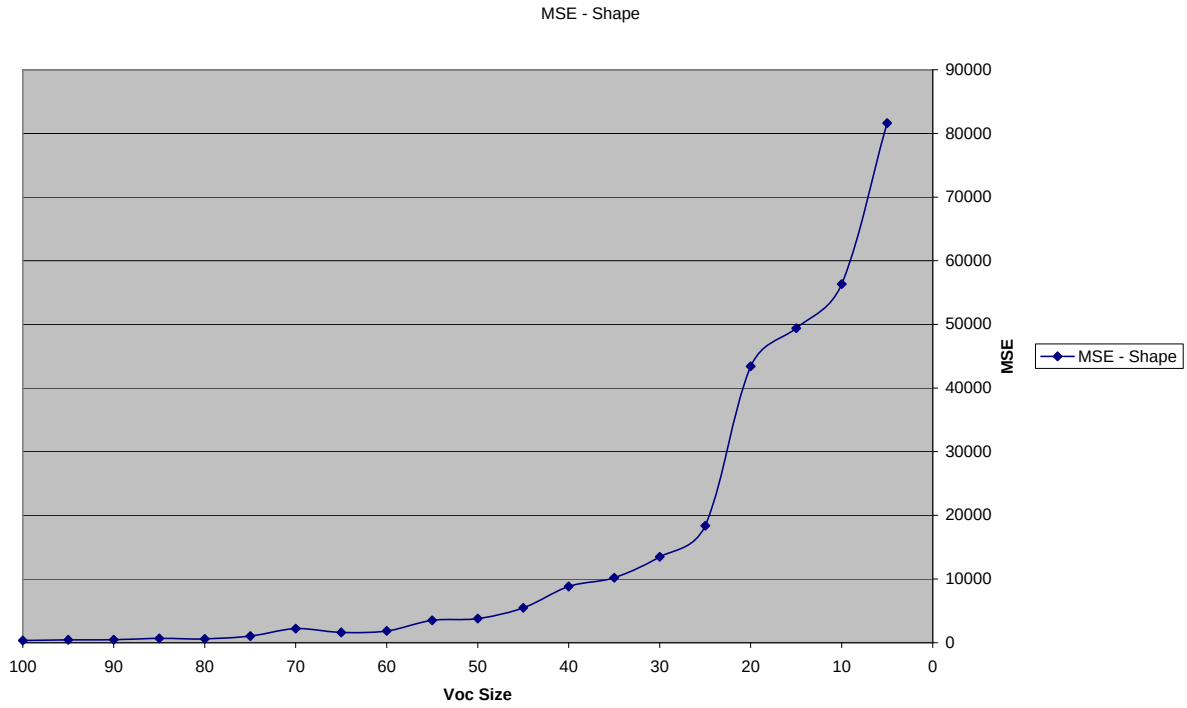


Figure 50: Evolution de la MSE ; Descripteurs de forme (moments de HU)

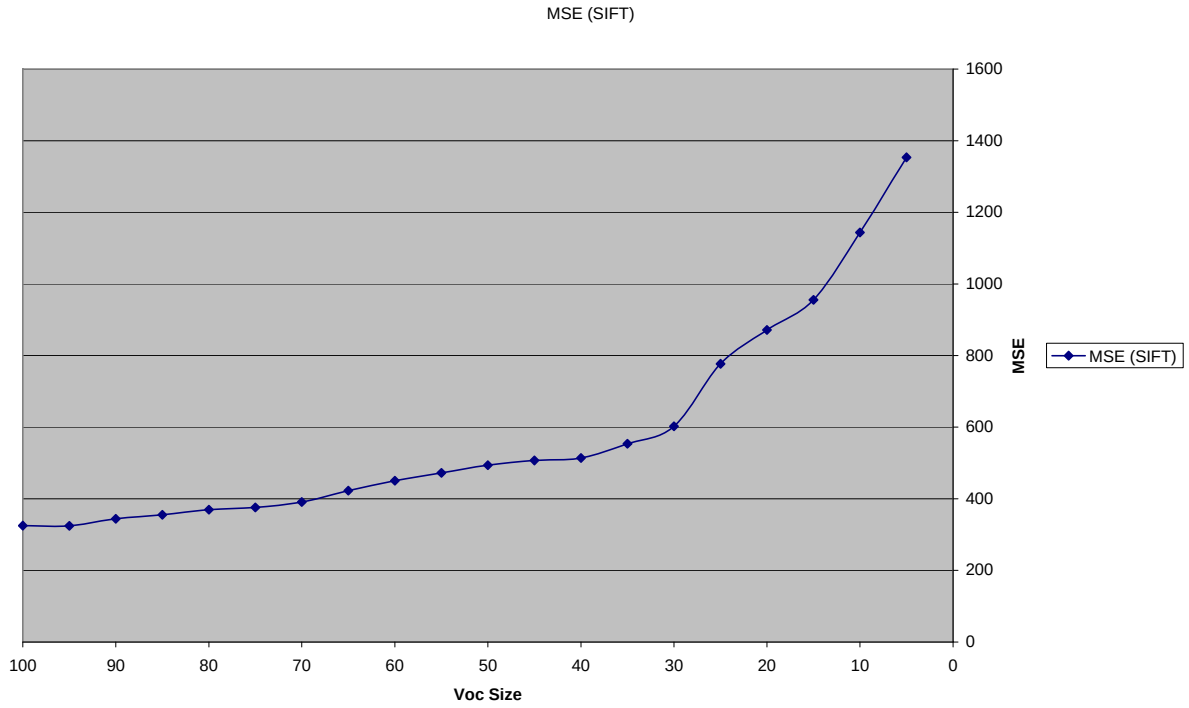


Figure 51: Evolution de la MSE ; SIFT

Une fois le vocabulaire visuel choisi pour chaque type de descripteur, une image sera alors modélisée par une concaténation des histogrammes issus de ces vocabulaires visuels par descripteur comme décrit dans la section 3.2.

5.3.2. Comparaison des différentes caractéristiques

Dans un premier temps, nous proposons d'étudier les caractéristiques comparativement sur quelques classes d'objet représentatives en vue de mettre en lumière les comportements de chaque type de descripteur ainsi de leur combinaison. Par cette expérimentation, nous tâcherons de répondre aux questions suivantes :

1) Comment nos descripteurs de région basés sur les segments se comportent-ils par rapport au descripteur populaire SIFT dans la discrimination des classes ?
2) Nos descripteurs de région basés sur les segments sont-ils complémentaires au descripteur SIFT ?
3) La forme des régions dans une image segmentée participe-t-elle dans la discrimination des objets visuels ?

Les classes représentatives choisies de bal ase VOC Pascal 2007 sont les suivantes :

- Classe présentant une structure non déformable : Bus (360 images positives + 360 images négatives)
- Classe présentant une structure non déformable de petite taille : Bouteille (456 images positives + 456 images négatives)
- Classe "animal" : Chat (659 images positives + 659 images négatives)
- Classe personne (particulièrement représentée :)

a) Descripteurs basés sur les segments et SIFT

Rappelons que nos descripteurs de segment sont calculés pour chacune des régions dans une image segmentée et permettent de capturer à la fois une information de forme géométrique et une information de texture locale. Comment se comportent-ils par rapport au descripteur SIFT dans la discrimination des classes ? Nous présentons donc les performances de chacun de nos descripteurs de forme (ainsi que SIFT) associés au descripteur de couleur.

Comme décrit précédemment on utilise des SVM avec un noyau basé sur des RBF et chaque caractéristique et normalisée en moyenne et en variance.

	Rappel	Précision
NO-BUS	0,6379	0,7115
BUS	0,7414	0,6719
Error rate	0,3103	

(a) complet

	Rappel	Précision
NO-BUS	0,523	0,6454
BUS	0,7126	0,599
Error rate	0,3822	

(b) Matrice de coocurrence (Cooc)

	Rappel	Précision
NO-BUS	0,6609	0,7143
BUS	0,7356	0,6845
Error rate	0,3017	

(c) Histogramme de segments (Hist)

	Rappel	Précision
NO-BUS	0,4943	0,6825
BUS	0,7701	0,6036
Error rate	0,3678	

(d) Cooc invariante en rotation (CoocI)

	Rappel	Précision
NO-BUS	0,5	0,6444
BUS	0,7241	0,5915
Error rate	0,3879	

(e) SIFT

Tableau 13: Descripteurs de forme sur la classe BUS

	Rappel	Précision
BOTTLE	0,7594	0,6289
NO-BOTTLE	0,5519	0,6964
Error rate	0,3443	

(a) complet

	Rappel	Précision
BOTTLE	0,7358	0,6265
NO-BOTTLE	0,5613	0,68
Error rate	0,3514	

(b) Matrice de coocurrence (Cooc)

	Rappel	Précision
BOTTLE	0,7972	0,6426
NO-BOTTLE	0,5566	0,7329
Error rate	0,3231	

(c) Histogramme de segments (Hist)

	Rappel	Précision
BOTTLE	0,6887	0,6432
NO-BOTTLE	0,6179	0,665
Error rate	0,3467	

(d) Cooc invariante en rotation (CoocI)

	Rappel	Précision
BOTTLE	0,6792	0,64
NO-BOTTLE	0,6179	0,6583
Error rate	0,3514	

(e) SIFT

Tableau 14: Descripteurs de forme sur la classe bouteille

	Rappel	Précision
CAT	0,6335	0,6915
NO-CAT	0,7174	0,6619
Error rate	0,3245	

(a) complet

	Rappel	Précision
CAT	0,6739	0,5726
NO-CAT	0,4969	0,6038
Error rate	0,4146	

(b) Matrice de coocurrence (Cooc)

	Rappel	Précision
CAT	0,6708	0,6526
NO-CAT	0,6429	0,6613
Error rate	0,3432	

(c) Histogramme de segments (Hist)

	Rappel	Précision
CAT	0,6553	0,5877
NO-CAT	0,5404	0,6105
Error rate	0,4022	

(d) Cooc invariante en rotation (CoocI)

	Rappel	Précision
CAT	0,5311	0,5797
NO-CAT	0,6149	0,5673
Error rate	0,427	

(e) SIFT

Tableau 15: Descripteurs de forme sur la classe Chat

	Rappel	Précision
NO-PERSON	0,5207	0,6194
PERSON	0,6801	0,5866
Error rate	0,3996	

(a) complet

	Rappel	Précision
NO-PERSON	0,5037	0,5954
PERSON	0,6577	0,5699
Error rate	0,4193	

(b) Matrice de coocurrence (Cooc)

	Rappel	Précision
NO-PERSON	0,6213	0,6379
PERSON	0,6472	0,6309
Error rate	0,3657	

(c) Histogramme de segments (Hist)

	Rappel	Précision
NO-PERSON	0,6104	0,5669
PERSON	0,5336	0,578
Error rate	0,428	

(d) Cooc invariante en rotation (CoocI)

	Rappel	Précision
NO-PERSON	0,5112	0,57
PERSON	0,6143	0,5569
Error rate	0,4372	

(e) SIFT

Tableau 16: Descripteurs de forme sur la classe personne

Nous voyons que si la matrice de coocurrence était le descripteur le plus efficace pour les images globales, c'est ici l'histogramme 3D de segments qui offre les meilleures performances, parfois même meilleures que celles du classificateur complet. Ceci est vraisemblablement dû aux dimensions trop importantes du vecteur de caractéristiques par rapport au peu d'informations apportées par certaines composantes.

Globalement nos descripteurs semblent apporter plus d'informations que SIFT. On notera aussi que l'apport de l'invariance sur les matrices de cooccurrence est bénéfique ou non selon la classe. Les performances des deux descripteurs restent toutefois similaires.

b) Complémentarité par rapport au SIFT ?

