

Optimisation du dictionnaire pour la quantification vectorielle d'images couleur sous contraintes psychophysiques

Quantization codebook optimization for color image under psychophysical constraints

par **Christophe CHARRIER, Hocine CHERIFI**

Université Jean Monnet, Equipe Ingénierie de la Vision – UMR CNRS 5516
3, rue Javelin Pagnon BP 505, 42007 Saint-Etienne Cedex 01 – FRANCE
Phone : +33 477 923030 – Fax : +33 477 923039
{charrier|cherifi}@univ-stetienne.fr

résumé et mots clés

L'information contenue dans une image couleur est redondante spatialement, spectralement et perceptuellement. Dans le contexte de la compression d'image couleur par quantification vectorielle, cette redondance devient rapidement un handicap en termes de performances (complexité, qualité, taux de compression). En combinant des éléments de perception et de classification, il est possible d'améliorer la qualité du dictionnaire tout en réduisant le temps de construction associé. Dans cet article, nous proposons une méthode de réduction de la base d'apprentissage. En associant aux éléments de cette base une mesure perceptuellement significative, nous en extrayons un sous-ensemble représentatif. Une étape de classification reposant sur un modèle paramétrique de la mesure est ensuite appliquée sur ce sous-ensemble. Pour chacune des classes obtenues, un dictionnaire est déterminé en appliquant l'algorithme LBG. L'ensemble de ces dictionnaires constitue le dictionnaire final. L'utilisation de tests psychophysiques et de mesures statistiques de la qualité nous ont permis de valider notre approche en termes de temps de calcul, de qualité des images reconstruites et du taux de compression.

Compression d'image couleur, classification, qualité, indice de pertinence, échantillon aléatoire.

abstract and key words

The information contained into an image is spatially, spectrally and perceptually redundant. In the context of the color Vector Quantization (VQ) compression scheme, this redundancy is a handicap in terms of performances (complexity, quality and compression rate). Combining perceptual and classification criteria, one can improve the codebook quality when reduce construction time associated. In this paper, we propose a training set reduction method. Associating to vectors of the training set a perceptually relevant measure, one can extract a subset from it. A classification step is then applied on this subset. Finally, the codebook is construct using the LBG algorithm on each obtained cluster, and joining together all code vectors. Psychophysics and statistical measures of image quality allow us to validate our method in terms of construction time, reconstructed image quality and compression rate.

Color image compression, classification, quality, measure of relevance, random sample

1. introduction

L'objectif d'un algorithme de codage d'image avec perte d'informations est d'exploiter au mieux la redondance d'ordre statistique et perceptuelle présente dans une image. L'exploitation de cette redondance doit permettre de déterminer le plus petit nombre de bits utilisés pour représenter l'image tout en conservant une qualité visuelle « acceptable » de l'image décompressée. Ce type d'algorithme de compression doit alors effectuer un compromis entre la qualité de reconstruction des images et le taux de compression associé. De plus, lors de l'étape de compression, on note que plus les données à encoder sont de nature complexe et plus le temps nécessaire à cet encodage est important. La majeure partie du traitement est en effet consacrée à la recherche de la table des codes. Ce coût peut devenir un réel handicap lorsque l'on effectue la compression en très haute résolution, *i.e.* avec une très haute qualité de reconstruction des données et plus particulièrement pour les images couleur.

Parmi les algorithmes de compression avec perte d'informations, la méthode de compression par Quantification Vectorielle (QV) a déjà montré ses performances pour la compression de la parole et de l'image [1, 2].

De nombreux travaux [3, 4, 5] ont été réalisés pour obtenir le meilleur compromis possible en se basant sur la corrélation spatiale des blocs sur des images à niveaux de gris. Cependant pour une image couleur, il existe également des corrélations spectrales entre les composantes couleur (R, V, B). Ceci accentue alors encore plus le problème du temps consacré à la recherche de la table des codes, et apporte une contrainte supplémentaire au problème du compromis qualité/taux de compression.

Nos travaux effectués dans le cadre de la compression d'image couleur par la QV, apportent une solution à ces problèmes en faisant coopérer les approches statistiques et les approches perceptuelles afin de minimiser la redondance spatiale et spectrale.