L'expérience suivante est d'évaluer la complémentarité des différents descripteurs. Nous proposons donc d'étudier leurs performances en les combinant deux à deux. Une amélioration notable de performance dans la classification est une indication claire d'apport d'information mutuelle pour la discrimination de classes. Les tableaux suivants donnent les performances de classification des diverses combinaisons possibles (nous utilisons les abréviations de l'expérience a)):

	Rappel	Précision
NO-BUS	0,5632	0,6282
BUS	0,6667	0,6042
Error rate	0,3851	

(a) Couleur + Cooc + CoocI

	Rappel	Précision
NO-BUS	0,5517	0,6316
BUS	0,6782	0,602
Error rate	0,3851	

(b) Couleur + Cooc + SIFT

	Rappel	Précision
NO-BUS	0,7644	0,652
BUS	0,592	0,7153
Error rate	0,3218	

(c) Couleur + CoocI + SIFT

	Rappel	Précision
NO-BUS	0,6897	0,7547
BUS	0,7759	0,7143
Error rate	0,2672	

(d) Couleur + Cooc + Hist

	Rappel	Précision
NO-BUS	0,6207	0,6923
BUS	0,7241	0,6562
Error rate	0,3276	

(e) Couleur + Hist + CoocI

	Rappel	Précision
NO-BUS	0,7069	0,6758
BUS	0,6609	0,6928
Error rate	0,3161	

(f) Couleur + Hist + SIFT

Tableau 17: Combinaisons de descripteurs : résultats sur la classe bus

	Rappel	Précision
BOTTLE	0,6887	0,6404
NO-BOTTLE	0,6132	0,6633
Error rate	0,3491	

(a) Couleur + Cooc + CoocI

	Rappel	Précision
BOTTLE	0,684	0,6444
NO-BOTTLE	0,6226	0,6633
Error rate	0,3467	

(b) Couleur + Cooc + SIFT

	Rappel	Précision
BOTTLE	0,684	0,6444
NO-BOTTLE	0,6226	0,6633
Error rate	0,3467	

(c) Couleur + CoocI + SIFT

	Rappel	Précision
BOTTLE	0,7972	0,6426
NO-BOTTLE	0,5566	0,7329
Error rate	0,3231	

(d) Couleur + Cooc + Hist

	Rappel	Précision
BOTTLE	0,7972	0,6426
NO-BOTTLE	0,5566	0,7329
Error rate	0,3231	

(e) Couleur + CoocI + Hist

	Rappel	Précision
BOTTLE	0,7689	0,6245
NO-BOTTLE	0,5377	0,6994
Error rate	0,3467	

(f) Couleur + SIFT + Hist

Tableau 18: Combinaisons de descripteurs : résultats sur la classe bouteille

	Rappel	Précision
CAT	0,6739	0,5593
NO-CAT	0,4689	0,5898
Error rate	0,4286	

(a) Couleur + Cooc + CoocI

	Rappel	Précision
CAT	0,5186	0,588
NO-CAT	0,6366	0,5694
Error rate	0,4224	

(b) Couleur + Cooc + SIFT

	Rappel	Précision
CAT	0,5155	0,5887
NO-CAT	0,6398	0,5691
Error rate	0,4224	

(c) Couleur + CoocI + SIFT

	Rappel	Précision
CAT	0,677	0,6488
NO-CAT	0,6335	0,6623
Error rate	0,3447	

(d) Couleur + Cooc + Hist

	Rappel	Précision
CAT	0,6739	0,6536
NO-CAT	0,6429	0,6635
Error rate	0,3416	

(e) Couleur + CoocI + Hist

	Rappel	Précision
CAT	0,6429	0,6699
NO-CAT	0,6832	0,6567
Error rate	0,337	

(f) Couleur + SIFT + Hist

Tableau 19: Combinaisons de descripteurs : résultats sur la classe chat

	Rappel	Précision
NO-PERSON	0,4888	0,6
PERSON	0,6741	0,5687
Error rate	0,4185	

(a) Couleur + Cooc + CoocI

	Rappel	Précision
NO-PERSON	0,5097	0,5718
PERSON	0,6183	0,5578
Error rate	0,436	

(b) Couleur + Cooc + SIFT

	Rappel	Précision
NO-PERSON	0,5371	0,5731
PERSON	0,5999	0,5645
Error rate	0,4315	

(c) Couleur + CoocI + SIFT

	Rappel	Précision
NO-PERSON	0,5705	0,6491
PERSON	0,6916	0,6169
Error rate	0,369	

(d) Couleur + Cooc + Hist

	Rappel	Précision
NO-PERSON	0,5695	0,6494
PERSON	0,6926	0,6167
Error rate	0,369	

(e) Couleur + CoocI + Hist

	Rappel	Précision
NO-PERSON	0,5371	0,6337
PERSON	0,6896	0,5984
Error rate	0,3866	

(f) Couleur + SIFT + Hist

Tableau 20: Combinaisons de descripteurs : résultats sur la classe personne

Les combinaisons confirment en premier lieu les bonnes performances de l'histogramme de segments qui se combine bien avec tous les autres descripteurs qui sont des descripteurs de formes plus locaux. On constate en effet que les descripteurs SIFT, CoocI et Cooc semblent capturer le même type d'informations : en effet les gains de performances sur les combinaisons entre SIFT, CoocI et Cooc sont le plus souvent assez faibles. En revanche, la combinaison d'un de ces trois descripteurs avec le descripteur Hist produit une augmentation notable des performances. En particulier on remarque que la combinaison Hist + Cooc + Couleur affiche systématiquement de meilleures performances que l'ensemble des

descripteurs au complet, et des performances comparables sinon meilleures que les autres combinaisons.

Le tableau 21 récapitule les performances de chacune des combinaisons sur les 4 catégories représentatives que nous avons sélectionnées.

Combinaison	Rappel	Précision	Taux d'erreur
Couleur + Cooc + CoocI	0,605	0,607	0,395
Couleur + Cooc + SIFT	0,602	0,604	0,398
Couleur + CoocI + SIFT	0,619	0,621	0,381
Couleur + Cooc + Hist	*0,675*	*0,678*	*0.326*
Couleur + CoocI + Hist	0,660	0,663	0,340
Couleur + SIFT + Hist	0,653	0,656	0,347

Tableau 21: Récapitulatif des meilleures combinaisons de descripteurs

Ce récapitulatif met bien en valeur ce que nous venons de remarquer: les performances de l'ensemble Couleur + matrice de cooccurrence + histogramme sont notablement supérieures aux autres combinaisons de descripteurs.

c) La forme des régions dans une image segmentée participe-t-elle à la discrimination des classes ?

Enfin nous évaluons la pertinence du descripteur basé sur les moments de Hu qui nous permet de caractériser la forme des régions en le combinant aux combinaisons qui se sont déjà mises en évidence au cours de l'expérience précédente :

	Rappel	Précision
NO-BUS	0,7241	0,6632
BUS	0,6322	0,6962
Error rate	0,3218	

(a) Couleur + Hist + Cooc + Moments

	Rappel	Précision
NO-BUS	0,6207	0,675
BUS	0,7011	0,6489
Error rate	0,3391	

(a) Couleur + Hist + SIFT + Moments

Tableau 22: Evaluation des moments de HU sur la classe Bus

	Rappel	Précision
BOTTLE	0,8208	0,5959
NO-BOTTLE	0,4434	0,7121
Error rate	0,3679	

(a) Couleur + Hist + Cooc + Moments

	Rappel	Précision
BOTTLE	0,7642	0,6304
NO-BOTTLE	0,5519	0,7006
Error rate	0,342	

(b) Couleur + Hist + SIFT + Moments

Tableau 23: Evaluation des moments de HU sur la classe Bus

	Rappel	Précision
CAT	0,6398	0,6776
NO-CAT	0,6957	0,6588
Error rate	0,3323	

(a) Couleur + Hist + Cooc + Moments

	Rappel	Précision
CAT	0,6149	0,6923
NO-CAT	0,7267	0,6536
Error rate	0,3292	

(b) Couleur + Hist + SIFT + Moments

Tableau 24: Evaluation des moments de HU sur la classe Chat

	Rappel	Précision
NO-PERSON	0,568	0,6499
PERSON	0,6941	0,6164
Error rate	0,369	

(a) Couleur + Hist + Cooc + Moments

	Rappel	Précision
NO-PERSON	0,5267	0,6577
PERSON	0,726	0,6053
Error rate	0,3737	

(b) Couleur + Hist + SIFT + Moments

Tableau 25: Evaluation des moments de HU sur la classe Personne

Nous étudions ici l'ajout des moments de Hu à des combinaisons de descripteurs parmi les plus performantes. On constate qu'à l'exception de la classe "chat" où la forme des régions semble caractéristique, les moments de Hu ont plutôt tendance à nuire aux performances générales de notre classificateur. Ceci signifie que, dans la plupart des cas, les performances du descripteur ne sont pas suffisantes pour compenser l'augmentation de taille que son ajout engendre. Ces résultats étant bien entendus propres à notre implémentation et à la taille des jeux de données que nous manipulons ici.

On notera également que nous avons rencontré des problèmes d'overfitting. En effet la plupart des classes sont assez peu représentées par rapport aux dimensions des vecteurs de caractéristiques. Nous avons tenté de faire varier les algorithmes de décision. Ainsi l'utilisation de méthodes de décision réputées précises (SVM avec des fonctions de noyau gaussiennes, perceptrons multicouches avec un important nombre de neurones cachés, …) souffre du manque d'exemples pour une classe donnée. Seule la classe "personne" étant convenablement représentée. Les meilleures performances peuvent ainsi, dans la plupart des cas, être obtenues pour des SVM avec une fonction noyau linéaire.

d) Récapitulatif des différentes combinaisons de descripteurs

Les expériences précédentes nous poussent donc à exclure les descripteurs de forme globale de notre classificateur. Nous retenons donc la combinaison de descripteurs qui nous apparait comme la plus performante avec les données dont nous disposons. Evaluation sur la base Pascal VOC 2007

Etant donné ce premier test sur les combinaisons de caractéristiques, nous allons maintenant évaluer notre algorithme de classification sur l'ensemble de la base pascal VOC 2007 avec la combinaison Couleur + Hist + Cooc. Les performances de classification (précision/rappel) sont données dans le
Tableau 26 .

Catégorie	Rappel	Précision	Taux de classification correcte
AEROPLANE	0,7353	0,8065	78%
BICYCLE	0,8368	0,5634	59%
BIRD	0,695	0,6712	68%
BOAT	0,7384	0,8141	78%
BOTTLE	0,7972	0,6426	68%
BUS	0,7759	0,7143	73%
CAR	0,7684	0,648	68%
CAT	0,677	0,6488	66%
CHAIR	0,6978	0,6847	69%
COW	0,7244	0,7023	71%
DOG	0,7632	0,673	70%
HORSE	0,7445	0,6476	67%
MOTORBIKE	0,7703	0,631	66%
PERSON	0,6916	0,6169	63%
PLANT	0,6429	0,5692	58%
SHEEP	0,7526	0,7766	77%
SOFA	0,7175	0,6987	70%
TABLE	0,7789	0,6789	71%
TRAIN	0,7452	0,7148	72%
TV	0,738	0,668	69%
Moyenne	0,739545	0,67853	69%

Tableau 26: Résultats de classifications moyens sur 5 jeux d'exemples négatifs (base pascal VOC 2007)

On rappelle les conditions expérimentales : il s'agit de classificateurs binaires (Classe/non-classe) et les résultats sont la moyenne des résultats sur 5 jeux (toutes les données positives d'une classe + autant de données négatives tirées aléatoirement mais à chaque fois distinctes – dans la mesure du possible : la forte population de la classe personne ne permettant par exemple pas ceci).

Concernant les différentes classes, on notera les bonnes performances de classes dont le contexte est facilement identifiable : les bateaux et les avions. Ces deux classes paraissent intuitivement plus simples avec leurs formes géométriques assez caractéristiques malgré les changements d'aspect ainsi que leur contexte d'apparition (ciel, mer, aéroports, ports) bien défini. Elles conviennent particulièrement à une segmentation (formes qui se détachent bien) et aux descripteurs de segments. A l'opposé de l'échelle, on va trouver les classes vélo (cadre fin, difficile à segmenter combiné à une forte possibilité de confusion avec une moto – visible avec la faible précision par rapport au rappel de ces deux classes) et plante (même difficulté à segmenter, rarement de taille importante dans l'image). Ces résultats sont globalement encourageants étant donnée la grande difficulté de la base et les marges d'amélioration de notre algorithme par rapport au processus d'apprentissage et de classification pour lequel nous avons simplement utilisé une solution "clefs en main".

5.4. Discussion

Nous avons proposé une méthode de classification basée sur la notion de "vocabulaire visuel". Nous avons choisi de ne pas expérimenter la fusion précoce pour plusieurs raisons : d'abord parce que la "curse of dimensionnality" nous a empêchés de construire un vocabulaire pertinent lorsque la taille combinée des descripteurs dépassait 130. De plus, les expériences que nous avons conduites dans [133] (et résumés dans l'annexe 2) ont montré les médiocres performances de ce mode de fusion. A la place, nous avons donc proposé une stratégie de fusion intermédiaire qui consiste à produire un vocabulaire visuel par catégorie de descripteurs et qui permet de séparer les différentes composantes et d'en réduire les dimensions autant que nécessaire avant de les fusionner dans un algorithme de classification globale. Ceci nous permet de multiplier les canaux d'acquisition des données sans rendre les résultats inexploitables en raison d'un vecteur de caractéristiques trop important. Par ailleurs, l'étude séparée des diverses caractéristiques a confirmé la performance de nos descripteurs par rapport, notamment, au populaire descripteur SIFT. Une première expérimentation sur la difficile base Pascal 2007 nous donne des résultats encourageants et nous conforte dans la pertinence de notre approche perceptuelle.

6. Conclusion

En résumé, ce chapitre nous a permis d'introduire une nouvelle méthode d'expression des descripteurs par la fusion de vocabulaires spécifiques à chaque descripteur. Nous avons aussi pu distinguer deux types de problèmes de classification sémantique que sont la classification globale et la recherche d'objets visuels et ainsi mis en lumière la différence entre les approches efficaces sur chacun de ces deux problèmes. Ces expériences demeurent incomplètes à bien des égards et présentent des perspectives intéressantes de recherches futures. La détermination de la taille du vocabulaire est un point central de la méthode qui mérite d'être approfondi (ex : utilisation d'autres mesures au lieu de la MSE).

Notons enfin que les performances en question ne sont pas comparables avec les résultats évoqués dans [133] qui ne se basent que sur un sous-ensemble de la base pascal. A ce titre, des tests de toutes les méthodes de fusion (en particulier intermédiaire et tardive) sur la base complète (avec toutes les données négatives) seraient intéressantes à réaliser même si elles nécessitent la mise au point d'un classificateur adéquat.

Chapitre 7: Perspectives et conclusion

Dans cette thèse, nous avons tenté d'apporter des solutions au difficile problème de la classification sémantique d'images sous ses deux aspects complémentaires, à savoir la catégorisation globale d'images et la classification d'objets visuels. Nous avons eu une démarche constamment inspirée de la perception humaine, allant de la segmentation jusqu'à la classification en passant par la génération de descripteurs. Nous résumons ici d'abord nos contributions pour tracer ensuite les perspectives.

1. Nos contributions

La classification d'images par le contenu visuel est un domaine particulièrement actif et difficile de l'analyse d'images. En n'imposant aucune restriction sur les images traitées, on se retrouve en effet face à un contenu qui peut être composite, ambigu et qui plus est acquis dans de mauvaises conditions. Aussi difficile qu'elle puisse paraître, cette activité pose pourtant très rarement des problèmes à un être humain qui, quelle que soit la complexité de l'image d'origine, parvient toujours très rapidement à une décision.

Idéalement un système d'indexation automatique devrait permettre de rechercher des concepts dans une image hétérogène et de savoir détecter leur présence comme leur absence de manière non-mutuellement-exclusive. Notre objectif a d'abord été de nous inspirer de la performance de la classification humaine pour en tirer des procédés d'analyse nous mettant dans de bonnes conditions pour nous acquitter de cette tâche. Nous avons également déterminé des caractéristiques de forme pertinentes pour nous assister dans la tâche de classification. Enfin nous avons développé une classification efficace qui puisse s'adapter à ces conditions difficiles.

Notre premier axe de travail a concerné la couleur et en particulier la décomposition de l'image en régions de couleur homogène. A ce titre nous avons apporté deux contributions : la première est un algorithme de réduction du nombre de couleurs dans l'image mettant l'accent sur l'optimisation de la conservation de l'information de couleur telle qu'elle est perçue. Cet algorithme produit une erreur quadratique de quantification (MSE) inférieure aux algorithmes de l'état de l'art. La seconde est un algorithme de segmentation en régions se basant sur cette réduction ; il suit également les principes Gestalt et apporte, par rapport aux algorithmes de l'état de l'art une robustesse nécessaire pour traiter automatiquement une grande quantité d'images non contraintes.

Notre troisième contribution se situe au niveau des descripteurs utilisés. Toujours en relation avec les théories de la perception, nous cherchons à extraire de manière fiable des données de forme plus informatives que des données à l'échelle du pixel actuellement utilisées. En se basant sur la détection de segments pour l'analyse des lignes de fuite développée par Mohsen Ardabilian [15], nous avons développé une série de descripteurs de forme qui se sont avérés efficaces pour la caractérisation de contenu visuel. Les tests que nous avons réalisés avec différents mécanismes de classification que nous avons développés montrent des performances de classification supérieures aux populaires descripteurs SIFT.

Notre dernière contribution se trouve dans la classification. A l'aide des composants que nous avons développé et en suivant toujours notre ligne directrice constituée par les théories de la perception humaine, nous mettons en place une classification basée sur la notion

de "vocabulaire visuel". Nous introduisons un histogramme construit sur la notion d'appartenance floue d'un vecteur de caractéristiques aux "mots" d'un vocabulaire construit pour chaque type de caractéristiques. Cette méthode de classification produit des résultats consistants sur la difficile base Pascal VOC 2007 [3].

2. Perspectives

Bien que les résultats expérimentaux sont prometteurs, les bases d'images difficiles comme la base Pascal, ne permettent pas encore d'obtenir des résultats acceptables pour une utilisation industrielle par exemple. Il existe donc de très nombreux axes d'améliorations.

2.1. Amélioration des techniques développées

Les descripteurs basés sur les segments que nous avons introduits produisent déjà des résultats intéressants, nous pourrions toutefois y intégrer les principes de perception Gestalt afin de le focaliser sur les caractères les plus discriminants, l'idée étant de mettre en relief les segments les plus rares (par exemple les plus longs) et les caractériser plus finement que les segments plus communs qui seraient eux exprimés en termes statistiques. Ceci correspond également à un phénomène que nous avons pu observer qui est celui des segments courts présents en grand nombre qui représentent souvent des informations que l'on pourrait assimiler à des informations de texture. La séparation de ces deux types d'informations ainsi que leur caractérisation par différents moyens constitue une façon intéressante d'obtenir de l'information à partir des segments au sein d'une région. La structure locale n'étant pas exprimée, il est de plus possible qu'un tel descripteur se combine bien avec les matrices de cooccurrence.

D'autres pistes d'améliorations concernent le procédé de segmentation. D'abord dans le procédé lui-même avec la détermination automatique du nombre de clusters de couleurs. Une modélisation précise de la courbe ou un lissage qui permettrait une étude fiable de sa dérivée constituerait une solution plus précise et plus robuste que la méthode actuelle. Nous pourrions de surcroit tester d'autres critères que la MSE (information mutuelle, etc.).

Il serait également intéressant de poursuivre le travail de l'inspiration perceptuelle en intégrant des critères issus des lois Gestalt dans l'étape de fusion des régions, par exemple la tendance à créer des régions fermées (loi de clôture). Notons de plus que le procédé de détermination automatique du nombre de clusters que nous venons d'évoquer se retrouve dans la détermination de la taille du vocabulaire : les améliorations que nous pourrions apporter à ce procédé pourraient y être répercutées.

Enfin d'une manière générale il faut noter que nos algorithmes ont été implémentés de façon à être modulaires et ainsi permettre de substituer différentes techniques les unes aux autres afin de choisir la combinaison la mieux adaptée. La plupart des algorithmes utilisés ne font pas intervenir des opérations de forte complexité et gagneraient à être implémentés plus spécifiquement et optimisés.

2.2. Incorporation de nouvelles caractéristiques

L'incorporation de nouvelles caractéristiques est un autre axe d'améliorations important. Au cours de la thèse de nombreuses autres caractéristiques ont été abordées mais n'ont pas pu être complètement étudiées, faute de temps.

L'incorporation de caractéristiques de texture est un axe de travail majeur. Si nous avons choisi de retranscrire indirectement les informations de texture par le biais des matrices de cooccurrence de segments, une retranscription précise demanderait, en toute rigueur, d'aborder le problème de la segmentation en texture. Par ailleurs, nous avons émis l'idée d'analyser la texture par l'extraction de statistiques locales sur les caractéristiques des segments détectés dans les sous-images issues d'une transformées en ondelettes. Ceci constituerait également une piste pour des travaux sur une nouvelle méthode de caractérisation de la texture.

Nous avons également la possibilité d'élargir la notion de segments à d'autre primitives géométriques particulièrement indicatives pour certaines catégories de contenu. Il apparaitrait ainsi intéressant de caractériser les ellipses, qui sont des structures qui apparaissent dans de nombreuses catégories et sous différentes formes (yeux, roues de véhicules, …).

Plus globalement, les mécanismes de fusion abordés dans cette thèse permettent l'inclusion d'un grand nombre de descripteurs qui pourraient s'avérer utile afin de perfectionner les mécanismes de classification de contenu issus de nos travaux.

Par ailleurs, nous nous sommes attachés, tout au long de cette thèse, à suivre les principes de la perception humaine et notamment les lois Gestalt. Il apparait toutefois que nos caractéristiques basées sur des régions ne permettent pas encore de faire le passage entre le local et le global. Ceci est particulièrement visible lors des expériences sur la caractérisation globale lorsqu'on utilise des caractéristiques basées sur les régions. L'étude et la caractérisation des relations entre les régions mériteraient d'être plus poussées, notamment à partir de méthodes développées au sein de notre équipe comme dans [2].

Enfin, dans un tout autre registre, dans l'optique d'une application de nos travaux au travail sur des photographies numériques, il nous est paru judicieux de développer le travail sur les métadonnées EXIF incluses par tous les appareils numériques actuels et qui donnent de précieuses informations sur les circonstances de la prise de vue (utilisation du flash, ouverture, vitesse, éventuel programme résultat utilisé, etc.). Si nous avons développé cet outil, les bases d'évaluation actuelles ne présentent malheureusement pas ce type d'informations qui seraient pourtant particulièrement utiles pour la réalisation d'un classificateur performant en vue d'une utilisation directe pour, par exemple, la structuration d'albums de photos.

2.3. Procédés de classification

La technique de fusion des informations par des vocabulaires visuels permet d'obtenir de bons résultats mais ne doit pas être considérée exclusivement. Nous devons également implémenter et expérimenter la fusion tardive ou encore d'autre stratégies de fusion intermédiaire afin de pouvoir réellement évaluer notre méthode de fusion. La méthode de la fusion des caractéristiques par modélisation polynomiale [133] est également une piste explorée dans notre équipe et a produit des résultats prometteurs.

Pour ce qui est de l'algorithme de classification supervisé utilisé pour la décision de classification finale, nous avons utilisé les réseaux de neurones tout simplement parce qu'ils présentaient un compromis entre performance et flexibilité d'utilisation. Comme nous l'avons remarqué, la plupart des travaux actuels travaillent ou avec une méthode de classification propre (méthode des "Fisher Kernels" [145]) ou en tentant d'optimiser les paramètres de SVM pour obtenir de meilleurs performances de classification [156]. Il s'agit là encore d'un axe d'amélioration significatif.