Le but de ce travail est d'optimiser la qualité de reproduction des images couleur compressées par quantification vectorielle. Cet objectif peut être atteint en optimisant l'espace de représentation et la quantification. Cette optimisation conjointe est aujourd'hui encore un problème ouvert relativement complexe. Nous avons donc choisi de travailler dans le domaine spatio-colorimétrique. En effet, dans le domaine spatial les stratégies pour former des vecteurs sont relativement simples; les pixels adjacents sont fortement corrélés, et la quantification vectorielle est plus performante. Dans les domaines transformés, outre les corrélations entre canaux, il existe également des corrélations inter-bandes et intra-bandes. La stratégie optimale pour former des vecteurs n'est pas évidente. De plus, la relation qualité/quantification des coefficients est difficile à contrôler [6, 7]. En effet, dans ce cas les erreurs sont distribuées non seulement spatialement mais aussi spectralement et interagissent de manière complexe. Néanmoins, la technique mise au point peut être transposée dans le domaine fréquentiel ou spatio-fréquentiel afin de permettre une utilisation

de la quantification vectorielle sur des coefficients d'ondelettes, par exemple. Il est alors nécessaire de vérifier que les ajustements perceptuels des paramètres sont encore valables. Si tel n'est pas le cas, un nouvel ajustement doit alors être opéré. Pour ce faire, la méthodologie d'analyse développée dans cet article peut être utilisée.

Lors du codage de l'information par QV, la première étape repose sur la construction de la table des codes, appelée *dictionnaire*, à partir de la *base d'apprentissage* formée de vecteurs, appelés *vecteurs à coder*. L'algorithme proposé par Linde, Buzo et Gray [8] est la technique la plus couramment utilisée pour construire le dictionnaire. Cette méthode est connue sous le terme algorithme LBG ou algorithme de Lloyd généralisé. L'idée directrice est le partitionnement de la base d'apprentissage en N classes. Chacune des classes est alors représentée par son barycentre appelé aussi *vecteur code*. Cependant, même si cet algorithme est très simple à mettre en œuvre, les images décompressées présentent un certain nombre de dégradations très facilement identifiables. Ainsi, les deux principaux artefacts sont l'apparition de structures de blocs et une perte des couleurs par rapport à l'image originale (figure 1). La figure 1b illustre les inconvénients dus à l'apparition des structures de blocs. En effet, plus la taille des blocs est importante et plus cette structure de blocs est visible. Ce premier artefact est principalement lié à la décomposition de l'image originale en vecteurs de $(n \times p)$ pixels, traités indépendamment les uns des autres. La figure 1c illustre le phénomène de la perte des couleurs. En effet plus le taux de compression est élevé et plus on observe une perte des couleurs lorsqu'on utilise des blocs de petite taille (pour éviter l'apparition des structures de blocs). Ce second artefact est lié à l'effet moyenneur de l'algorithme. En effet, le nuage couleur représentatif de l'image à encoder est divisé en N classes. Chacune de ces N classes est ensuite représentée par son barycentre. Il est alors possible de noter l'apparition de couleurs qui ne sont pas présentes dans le nuage couleur de l'image originale. La continuité colorimétrique entre deux vecteurs juxtaposés de l'image peut alors disparaître.

Diverses recherches ont été menées pour améliorer le partitionnement de la base d'apprentissage. Les travaux de Orchard [3] ont permis de construire un algorithme efficace de recherche du plus proche voisin. Cet algorithme est basé sur l'inégalité triangulaire de la distance euclidienne. L'idée est d'utiliser quelques points de référence et de calculer la distance d_j entre ces points et les vecteurs code existants. Ensuite, la distance entre le vecteur à coder et chacun des points de référence est établie. La comparaison entre ces différentes distances permet alors d'éliminer de nombreux vecteurs code candidats au plus proche voisin sans avoir à recalculer toutes les distances. Cette méthode permet ainsi d'obtenir un partitionnement de la base plus rapidement. Cependant, l'effet moyenneur de l'algorithme LBG reste conservé.

Huang *et al.* [4] ont introduit une contrainte supplémentaire sur la recherche des vecteurs code. Tout d'abord la distance entre chaque vecteur à coder et le vecteur code est calculée. La recherche du meilleur vecteur code se fait sur une zone restreinte de l'espace des

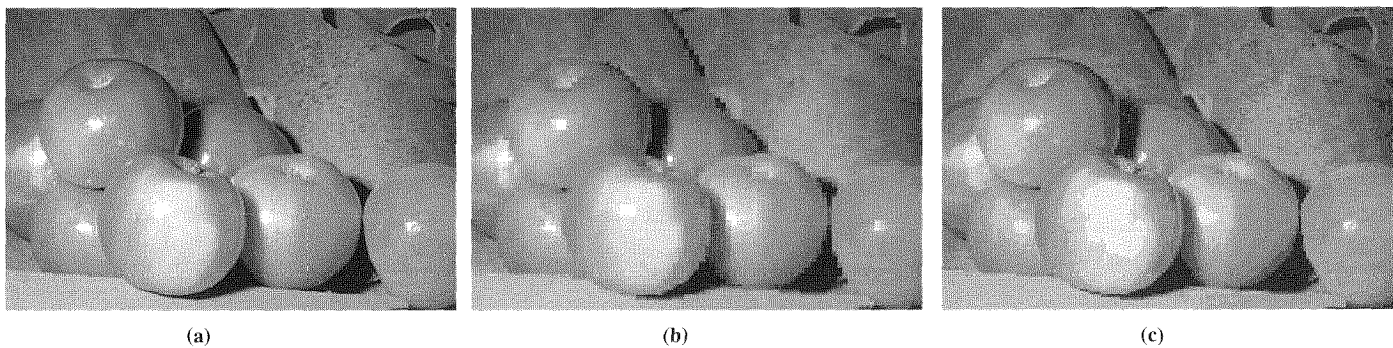


Figure 1. – Illustration des deux principaux problèmes inhérents à la technique de compression par quantification vectorielle. (a) Image originale, (b) Apparition de structures de blocs sur les contours lors de l'utilisation de blocs de taille importante, (c) Perte des couleurs à fort taux de compression, avec des blocs de petite taille.

données définies autour du vecteur code. De ce fait, le temps de calcul du dictionnaire se trouve accéléré puisque la recherche des vecteurs code ne s'effectue pas sur toute la base d'apprentissage. Cependant, la même technique de définition des vecteurs code est conservée, et donc, le problème de l'effet moyenneur également.

Monawer [5] a proposé une version de l'algorithme LBG reposant sur un traitement des vecteurs code à chaque itération. Ce traitement consiste à remplacer les vecteurs code calculés par leur représentant le plus proche appartenant à l'image originale. Les dégradations dues aux couleurs n'appartenant pas à l'image sont ainsi résolues mais le problème de perte de couleur demeure. En effet, les vecteurs code sont toujours détectés en utilisant l'algorithme LBG. Ce procédé ne permet pas d'annuler totalement l'effet moyenneur de cet algorithme, même s'il permet quelque peu de l'atténuer.

Les travaux précédents n'apportent qu'une solution relative à la définition des vecteurs code, en termes de perception visuelle. Dans [9] nous avons proposé une solution qui permet une meilleure définition des vecteurs code. En effet, dans tous les travaux précédents, les nouveaux vecteurs code sont calculés à partir des anciens, en y introduisant une perturbation aléatoire. Par contre, dans ces travaux les nouveaux vecteurs code sont définis à l'aide de la règle du Max-Min. Les éléments les plus éloignés de tous les vecteurs code existants sont recherchés pour en définir de nouveaux. Cet algorithme possède alors l'avantage d'annuler l'effet moyenneur de la recherche des vecteurs code. Cependant, la recherche des nouveaux vecteurs code est très coûteuse en temps de calcul.