Enfin l'algorithme de classification global bénéficierait d'être testé avec une méthode de fusion tardive des résultats du classificateur par région et des résultats du classificateur global afin d'identifier si les informations produites par ces deux méthodes sont bien de nature distincte.

Annexes

1. Etude sur les systèmes CBIR

Le tableau suivant fournit un bref récapitulatif de systèmes CBIR existants et est principalement basé sur des études globales pratiquées dans [157] et [158], il couvre une assez grande variété de systèmes sans prétendre être exhaustif:

Système CBIR	Caractéristiques							Requête sur…
	Couleur	Texture	Forme	Position	Relation spatiale	Contours	Autres	
ALISA	Intégré à la texture	Caractéristiques incluant aussi les informations de couleur (par canal)	Déterminée par un ensemble de contours (invariant en échelle et rotation)	-	-	Déterminés par les groupes de textures homogènes	Possibilité de « découvrir » de nouvelles caractéristiques en combinant différents éléments du module texture	Exemples
ASSERT (domaine médical)	-	Matrice de cooccurrence	Moments et descripteurs de Fourier	-	-	Rapport pixels de contour / taille région	Des zones d'intérêt particulières sont marquées à la main.	Exemples + Relevance Feedback
Blobworld	Histogramme par région espace Lab	Point dans un espace 3D correspondant à 3 caractéristiques	-	Segmentation selon couleur, texture et position Coordonnées absolues	Position relative des regions (droite, gauche, haut, bas)	-	-	Exemples + sélection manuelle de Caractéristiques
C-Bird	Histogramme RGB normalisé + Couleurs les plus fréquentes (régions arbitraires)	Coefficients matrice DCT	-	Décomposition de l'image en 64 cases de même taille.	-	Présent mais non décrit	-	Selection manuelle des caractéristiques souhaitées

Nom	Couleur	Texture	Forme	Position	Relation spatiale	Contours	Autres	Requête…
Chabot	Histogramme RGB réduit	-	-	-	-	-	-	Selection manuelle
CBVQ	Histogramme	Energies d'Ondelettes de Haar	-	Segmentation en texture Segmentation en couleurs Dimension et position absolue des régions	-	-	-	Exemples + Selection manuelle des caractéristiques
DrawSearch	Couleur moyenne des régions	Markovian Random Fields	Descripteurs de Fourier	Division en régions 4*4	-	-	-	Croquis + Selection manuelle
Excalibur	Histogramme HSV	Caractéristiques	Orientation relative des lignes de contour	-	-	Présent mais non décrit	Modules spécialisés pour les empreintes digitales et les visages	Selection manuelle suivie de choix d'un exemple dans la base
Fast Multiresolution	Intégré à la texture	Décomposition en ondelettes de Haar par canal (R,G,B)	-	-	-	-	-	Croquis
FIDS	Histogramme Pixels de couleur « chair »	Statistiques sur les contours, Coefficients d'ondelettes de Haar	-	-	-	Sobel	-	Exemple, paramètres manuels
FIR	Intégré à la texture	Coefficients d'ondelettes de Haar par canal (LUV)	-	-	-	-	-	exemple

Nom	Couleur	Texture	Forme	Position	Relation spatiale	Contours	Autres	Requête…
FOCUS	Couleurs les plus fréquentes (espace HSV) dans chaque région	-	-	Divison en cases 100*100	Spatial Proximity Graph (graphe qui représente les couleurs adjacentes)	-	-	A partir d'exemples de la base ou de parties d'images de la base
ImageMiner	Histogrammes	Matrice de cooccurrence (réseau de neurones produisant des caractéristiques données)	Taille et coordonnées des bords des régions Rectangle englobant	Division en cases de taille fixe (non précisée). Puis agglomération des cases de couleurs et textures similaires.	Relations entre les groupes de cases représentées dans un graphe d'adjacence	-	-	Requêtes de type sql sur les caractéristiques utilisées
ImageRETRO	Histogramme HSV, Couleur la plus représentée, Variation de couleur, Taux de gris, Proportion « d'arrière plan » (déterminé par la teinte)	-	-	-	Détection de régions de couleur incluses dans une autre	-	Recherche par clustering	Choix d'une cascade d'images représentatives, utilisation de relevance feedback
ImageRover	Histogramme Luv Réduit à 64 bins	Detection des orientations de textures par des « Steerable pyramids »	-	5 régions fixes : Centre et 4 coins.	-	-	Réduction de l'espace de recherche par PCA.	Mots clés puis relevance feedback

Nom	Couleur	Texture	Forme	Position	Relation spatiale	Contours	Autres	Requête…
ImageScape / ImageSearch	Couleurs dominantes	Caractéristiques des textures (non précisées)	Histogramme de motifs de contours 3*3, Descripteurs de Fourier, Moments invariants	-	-	Sobel	Mesure d'information relative de Kullback pour trouver les critères les plus informatifs	Croquis, objets pré-définis (ciel, …)
Jacob	Histogramme RGB	Matrice de cooccurrence	pixels de contours / nombre de pixels	-	-	Présent mais non décrit	-	Exemple, paramètres manuels
LCPD	Intensité (projections en x et y – images en niveau de gris)	Histogramme de motifs 3*3	-	-	-	Gradient (projections en x et y)	Spécialisé pour les visages Réduction de la dimension par analyse des composantes principales	Exemple, paramètres manuels
MARS	Histogramme, Moments (espace HSV)	Modèle de Tamura, Matrice de cooccurrence	Descripteurs de Fourier	Décomposition en cases 5*5, segmentation dans l'espace couleur/texture (adapté à la base d'images pour détecter des objets uniques)	-	-	Coefficients de décomposition en ondelettes (ondelette mère non précisée)	Exemple, requête booléenne sur des paramètres manuels, relevance feedback

Nom	Couleur	Texture	Forme	Position	Relation spatiale	Contours	Autres	Requête…
Multiresolution Image Database Search	Histogramme Lab lissé	Histogrammes des intensités du gradient pour chaque case	-	Décomposition en cases 4*4	-	Gradient et orientation du gradient	Utilisation de quantification vectorielle en arbre (TSVQ) pour accélérer.	Paramètres manuels puis choix d'un exemple.
Netra	Proportions de couleurs dans chaque région (espace RGB)	Gabor	Descripteurs de Fourier, Rectangle englobant	Rectangle englobant et centre de gravité de chaque région, Segmentation en 6-12 régions (couleur / texture)	-	-	-	Exemple, Région d'une image, paramètres manuels
PhotoBook	-	Une texture est considérée comme un champ aléatoire. Plusieurs modèles existent pour rechercher plusieurs types de textures.	Modèle de contours élastiques selon la méthode des éléments finis	-	-	Présent mais non décrit	Analyse en composantes principales (eigenfaces)	Exemples, relevance feedback
Picasso	Vecteur de couleurs quantifiées (espace non précisé)	-	Ellipses englobantes	Segmentation en couleurs, Centre de gravité de chaque région, rectangles englobants.	2D Strings	Canny edge detector	Analyse pyramidale de l'image (méthode exacte non précisée)	Croquis, paramètres manuels

Nom	Couleur	Texture	Forme	Position	Relation spatiale	Contours	Autres	Requête…
PicHunter	Histogramme HSV, « Auto-correllogram », Vecteur de cohérence (RGB)	-	-	-	-	-	-	Exemple, relevance feedback
PicSOM	Valeurs moyennes dans chaque région (RGB)	Pour chaque région : probabilités sur les intensités relatives des pixels	Descripteurs de Fourier, matrices de cooccurrence des contours	5 régions fixes (centre, haut, bas, droite, gauche)	-	Sobel	Quantification vectorielle par le biais de SOMs (TS-SOMs)	Exemple puis relevance feedback
PicToSeek	Histogramme RGB normalisé	-	Detecteur d'angles, histogramme d'orientation des contours	-	-	Présent mais non décrit	-	Exemple et selection manuelle de la similarité recherchée
QBIC	Par objet ou pour l'image : Couleur moyenne dans différents espaces (RGB, YIQ, Lab, Munsell), Histogramme RGB réduit	Variante des caractéristiques du modèle de Tamura	Surface, Circularité, Moments invariants, Angles des gradients.	-	-	Présent mais non décrit	Extraction d'objets semi automatisée par contours actifs ou par croissance de région	Exemple, croquis, sélection manuelle des critères

Nom	Couleur	Texture	Forme	Position	Relation spatiale	Contours	Autres	Requête…
Quicklook	Histogramme RGB, Vecteur de cohérence LAB, Barycentre de chaque couleur quantifiée, Cooccurrence LAB, Informations de contraste aux points de contour	Caractéristiques extraites de l'intensité, Coefficients de la transformation en ondelettes de Daubechies.	Moments invariants, histogramme des directions des gradients	-	-	Canny edge detector	-	Exemple puis relevance feedback
SIMPLIcity	Présent mais non décrit	Présent mais non décrit	Présent mais non décrit	Décomposition en cases 4*4 ainsi qu'en régions de couleur. Autres (non décrits)	-	-	La segmentation en régions dirige vers un type sémantique d'image. Au sein de chaque type, la similarité est calculée par des caractéristiques différentes.	Exemple
SQUID	-	-	Image dans le « Curvature scale space » : expression des changements de courbure	-	-	Contours extraits d'un objet unique par seuillage	-	Exemple issu de la base

Nom	Couleur	Texture	Forme	Position	Relation spatiale	Contours	Autres	Requête…
Surfimage	Histogramme RGB	Transformée en ondelettes (ondelette mère non précisée), matrice de coocurrence	Histogramme de courbure	Histogramme d'orientation des contours	-	Canny edge detector	ACP : « Eigen Images »	Exemples, Relevance feedback
SYNAPSE	-	-	Invariants associés à chaque point d'intérêt	-	-	-	Points d'interet	Exemple
Virage Image Engine	Couleur dominante, Variations dans chaque canal (HSV)	Caractéristiques (granularité, répétitivité, …)	Caractérisation globale des formes distinctement détectées (methode exacte non précisée)	-	Positions relatives de régions de couleurs (non précisé)	Présent mais non spécifié	-	Paramètres manuels
VisualSeek	Couleur de chaque région (espace HSV)	-	Surface et Rectangle englobant pour chaque région.	Décomposition en régions de couleur (BackProjection) Centre de gravité	-	-	-	Exemple ou croquis représentant l'organisation de régions colorées
WebSeer	Etude de la diversité de couleurs pour distinguer les photos des dessins Histogramme RGB	-	-	-	-	-	Texte (recherche web utilisant le contexte de la page et le nom du fichier)	Mots clés, Paramètres manuels

STATISTIQUES						
Couleur	**Texture**	**Forme**	**Position**	**Relation spatiale**	**Contours**	**Segmentation**
88,6 % Pour lesquels on rapporte les utilisations suivantes : **61,3% histogrammes** 19,4% les n plus fréquentes 12,9% couleur moyenne 9,7% analysées avec la texture 6,4% Vecteur de cohérence 3,2% Correlogramme 3,2% Moments de couleur 12,9 % autres méthodes (ou non précisées)	**71,4%** Pour lesquels on rapporte les utilisations suivantes : **48% caractéristiques/cooccurrence** **36% ondelettes/gabor/pyramides** 8% champs de Markov 8% basés sur les contours 4% basés sur la DCT 12% autres méthodes (ou non précisées)	**60%** Pour lesquels on rapporte les utilisations suivantes : 38% basés sur les contours (snakes + descripteurs basés sur la courbure, hors Fourier) 28,6% descripteur de Fourier 23,8% moments invariants 19% formes englobantes (rectangles le plus souvent) 9,5% autres méthodes (ou non précisées)	**37,1%**	**17%**	**51%** (le plus souvent utilisés pour la texture ou la forme)	**42,9%** Pour lesquels on rapporte les utilisations suivantes : **40% de regions fixes** 26,6% par couleur et texture 26,6% par couleur 6,7% par texture