Dans tous les travaux existants, on remarque que la définition des vecteurs code repose sur une classification plus ou moins adéquate des vecteurs à coder. L'idée d'une combinaison de la QV et de la classification semble alors naturelle. En effet, ces deux méthodes utilisent des techniques communes de recherche de classes. Le but d'une telle combinaison est alors d'incorporer une information de classification dans les vecteurs code en les triant pendant la construction du dictionnaire. En combinant la QV et une classification de bas niveau, certaines caractéristiques de l'image peuvent être regroupées dans une classe lors du processus de compression. Une

telle combinaison peut être utilisée pour améliorer le rendu d'une région (appartenant à une classe spécifique) de l'image décompressée. Ce type de processus de compression permet d'obtenir de meilleurs résultats en comparaison d'une utilisation séparée des algorithmes de compression et de classification [10, 11, 12].

Dans cet article, nous proposons d'utiliser une telle combinaison pour apporter une solution au compromis qualité/taux de compression/temps de calcul. Nous proposons une méthode de construction du dictionnaire basée sur une réduction de la base d'apprentissage suivie d'une étape de classification paramétrique du sous-ensemble obtenu.

En général, le dictionnaire est construit à partir d'une base d'apprentissage qui contient des informations redondantes. Dans une première étape, afin de réduire cette redondance et d'accélérer le temps de calcul du dictionnaire il est possible de rechercher un échantillon représentatif de la base d'apprentissage sous la contrainte qu'il ait la même distribution que la base originale et qu'il soit de taille inférieure. Cela revient à rechercher un échantillon $X[x_1, \dots, x_n]$ de variables aléatoires indépendantes et de même loi parente. Si cette recherche est relativement rapide dans le cas de variables aléatoires scalaires, le procédé devient plus complexe dans le cas multidimensionnel. Nous réduisons alors ce problème de nature vectorielle à un problème « scalaire ». Nous associons à chaque vecteur de la base d'apprentissage un scalaire représentatif du point de vue perceptuel. Dans cette situation, réduire la base d'apprentissage vectorielle revient à réduire la base d'apprentissage scalaire. Nous construisons alors un échantillon représentatif de même distribution que la distribution de la base d'apprentissage scalaire.

Dans une deuxième étape, nous utilisons un processus de classification des données pendant la construction du dictionnaire. Nous déterminons le dictionnaire à partir de l'échantillon représentatif. Lors de la construction du dictionnaire, nous utilisons une technique de classification basée sur un modèle de lois mélange de Gaussiennes, car cette modélisation permet de représenter les distributions empiriques observées pour les images réelles [13, 14]. Une fois que la préclassification de l'échantillon scalaire a été effectuée, nous pouvons en déduire une préclassification de

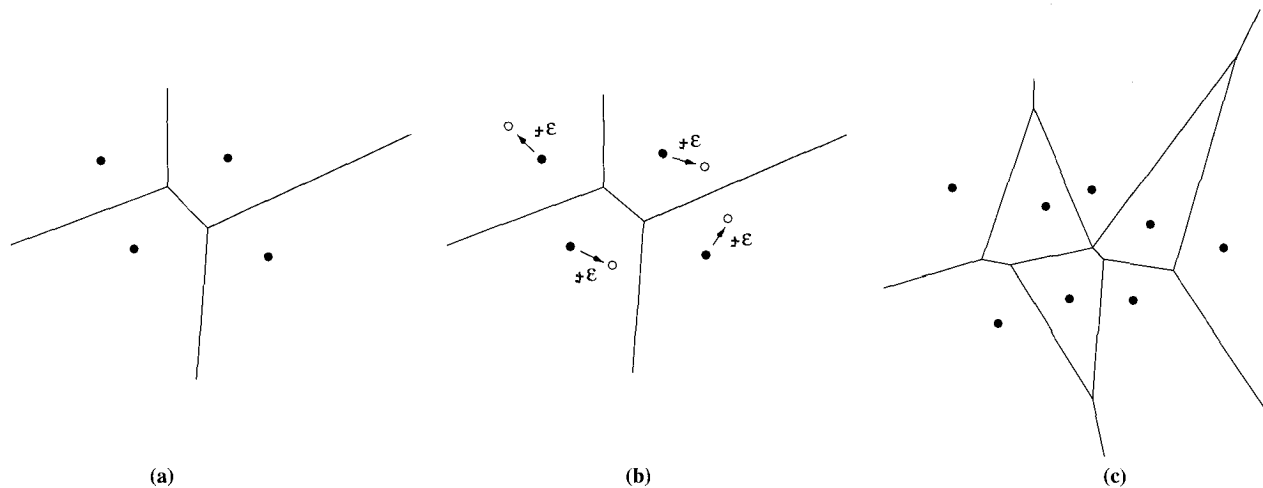


Figure 2. – Principe de l'algorithme LBG. (a) Positions des barycentres à l'itération m , (b) Construction des germes à partir des barycentres, (c) Position des nouveaux barycentres à l'itération $(m + 1)$.