2. **Résultat de l'évaluation des descripteurs basés sur les segments**

(a) Classification rate

Classification rate	Plane	Bicycle	Bus	Horse	Person
SIFT	65,00%	55,21%	60,75%	65,49%	58,94%
RCM	72,69%	61,57%	67,90%	65,84%	62,77%
RHS	76,60%	61,98%	66,13%	62,59%	63,45%
EF(RCM+RHS)	80,34%	63,97%	70,75%	65,63%	65,17%
EF(RCM+RHS+SIFT)	81,47%	64,63%	69,30%	66,43%	65,50%
LF(RCM+RHS)	82,02%	70,95%	91,99%	79,65%	66,74%
LF(RCM+RHS+SIFT)	85,21%	72,73%	92,74%	81,54%	69,41%

(b) Recall rate

Recall rate	Plane	Bicycle	Bus	Horse	Person
SIFT	68,66%	57,90%	62,58%	71,57%	60,93%
RCM	73,45%	64,10%	68,06%	66,53%	67,27%
RHS	76,55%	68,32%	71,61%	67,09%	69,11%
EF(RCM+RHS)	80,17%	65,67%	70,86%	67,09%	68,42%
EF(RCM+RHS+SIFT)	81,43%	66,67%	70,54%	70,10%	68,57%
LF(RCM+RHS)	84,20%	73,86%	89,35%	79,48%	70,01%
LF(RCM+RHS+SIFT)	85,38%	74,86%	89,78%	83,89%	72,89%

(c) Precision rate

Precision rate	Plane	Bicycle	Bus	Horse	Person
SIFT	63,98%	54,90%	60,37%	63,76%	58,60%
RCM	72,35%	60,98%	67,85%	65,56%	61,72%
RHS	76,62%	60,60%	64,53%	61,49%	62,08%
EF(RCM+RHS)	80,44%	63,47%	70,71%	65,13%	64,24%
EF(RCM+RHS+SIFT)	81,50%	64,02%	68,84%	65,25%	64,61%
LF(RCM+RHS)	80,68%	69,77%	94,32%	79,71%	65,72%
LF(RCM+RHS+SIFT)	85,09%	71,77%	95,43%	80,08%	68,14%

Il s'agit de résultats extraits de [133] et obtenus à partir de tests de validation croisée (4 jeux, les résultats présentés sont la moyenne des résultats de 5 partitions/exécutions indépendantes) sur des classificateurs binaires entrainés sur chaque catégorie. Le nombre d'exemples positifs et le même que le nombre d'exemples négatifs pour chaque classe. Les sigles RCM et RHS désignent respectivement les moments de couleur et les histogrammes de segments, LF désigne une fusion tardive et EF une fusion précoce (voir chapitre 6).

3. Exemples d’images de la base PASCAL VOC 2007

3.1. Exemples d’images de la classe vélo

3.2. Exemple d’images de la classe chaise :

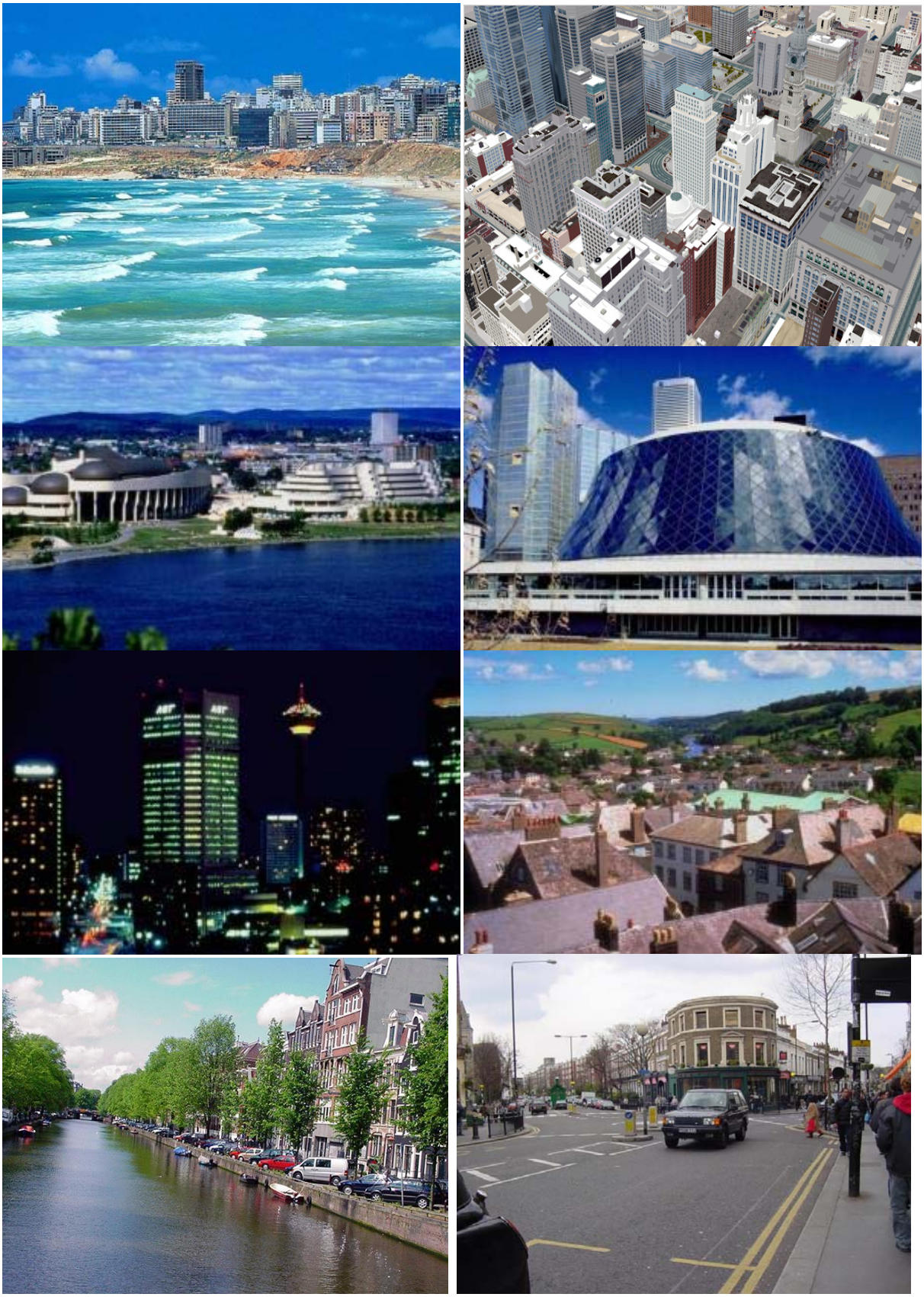

3.3. Exemples d’images de la classe "train"






4. Exemples d’images de la base ville/non-ville

4.1. Exemples d’images de la classe "ville"



4.2. Exemples d’images de la classe "non-ville"







5. Exemples d’images de la base de classification globale

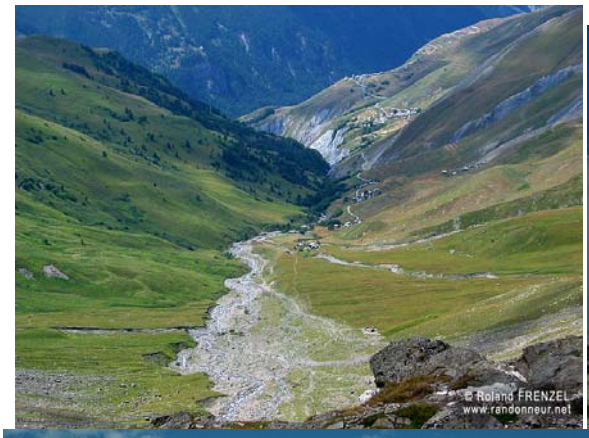
5.1. Exemples d’images de la classe "coucher de soleil"

5.2. Exemples d’images de la classe "mer/paysages maritimes"


5.3. Exemples d’images de la classe "plage/désert"

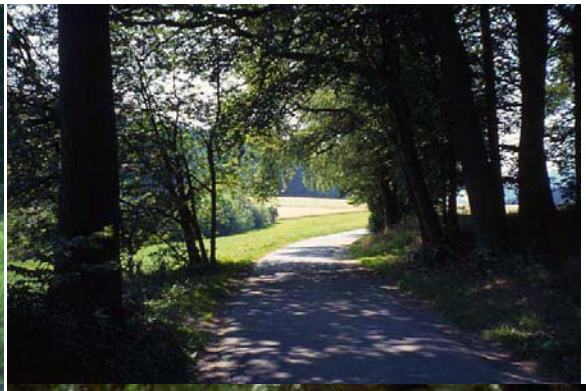



5.4. Exemples d’images de la classe "montagne"





5.5. Exemples d’images de la classe "forêt/verdure"

6. Evaluation des caractéristiques avec plusieurs algorithmes de classification

On se propose ici d'évaluer une série d'algorithmes différents avec un même jeu de caractéristiques sur notre base de 600 images (dont quelques exemples sont donnés dans l'annexe 5). Le but étant de confirmer que les performances de nos caractéristiques sont indépendantes de l'algorithme de classification. On procèdera pour cela à 5 tirages de validation croisée sur 10 groupes, 4 groupes et 2 groupes.

Les résultats détaillés sont présentés pour 10 groupes, avec les caractéristiques de cooccurrence. On précisera seulement le taux d'erreur pour les deux autres expériences qui produisent un taux voisin et montrent ainsi que les résultats sur 10 groupes sont représentatifs. Ces résultats servent de témoin pour observer leur évolution, sachant que toutes les caractéristiques sont affectées de la même façon. On ne fournira donc qu'un tableau récapitulatif des taux de classification en fin de section.

6.1. Résultats détaillés sur les caractéristiques de cooccurrence

6.1.1. Arbres de décision (C4.5)

Théoriquement l'algorithme de partition le moins efficace, il n'est ni particulièrement adapté au type de données, ni aussi puissant, dans l'absolu, que les réseaux de neurones ou les SVM. Cet algorithme est présent à titre de référence et, sans surprise, ses résultats sont en retrait par rapport aux autres méthodes.

Tirage	Taux d'erreur
1	0,4833
2	0,4900
3	0,4650
4	0,4950
5	0,4983

Tableau 27 : Taux d'erreur pour chaque tirage (C4.5)

Taux d'erreur moyen	0,4863

Résultats par classe

Classe	Rappel	Précision
CDS	0,6580	0,7263
Mer	0,5280	0,4774
Montagne	0,3320	0,312
Verdure	0,5580	0,5717
Ville	0,5760	0,5772
Plage	0,4300	0,4526

Matrice de confusion

	CDS	Mer	Montagne	Verdure	Ville	Plage	Total
CDS	329	48	43	14	19	47	500
Mer	25	264	114	10	11	76	500
Montagne	23	87	166	78	66	80	500
Verdure	21	29	77	279	70	24	500
Ville	12	15	63	89	288	33	500
Plage	43	110	69	18	45	215	500
Total	453	553	532	488	499	475	3000

Tableau 28 : Résultats détaillés de classification (C4.5)

Les taux d'erreur sur les validations croisées avec 4 et 2 groupes sont respectivement de 0,4767 et 0,5063.

6.1.2. K-Plus proches voisins

Cet algorithme simple a été évalué parce que les dimensions de l'espace de caractéristiques étaient assez importantes par rapport à la population de test et qu'il n'était par conséquent pas évident que la masse de données soit suffisante pour des algorithmes de classification supervisée plus complexes et plus précis. Les résultats obtenus sont les suivants:

Tirage	Taux d'erreur
1	0,4117
2	0,4033
3	0,4000
4	0,3900
5	0,4000

Tableau 29 : Taux d'erreur pour chaque tirage (KNN)

Taux d'erreur moyen	0,4010

Résultats par classe

Classe	Rappel	Précision
CDS	0,7500	0,6696
Mer	0,4800	0,6061
Montagne	0,4740	0,4788
Verdure	0,7560	0,6176
Ville	0,7260	0,6471
Plage	0,4080	0,5426

Matrice de confusion

	CDS	Mer	Montagne	Verdure	Ville	Plage	Total
CDS	375	33	35	14	26	17	500
Mer	65	240	80	29	19	67	500
Montagne	33	26	237	68	66	70	500
Verdure	4	7	51	378	52	8	500
Ville	17	16	11	83	363	10	500
Plage	66	74	81	40	35	204	500
Total	560	396	495	612	561	376	3000

Tableau 30 : Résultats détaillés de classification (KNN)

Les taux d'erreur pour les validations croisées sur 4 et 2 groupes étant respectivement de 0,4173 et 0,4460.

6.1.3. Réseaux de neurones

Algorithme de classification supervisée présentant traditionnellement de bonnes performances, il impose toutefois quelques choix en termes de paramétrage, essentiellement la taille de la couche cachée et l'algorithme d'apprentissage.
Le choix initial de la taille de la couche cachée s'est effectué de manière purement expérimentale en prenant en compte le nombre d'exemples disponibles, la taille du vecteur de caractéristiques et la complexité attendue du problème (à priori assez importante). Des essais ont donc été effectués à 25, 30, 35, 40, 45 et 50 neurones avec de meilleures performances obtenues pour entre 35 et 45 neurones sur la couche cachée (résultats voisins aux variations dues à la sélection aléatoire près). La plateforme de tests utilisée ne permettait que d'utiliser un apprentissage basique par rétropropagation du gradient ; des tests effectués séparément ont montré que l'utilisation d'algorithmes plus efficaces (Levenberg-Marquardt, Scaled

Conjugate Gradient, …) permettait de gagner quelques % sur le taux d'erreur. Le taux obtenu restait toutefois légèrement en deçà de celui obtenu avec les SVM mais était tout à fait comparable. Voici les résultats obtenus pour 35 neurones sur la couche cachée:

Tirage	Taux d'erreur
1	0,3633
2	0,3367
3	0,3433
4	0,3267
5	0,3517

Tableau 31 : Taux d'erreur pour chaque tirage (MLP)

Taux d'erreur moyen	0,3443

Résultats par classe

Classe	Rappel	Précision
CDS	0,8140	0,8108
Mer	0,6120	0,6157
Montagne	0,5360	0,5134
Verdure	0,6860	0,6725
Ville	0,7220	0,722
Plage	0,5640	0,6013

Matrice de confusion

	CDS	Mer	Montagne	Verdure	Ville	Plage	Total
CDS	407	26	17	4	20	26	500
Mer	26	306	76	11	15	66	500
Montagne	15	67	268	63	25	62	500
Verdure	6	10	68	343	62	11	500
Ville	12	23	25	57	361	22	500
Plage	36	65	68	32	17	282	500
Total	502	497	522	510	500	469	3000

Tableau 32 : Résultats détaillés de classification (MLP)

Les taux d'erreur pour les validations croisées sur 4 et 2 groupes étant respectivement de 0,3627 et 0,3767.

6.1.4. SVM

Théoriquement l'algorithme le plus performant, il requiert aussi un certain degré de paramétrage (fonction noyau, paramètres C et gamma…) qui implique donc plusieurs essais avec différentes valeurs et différentes fonctions noyau. Les noyaux polynomiaux donnent des résultats proches des réseaux de neurones, les meilleurs résultats étant obtenus par des noyaux basés sur des RBF. A partir de là, de nombreuses itérations ont alors été nécessaires pour déterminer des paramètres optimaux sachant que les résultats variaient considérablement (jusqu'à 80% d'erreur pour des paramètres mal choisis). Voici le résumé des meilleures performances obtenues avec des SVM :

Tirage	Taux d'erreur
1	0,2983
2	0,3017
3	0,2867
4	0,2900
5	0,2950

Tableau 33 : Taux d'erreur pour chaque tirage (SVM)

Taux d'erreur moyen	0,2943
Résultats par classe	Matrice de confusion

Classe	Rappel	Précision
CDS	0,8460	0,836
Mer	0,6440	0,7436
Montagne	0,6520	0,525
Verdure	0,7720	0,7175
Ville	0,7400	0,8114
Plage	0,5800	0,6502

	CDS	Mer	Montagne	Verdure	Ville	Plage	Total
CDS	423	11	38	7	10	11	500
Mer	23	322	71	8	5	71	500
Montagne	25	21	326	65	13	50	500
Verdure	0	9	71	386	33	1	500
Ville	16	9	24	58	370	23	500
Plage	19	61	91	14	25	290	500
Total	506	433	621	538	456	446	3000

Tableau 34 : Résultats détaillés de classification (SVM)

Les taux d'erreur pour les validations croisées sur 4 et 2 groupes étant respectivement de 0,3030 et 0,3250.

6.1.5. Adaboost ; algorithme de base C4.5

Adaboost a également obtenu de bons résultats dans beaucoup d'applications de classification aussi a-t-on décidé d'évaluer ses performances sur notre problème. Sachant que l'algorithme construit un perceptron à partir de l'ensemble des classificateurs de base il apparait déraisonnable d'utiliser un perceptron multicouches comme algorithme de base, un perceptron simple boosté devrait produire des performances similaires à un perceptron multicouches (ceci a été confirmé par une expérience : le taux d'erreur moyen obtenu était légèrement meilleur : 0,3383 mais globalement les résultats étaient tellement voisins qu'on peut considérer que la différence est due au tirage aléatoire). De la même façon utiliser des SVM avec une fonction noyau linéaire comme algorithme de base donne de bons résultats mais finalement rien de bien meilleur que des SVM avec une fonction noyau plus évoluée (avec ici un taux d'erreur de 0,3120). Nous présentons donc les résultats avec C4.5 puisqu'il s'agit du seul algorithme qui ait réellement bénéficié d'Adaboost, le transformant en un algorithme aux performances très correctes. En se basant sur les expériences faites sur les réseaux de neurones pour la taille de la couche cachée le nombre d'algorithmes de base utilisés a été dimensionné aux environs de 35 avant d'être ajusté expérimentalement.
A titre de confirmation les random forests ont aussi été expérimentées avec des résultats supposés proches, le principe des deux algorithmes étant le même. Sans surprise les résultats se sont révélés très similaires. Voici donc les résultats obtenus :

Tirage	Taux d'erreur
1	0,3700
2	0,3600
3	0,3383
4	0,3783
5	0,3583

Tableau 35 : Taux d'erreur pour chaque tirage (Adaboost)

Taux d'erreur moyen	0,3610
Résultats par classe	Matrice de confusion

Classe	Rappel	Précision
CDS	0,7640	0,729
Mer	0,6400	0,6026
Montagne	0,4640	0,4968
Verdure	0,6840	0,6951
Ville	0,7340	0,7182
Plage	0,5480	0,5768

	CDS	Mer	Montagne	Verdure	Ville	Plage	Total
CDS	382	34	26	3	15	40	500
Mer	25	320	68	8	3	76	500
Montagne	37	66	232	65	41	59	500
Verdure	14	16	60	342	58	10	500
Ville	23	11	26	57	367	16	500
Plage	43	84	55	17	27	274	500
Total	524	531	467	492	511	475	3000

Tableau 36 : Résultats détaillés de classification (Adaboost)

Les taux d'erreur pour les validations croisées sur 4 et 2 groupes étant respectivement de 0,3737 et 0,3970.

6.1.6. Conclusion

On remarque que les algorithmes SVM et MLP se comportent d'une façon très similaire. La classification de type KNN perd en précision sur les classes montagne, plage et mer. Enfin C4.5 affecte tout particulièrement la classe ville, ce qui se retranscrit sur la version améliorée par Adaboost qui affiche par ailleurs de très bonnes performances.

6.2. Récapitulatif des résultats sur tous les algorithmes

Le Tableau 37 résume les résultats obtenus avec l'ensemble des caractéristiques pour tous les algorithmes de classification.

	SVM	MLP	KNN	C4.5
Histogramme basé gradient	34.47	37.77	43.4	50.07
Matrice de Coocurrence	32.93	35.4	38.07	**45.43**
Histogramme de segments	32.27	36.23	39.4	48.4
Matrice de coocurrence de segments	**27.93**	**33.5**	**37.13**	46.63

Tableau 37: Résultats de l'exécution de différents algorithmes pour différentes caractéristiques

Ces résultats sont sans surprise si l'on considère les conclusions de l'expérience sur les caractéristiques de cooccurrence et qu'on les compare aux performances détaillées de chaque caractéristique sur le Tableau 8 : la matrice cooccurrence de segments étant particulièrement efficace sur les classes ville et verdure/forêt, elle se combine assez mal avec C4.5 qui produit de mauvais résultats précisément sur ces classes.

Bibliographie

[1]Y. Chahir, "Indexation et Recherche par le contenu d'informations visuelles", Ecole Centrale de Lyon, 2000.

[2]Y. Chahir and L. Chen, "Efficient Content-Based Image Retrieval Based on Color Homogeneous Objects Segmentation and their Spatial Relationship Characterization", *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, IEEE Computer Society, 2, p. 705, 1999.

[3]M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results", http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

[4]G.A. Miller, C. Fellbaum, R. Tengi, P. Wakefield, H. Langone and B.R. Haskell, "Wordnet: a lexical database for the English language", http://wordnet.princeton.edu/, Cognitive Science Laboratory, Princeton University.

[5]C. Dance, J. Willamowski, L. Fan, C. Bray and G. Csurka, "Visual categorization with bags of keypoints", *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[6]F. Rothganger, S. Lazebnik, C. Schmid and J. Ponce, "Object modeling and recognition using local affine-invariant image descriptors and multi-view spatial contraints", *International Journal of Computer Vision*, 66, 3, 2006.

[7]S. Lazebnik, C. Schmid and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories", *IEEE Conference on Computer Vision & Pattern Recognition*, 2006.

[8]T.S. Lee, "Image Representation Using 2D Gabor Wavelets", *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, 18, 10, p. 959--971, 1996.

[9]B. Alexandre, " Le système visuel humain au secours de la vision par ordinateur", Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), 2007.

[10] F. Wörgötter, N. Krüger, N. Pugeault, D. Calow, M. Lappe, K. Pauwels, M. Van-Hulle, S. Tan and A. Johnston, "Early cognitive vision: Using Gestalt laws for task-dependent, active image processing", *Natural Computing*, 3, p. 293--321, 2004.

[11] A. Desolneux, L. Moisan and J.M. Morel, "From Gestalt Theory to Image Analysis: A Probabilistic Approach", Springer, 2008.

[12] M. Swain and D. Ballard, "Color Indexing", *International Journal of Computer Vision (IJCV)*, 7, 1, p. 11--32, 1991.

[13] T. Lindeberg, "Feature Detection with Automatic Scale Selection", *Int. J. Comput. Vision*, Kluwer Academic Publishers, Hingham, MA, USA, 30, 2, p. 79--116, 1998.

[14] R. Bellman, "Adaptive Control Processes: A Guided Tour", Princeton University Press, 1961

[15] M. Ardebilian and L. Chen, "A New Line Extraction Algorithm: Fast Connective Hough Transform", *proceedings of PRIP'2001*, Informa, p. 127, 2001.

[16] D. Navon, "Forest before trees: The precedence of global features in visual perception", *Cognitive Psychology*, 9, 3, p. 353--383, 1977.

[17] Rapport interne au LIRIS écrit par Clément Metge sous la direction de Mohsen Ardebilian et avec la collaboration d'Alain Pujol. A paraître.

[18] Commission Internationale de l'Eclairage, http://www.cie.co.at/

[19] A. Tremeau, C. Fernandez-Maloigne and P. Bonton, "Image numérique couleur, de l'acquisition au traitement", Dunod, 2004.