l'échantillon vectoriel. Nous appliquons alors l'algorithme LBG sur chacune des classes vectorielles de manière à définir le dictionnaire.

Dans la section 2 de cet article, nous présentons et commentons la technique de construction du dictionnaire par l'algorithme LBG. Ensuite, nous proposons la méthode de génération du dictionnaire basée sur un « sous-échantillonnage » de la base d'apprentissage et une préclassification de cet échantillon. Puis dans la section 3, nous abordons les problèmes liés à la définition de critères mesurant la qualité d'une image. Dans la section 4 enfin, nous présentons les résultats expérimentaux qui ont conduit à la sélection de l'indice d'activité le plus pertinent. Nous présentons également les résultats issus de la comparaison entre l'algorithme LBG et la technique proposée, en terme de qualité d'image.

2. construction du dictionnaire par l'algorithme LBG

Avant d'envisager la construction du dictionnaire, il est nécessaire de décomposer l'image originale en un ensemble de groupes de pixels appelés aussi *vecteur*. Chacun des vecteurs $(\vec{x}_i)_{i \in [1, \dots, N]}$ ainsi formé contient donc une composante spatiale (nombre de pixels) et une composante spectrale (composantes couleur). Cet ensemble \mathcal{B} de vecteurs est communément appelé *base d'apprentissage*, de taille N . Par définition, la compression par QV sélectionne un vecteur d'entrée \vec{x} de dimension k et le remplace par un vecteur \vec{y} de même dimension appartenant à un ensemble fini \mathcal{D} (*dictionnaire*) de *vecteurs code*. La construction du dictionnaire est l'étape cruciale de la QV, puisqu'elle conditionne les performances de la compression en terme de qualité.

L'algorithme LBG est la méthode de construction du dictionnaire la plus couramment utilisée en compression par QV. Cette méthode est une combinaison de l'itération de Lloyd [15] (ou encore algorithme des K -moyennes) et d'une étape de division. Le dictionnaire initial est réduit à un vecteur code \vec{y}_0 qui correspond au barycentre de la base d'apprentissage totale. Puis ce vecteur code \vec{y}_0 est décomposé en un germe $\vec{y}_0 \pm \vec{\epsilon}$, où $\vec{\epsilon}$ est un vecteur aléatoire de perturbation de faible énergie. L'itération de Lloyd est ensuite appliquée sur ce germe ainsi que sur le vecteur code \vec{y}_0 pour définir deux nouveaux vecteurs code \vec{y}_1 et \vec{y}_2 . Ce processus est réitéré jusqu'à ce que la taille désirée du dictionnaire soit atteinte.

L'algorithme LBG permet de créer des dictionnaires dont la taille augmente, jusqu'à obtenir la taille désirée (à chaque nouvelle itération, la taille du dictionnaire est multipliée par deux). On remarque facilement que cette technique de construction des vecteurs code engendre la notion classique d'effet moyenneur des couleurs.

La figure 2 récapitule schématiquement les principes de l'algorithme LBG. Dans la figure 2a, les barycentres calculés à l'itération m sont symbolisés par les points noir. Au début de l'itération suivante, ces barycentres subissent une perturbation $\vec{\epsilon}$. On obtient alors de nouveaux germes (figure 2b). On applique sur ces germes l'itération de Lloyd de façon à définir les nouveaux barycentres (figure 2c).