[20] M. Unser, "Texture classification and segmentation using wavelet frames", *IEEE Transactions on Image Processing*, 4, 11, p. 1549--1560, Nov 1995.

[21] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, 60, 2, p. 91--110, 2004.
[22] M.A. Stricker and A. Dimai, "Color indexing with weak spatial constraints", *Proc. SPIE Storage and Retrieval for Still Image and Video Databases IV*, 2670, p. 29-40, 1996.
[23] J. Lin, "Divergence measures based on the Shannon entropy", *IEEE Transactions on Information Theory*, 37, 1, p. 145-151, Jan 1991.
[24] M.A. Stricker and M. Orengo, "Similarity of Color Images", *Storage and Retrieval for Image and Video Databases (SPIE)*, p. 381-392, 1995.
[25] Y. Deng, B.S. Manjunath, C. Kenney, M.S. Moore and H. Shin, "An efficient color representation for image retrieval", *IEEE Transactions on Image Processing*, 10, 1, p. 140--147, Jan 2001.
[26] G. Pass and R. Zabih, "Comparing images using joint histograms", *Multimedia Systems*, 7, 3, p. 234--240, 1999.
[27] G. Pass, R. Zabih and J. Miller, "Comparing images using color coherence vectors", *MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia*, p. 65--73, 1996.
[28] J. Huang, S.R. Kumar, M. Mitra, W. Zhu and R. Zabih, "Image Indexing Using Color Correlograms", *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, p. 762, 1997.
[29] M.S. Landy and N. Graham, "Visual Perception of Texture", 2002.
[30] J. Beck, "Similarity Grouping and Peripheral Discriminability under Uncertainty", *The American Journal of Psychology*, 85, 1, p. 1--19, 1972.
[31] H. Tamura, S. Mori and T. Yamawaki, "Textural features corresponding to visual perception", *IEEE Trans. Syst. Man Cybern*, 8, 6, p. 460--473, 1978.
[32] B.S. Manjunath and W.Y. Ma, "Texture Features for Browsing and Retrieval of Image Data", *IEEE Trans. Pattern Anal. Mach. Intell.*, 18, 8, p. 837--842, 1996.
[33] R. M. Haralick, "Statistical and structural approaches to texture", *Proceedings of the IEEE*, 67, 5, p. 786-804, 1979.
[34] J.A. McLaughlin and J. Raviv, "Nth-order autocorrelations in pattern recognition", *Information and Control*, 12, 2, p. 121--142, 1968.
[35] F. Goudail, E. Lange, T. Iwamoto, K. Kyuma and N. Otsu, "Face Recognition System Using Local Autocorrelations and Multiscale Integration", *IEEE Trans. Pattern Anal. Mach. Intell.*, 18, 10, p. 1024--1028, 1996.
[36] Y. Kang, K. Morooka and H. Nagahashi, "Scale Invariant Texture Analysis Using Multi-scale Local Autocorrelation Features.", *Scale-Space*, Springer, Lecture Notes in Computer Science, 3459, p. 363-373, 2005.
[37] S.L. Tanimoto, "An Optimal Algorithm for Computing Fourier Texture Descriptors", *IEEE Trans. Comput*, 27, 1, p. 81--84, 1978.
[38] R. Bajcsy and Lieberman, "Texture gradient as a depth cue", *Computer graphics and image processing*, 5, 1976.
[39] M. E. Jernigan and F. D'Astous, "Entropy-based texture analysis in the spatial frequency domain", *IEEE Trans. Pattern Anal. Mach. Intell*, 6, 2, p 237--243, 1984.
[40] S. E. Grigorescu, N. Petkov and P. Kruizinga, "Comparison of texture features based on Gabor filters", *IEEE Transactions on Image Processing*, 11, 10, p. 1160--1167, 2002.
[41] S. Mallat, "A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis and Its Applications)", Academic Press, Hardcover, 1999.
[42] T. Randen and J. H. Husoy, "Filtering for texture classification: a comparative study", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, 4, p. 291--310, 1999.

[43] J. Mao and A.K. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models", *Pattern Recognition*, 25, 2, p. 173--188, 1992.
[44] G.R. Cross and A.K. Jain, "Markov random field texture models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5, 1, p. 25--39, 1983.
[45] J. Luo and A. Savakis, "Texture-based segmentation of natural images using multiresolution autoregressive models", *IEEE Western New York Image Processing Workshop*, 1998.
[46] J.C. Russ, "Image Processing Handbook, Fourth Edition", CRC Press, Inc., 2002.
[47] I. Bloch, "Information combination operators for data fusion: a comparative review with classification", *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 26, 1, p. 52--67, 1996.
[48] S. Di-Zenzo, "A note on the gradient of a multi-image", *Computer Vision, Graphics and Image Processing*, Academic Press Professional, Inc., 33, 1, p. 116--125, 1986.
[49] J. Canny, "A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 6, p. 679--698, 1986.
[50] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 7, p. 629--639, 1990.
[51] W. Ma and B. Manjunath, "EdgeFlow: a technique for boundary detection and image segmentation", *IEEE Transactions on image processing*, 9, 8, p. 1375--1388, 2000.
[52] S. Sclaroff and A.P. Pentland, "Modal Matching for Correspondence and Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 6, p. 545--561, 1995.
[53] A. Pentland, R.W. Picard and S. Sclaroff, "Photobook: content-based manipulation of image databases", *International Journal of Computer Vision*, 18, 3, p. 233--254, 1996.
[54] V. Caselles, R. Kimmel and G. Sapiro, "Geodesic Active Contours", *International Journal of Computer Vision*, Kluwer Academic Publishers, 22, 1, p. 61--79, 1997.
[55] N. Paragios and R. Deriche, "Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 3, p. 266--280, 2000.
[56] M. Peura and J. Iivarinen, "Efficiency of Simple Shape Descriptors", *3rd International Workshop on Visual Form*, p. 443-451, 1997.
[57] E. Persoon and K.S. Fu, "Shape discrimination using Fourier descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 3, p. 388--397, 1986.
[58] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker, "Query by Image and Video Content: The QBIC System", *Computer*, 28, 9, p. 23--32, 1995.
[59] C. Teh and R.T. Chin, "On Image Analysis by the Methods of Moments", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10, 4, p. 496--513, 1988.
[60] M.E. Celebi and Y.A. Aslandogan, "A Comparative Study of Three Moment-Based Shape Descriptors", *ITCC '05: Proceedings of the International Conference on Information Technology: Coding and Computing*, 1, p. 788--793, 2005.
[61] A. Jain and A. Vailaya, "Image retrieval using color and shape", *Pattern Recognition*, 29, 8, p. 1233--1244, 1996.
[62] A. Vailaya, A. Jain and H.J. Zhang, "On image classification: city images vs. landscapes", *Pattern Recognition*, 31, 12, p. 1921--1935, 1998.
[63] S. Brandt, J. Laaksonen and E. Oja, "Statistical shape features in content-based image retrieval", *Proceedings of the 15th International Conference on Pattern Recognition*, 2, p. 1062--1065, 2000.

[64] K. Mikolajczyk and C. Schmid, "Scale & Affine Invariant Interest Point Detectors", *Int. J. Comput. Vision*, 60, 1, p. 63--86, 2004.
[65] C. Harris and M. Stephens, "A Combined Corner and Edge Detection", *Proceedings of The Fourth Alvey Vision Conference*, p. 147--151, 1988.
[66] Y. Dufournaud, C. Schmid and R. Horaud, "Matching images with different resolutions", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1, p. 612--618, 2000.
[67] F. Li and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories", *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, p. 524--531, 2005.
[68] F. Mindru, T. Moons and L. J.V. Gool, "Recognizing Color Patterns Irrespective of Viewpoint and Illumination", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1, p. 368--373, 1999.
[69] J.J. Koenderink and A.J. van Doom, "Representation of local geometry in the visual system", *Biological Cybernetics*, 55, 6, p. 367--375, 1987.
[70] J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid, "Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study", *International Journal of Computer Vision*, 73, 2, p. 213--238, 2007.
[71] A.E. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, 21, 5, p. 433--449, 1999.
[72] S. Lazebnik, C. Schmid and J. Ponce, "A Sparse Texture Representation Using Local Affine Regions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 8, p. 1265--1278, 2005.
[73] T. Coogan and A. Sutherland, "Transformation Invariance in Hand Shape Recognition", *ICPR 2006: 18th International Conference on Pattern Recognition*, 3, p. 485--488, 2006.
[74] S. L. Dockstader and N. S. Imennov, "Prediction for human motion tracking failures", *IEEE Transactions on Image Processing*, 15, 2, p. 411--421, 2006.
[75] J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner and W. Niblack, "Efficient Color Histogram Indexing for Quadratic Form Distance Functions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 7, p. 729--736, 1995.
[76] Y. Rubner, C. Tomasi and L.J. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval", *International Journal of Computer Vision*, Kluwer Academic Publishers, 40, 2, p. 99--121, 2000.
[77] J. Lin, "Divergence measures based on the Shannon entropy", *IEEE Transactions on Information Theory*, 37, 1, p. 145--151, Jan 1991.
[78] S. Liapis, E. Sifakis and G. Tziritas, "Colour and texture segmentation using wavelet frame analysis, deterministic relaxation, and fast marching algorithms", *Journal of Visual Communication and Image Representation*, 15, 1, p. 1--26, 2004.
[79] X. Haisong and Y. Hirohisa, "Visual evaluation at scale of threshold to suprathreshold color difference", *Color Research and Application*, 30, 3, p. 198--208, 2005.
[80] A.K. Jain, M.N. Murty and P.J. Flynn, "Data clustering: a review", *ACM Computing Surveys*, ACM, 31, 3, p. 264--323, 1999.
[81] R. Xu and D. Wunsch-II, "Survey of clustering algorithms", *IEEE Transactions on Neural Networks*, 16, 3, p. 645--678, 2005.
[82] P.H.A. Sneath and R.R. Sokal, "Numerical Taxonomy", freeman, 1973.
[83] B. King, "Step-wise clustering procedures", *Journal of the American Statistical Association*, 69, p. 86--101, 1967.

[84] A.P. Dempster, N.M. Laird and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, 39, 1, p. 1--38, 1977.
[85] I. Karkkainen and P. Franti, "Dynamic local search for clustering with unknown number of clusters", *Proceedings of the 16th International Conference on Pattern Recognition*, 2, p. 240--243, 2002.
[86] P. Franti and J. Kivijarvi, "Randomised Local Search Algorithm for the Clustering Problem", *Pattern Analysis and Applications*, 3, 4, p. 358--369, 2000.
[87] H. Frigui and R. Krishnapuram, "Clustering by Competitive Agglomeration", *PR*, 30, 7, p. 1109--1119, 1997.
[88] H. Frigui and R. Krishnapuram, "A Robust Competitive Clustering Algorithm With Applications in Computer Vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, 21, 5, p. 450--465, 1999.
[89] B.L. Saux and N. Boujemaa, "Unsupervised robust clustering for image database categorization", *Proceedings of the 16th International Conference on Pattern Recognition*, 1, p. 259--262, 2002.
[90] T. Kohonen, "Self-organized formation of topologically correct feature maps", MIT Press, p. 509--521, 1988.
[91] T. Martinetz and K. Schulten, "A Neural-Gas Network Learns Topologies", *Artificial Neural Networks*, 1, p. 397--402, 1991.
[92] T. M. Martinetz, S. G. Berkovich and K. J. Schulten, "Neural-gas" network for vector quantization and its application to time-series prediction", *IEEE Transactions on Neural Networks*, 4, 4, p. 558--569, 1993.
[93] B. Fritzke, "A growing neural gas network learns topologies", *Advances in Neural Information Processing Systems 7*, MIT Press, p. 625--632, 1995.
[94] F. Marques, "Multiresolution Image Segmentation Based on Compound Random Fields: Application to Image Coding", Universitat Politècnica de catalunya, 1992.
[95] D. Martin, C. Fowlkes, D. Tal and J. Malik, "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics", *Proceedings of the 8th International Conference on Computer Vision*, 2, p. 416--423, 2001.
[96] T. Poggio and C. Koch, "Ill-Posed Problems in Early Vision: From Computational Theory to Analogue Networks", *Proceedings of the Royal Society of London. Series B, Biological Sciences*, The Royal Society, 226, 1244, p. 303--323, 1985.
[97] L. G. Shapiro R. M. Haralick, "Image Segmentation Techniques", *Computer vision, graphics, and image processing*, Academic Pres, 29, 1, 1985.
[98] J. Benois and D. Barba, "Image segmentation by region-contour cooperation for image coding", *Proceedings of the 11th IAPR International Conference on Image, Speech and Signal Analysis*, 3, p. 331--334, 1992.
[99] X. Jie and S. Peng-fei, "Natural color image segmentation", *in Proceedings of International Conference on Image Processing ICIP*, p. 973--976, 2003.
[100] S. Sclaroff and A.P. Pentland, "Modal Matching for Correspondence and Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, 17, 6, p. 545--561, 1995.
[101] V. Caselles, R. Kimmel and G. Sapiro, "Geodesic Active Contours", *International Journal of Computer Vision*, Kluwer Academic Publishers, 22, 1, p. 61--79, 1997.
[102] S.C. Zhu, T.S. Lee and A.L. Yuille, "Region competition: unifying snakes, region growing, energy/Bayes/MDL for multi-band image segmentation", *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, IEEE Computer Society, p. 416, 1995.

[103]J. M. Chassery and C. Garbay, "An Iterative Segmentation Method Based on a Contextual Color and Shape Criterion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 6, p. 794--799, 1984.
[104]S. L. Horowitz and T. Pavlidis, "Picture Segmentation by a Directed Split and Merge Procedure", *Proceedings of the International Conference on Pattern Recognition*, p. 424--433, 1974.
[105]L. Priese and V. Rehrmann, "A Fast Hybrid Color Segmentation Method", *DAGM-Symposium*, p. 297--304, 1993.
[106]H. Digabel and C. Lantuejoul, "Iterative algorithms", *Proceedings of the 2nd European Symposium on Quantitative Analysis of Microstructures in Material Science and Medicine*, Riederer Verlag, p. 85--99, 1977.
[107]C. Lantuejoul, "La squelettisation et son application aux mesures topologiques des mosaiques polycristallines", Ecole des Mines de Paris, 1978.
[108]J.B.T.M. Roerdink and A. Meijster, "The Watershed Transform: Definitions, Algorithms and Parallelization Strategies", *Fundamenta Informaticae*, Polish Mathematical Society, 41, p. 187--228, 2000.
[109]R. Ohlander, K. Price and D.R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method", *Computer Graphics and Image Processing*, p. 313--333, 1978.
[110]Y. Ohta, T. Kanade and T. Sakai, "Color information for region segmentation", *Computer Graphics and Image Processing*, p. 222--241, 1980.
[111]C. Zhang and P. Wang, "A New Method of Color Image Segmentation Based on Intensity and Hue Clustering", *ICPR '00: Proceedings of the International Conference on Pattern Recognition*, IEEE Computer Society, p. 3617, 2000.
[112]J. Fauqueur and N. Boujemaa, "Region-based retrieval: coarse segmentation with fine color signature", *Proceedings of the IEEE International Conference on Image Processing*, p. 609--612, 2002.
[113]Y.W. Lim and S.U. Lee, "On the color image segmentation algorithm based on the thresholding and the fuzzy C-means techniques", *Pattern Recognition*, Elsevier Science Inc., 23, 9, p. 935--952, 1990.
[114]A. W. C. Liew, S. H. Leung and W. H. Lau, "Fuzzy image clustering incorporating spatial continuity", *IEE Proceedings - Vision, Image and Signal Processing*, 147, 2, p. 185--192, 2000.
[115]P. Lambert and H. Grecu, "A quick and coarse color image segmentation", *Proceedings of the IEEE International Conference on Image Processing*, 1, p. 965--968, 2003.
[116]S. Pateux, "Spatial segmentation of color images according to the MDL formalism", *Proceedings of the IEEE International Conference on Image Processing*, 2, p. 92--95, 2000.
[117]A. Pujol and L. Chen, "Color Quantization For Image Processing Using Self Information", *Proceedings of the IEEE international Conference on Information Communications and Signal Processing (ICICS)*, 2007.
[118]A. Pujol and L. Chen, "Coarse Adaptive Color Image Segmentation for Visual Object Classification", *Proceedings of the 15th International Conference on Systems, Signals and Image Processing*, 2008.
[119]J.P. Braquelaire and L. Brun, "Comparison and Optimization of Methods of Color Image Quantization", *IEEE Transactions on Image Processing*, 6, 7, p. 1048--1052, 1997.
[120]P. Scheunders, "A comparison of clustering algorithms applied to color image quantization", *Pattern Recognition Letters*, Elsevier Science Inc., 18, 11-13, p. 1379--1384, 1997.

[121] J. Ketterer, J. Puzicha, M. Held, M. Fischer, J.M. Buhmann and D.W. Fellner, "On Spatial Quantization of Color Images", *ECCV '98: Proceedings of the 5th European Conference on Computer Vision*, Springer-Verlag, 1, p. 563--577, 1998.
[122] P. Heckbert, "Color image quantization for frame buffer display", *SIGGRAPH Computer Graphics*, ACM, 16, 3, p. 297--307, 1982.
[123] X. Wu, "Color quantization by dynamic programming and principal analysis", *ACM Transactions on Graphics*, 11, 4, p. 348-372, 1992.
[124] N. Papamarkos, A. Atsalakis and C.P. Strouthopoulos, "Adaptive Color Reduction", *IEEE Transactions on systems man and cybernetics*, 32, 1, p. 44--56, 2002.
[125] A. Atsalakis, N. Papamarkos, N. Kroupis, D. Soudris and A. Thnailakis, "Colour quantization technique based on image decomposition and its embedded system implementation", *Proceedings of the IEE conference on Vision, Image and Signal Processing*, 151, 6, p. 511--524, 2004.
[126] J. Astola, P. Haavisto and Y. Neuvo, "Vector median filters", *Proceedings of the IEEE*, 78, 4, p. 678--689, 1990.
[127] P.V.C. Hough, "Methods and means for recognizing complex patterns", U.S. patent, 3, 069,654, 1962.
[128] V.F. Leavers, "Which Hough transform ?", *CVGIP: Image Understanding*, Academic Press, Inc., 58, 2, p. 250--264, 1993.
[129] J. Illingworth and J. Kittlet, "A survey of the Hough transform", *Computer Vision Graphics and Image Processing*, 44, p. 87--116, 1988.
[130] A. Pujol and L. Chen, "Hough Transform Based Cityscape Classifier", *6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2005.
[131] A. Pujol and L. Chen, "Line Segment Based Edge Features Using Hough transform", *The 7th IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP)*, 2007.
[132] N.I. Fisher, "Statistical Analysis of Circular Data", Cambridge University Press, 4, 1993.
[133] H. Fu, A. Pujol, E. Dellandréa and L. Chen, "Region based visual object categorization using segment features and polynomial image modeling", *7th International Workshop on Statistical Pattern Recognition*, 2008..
[134] T. Leung and J. Malik, "Recognizing Surfaces Using Three-Dimensional Textons", *ICCV '99: Proceedings of the International Conference on Computer Vision*, IEEE Computer Society, 2, p. 1010, 1999.
[135] I. Ulusoy and C.M. Bishop, "Generative versus Discriminative Methods for Object Recognition", *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*,Vol. 2, IEEE Computer Society, p. 258--265, 2005.
[136] A. Vailaya, M. Figueiredo, A. Jain and H.J. Zhang, "Content-based hierarchical classification of vacation images", *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1, p. 518--523, Jul 1999.
[137] Y. Wu, B.L. Tseng and J.R. Smith, "Ontology-based multi-classification learning for video concept detection", *In Proceedings of IEEE International Conference on Multimedia and Expo*, p. 1003--1006, 2004.
[138] D. Forsyth, J. Malik, M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson and C. Bregler, "Finding Pictures of Objects in Large Collections of Images", University of California at Berkeley, Berkeley, CA, USA, 1996.
[139] A. Vailaya, "Semantic classification in image databases", Michigan State University, Adviser: Anil K. Jain, 2000.

[140]P. Grünwald, "A Tutorial Introduction to the Minimum Description Length Principle", *Advances in Minimum Description Length: Theory and Applications*, MIT Press, 2005.
[141]A.B. Torralba and A. Oliva, "Semantic Organization of Scenes Using Discriminant Structural Templates", *ICCV '99: Proceedings of the International Conference on Computer Vision*, IEEE Computer Society, 2, p. 1253, 1999.
[142]S. Ullman, E. Sali and M. Vidal-Naquet, "A Fragment-Based Approach to Object Representation and Classification", *IWVF-4: Proceedings of the 4th International Workshop on Visual Form*, Springer-Verlag, p. 85--102, 2001.
[143]A. McCallum and K. Nigam, "A comparison of event models for Naive Bayes text classification", *In AAAI-98 Workshop on Learning for Text Categorization*, AAAI Press, p. 41--48, 1998.
[144]K. Nigam, J. Lafferty and A. McCallum, "Using maximum entropy for text classification", *In IJCAI-99 Workshop on Machine Learning for Information Filtering*, p. 61--67, 1999.
[145]F. Perronnin and C. Dance, "Fisher Kernels on Visual Vocabularies for Image Categorization", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 1--8, 2007.
[146]F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition", *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV 2005)*, 1, p. 604--610, 2005.
[147]K. Barnard, P. Duygulu, R. Guru, P. Gabbur and D.A. Forsyth, "The effects of segmentation and feature choice in a translation model of object recognition", *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2003.
[148]K. Barnard, P. Duygulu, N.d. Freitas, D. Forsyth, D. Blei and M. Jordan, "Matching words and pictures", *Journal of Machine Learning Research*, MIT Press, 3, p. 1107--1135, 2003.
[149]M.C. Burl, M. Weber and P. Perona, "A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry", *ECCV '98: Proceedings of the 5th European Conference on Computer Vision*, Springer-Verlag, 2, p. 628--641, 1998.
[150]S. Ioffe and D.A. Forsyth, "Probabilistic Methods for Finding People", *International Journal of Computer Vision*, Kluwer Academic Publishers, 43, 1, p. 45--68, 2001.
[151]C. Papageorgiou and T. Poggio, "Trainable pedestrian detection", *International Conference on Image Processing*, 4, p. 35--39, 1999.
[152]P. Felzenszwalb, D. Mcallester and D. Ramanan, "A Discriminatively Trained, Multiscale, Deformable Part Model", *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June, 2008.
[153]O. Chum and A. Zisserman, "An Exemplar Model for Learning Object Classes", *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
[154]C.G.M. Snoek, M. Worring and A.W.M. Smeulders, "Early versus late fusion in semantic video analysis", *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, ACM, p. 399--402, 2005.
[155]S. Nowozin, "libsift - Scale-Invariant Feature Transform implementation", http://user.cs.tu-berlin.de/~nowozin/libsift/, 2005.
[156]M. Marszalek, C. Schmid, H. Harzallah and J.v.d. Weijer, "Learning Object Representations for Visual Object Class Recognition", *Visual Recognition Challange workshop, in conjunction with ICCV*, 2007.
[157]B.Johansson, "A survey on: Contents based search in image databases", Technical report, Linköping University, Department of Electrical Engineering, http://www.isy.liu.se/cvl/Projects/VISIT-bjojo/, 2000.

[158] R. C. Veltkamp and M. Tanase, "Content-Based Image Retrieval Systems: A Survey", Technical report, Utrecht University, Department of Computing Science, http://www.aa-lab.cs.uu.nl/cbirsurvey/cbir-survey/, 2001.