La complexité de cet algorithme est définie en fonction des paramètres du dictionnaire \mathcal{D} (*i.e.* la dimension k des vecteurs code et la taille M de \mathcal{D} ainsi que la taille N de la base d'apprentissage). À la dernière itération l , $k.M.N.l$ multiplications ont été nécessaires pour définir les nouveaux germes. Le calcul des nouveaux barycentres a nécessité $k.M.l$ multiplications. Ainsi, plus la taille du dictionnaire est importante et plus le temps nécessaire à l'étape de construction du dictionnaire est élevé.

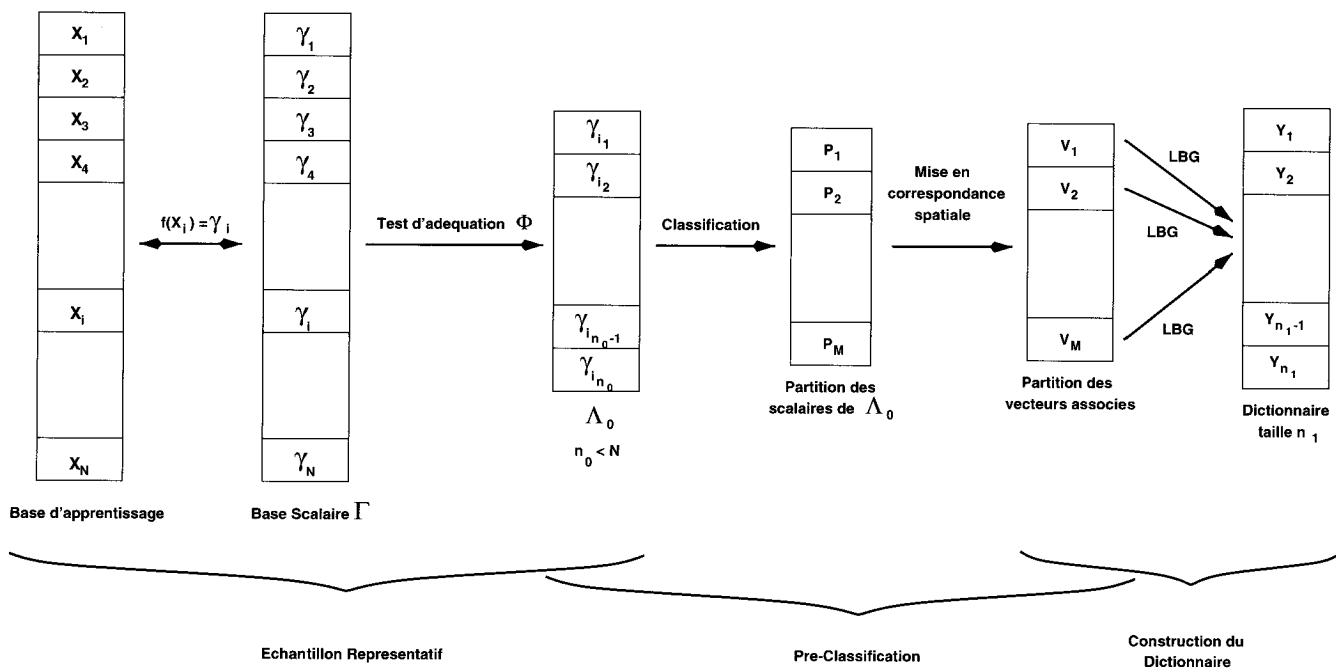


Figure 3. – Récapitulatif schématique de la méthode proposée.

3. construction du dictionnaire par l'algorithme LBG-SPE

Comme nous l'avons mentionné précédemment, l'algorithme LBG possède deux handicaps majeurs : le premier est inhérent au temps de calcul, tandis que le deuxième est lié à la définition des vecteurs représentatifs.

La figure 3 présente de manière schématique l'algorithme LBG-SPE (pour algorithme LBG Sous-échantillonné et Préclassifié). Cet algorithme se décompose en trois phases principales. La première permet d'extraire un échantillon représentatif de la base \mathcal{B} , la deuxième effectue une classification des éléments de l'échantillon pour améliorer la qualité du dictionnaire, et le dictionnaire est construit en fonction de la classification obtenue dans une troisième étape.

Nous allons maintenant détailler les différentes phases de notre algorithme.

3.1. échantillon représentatif

La recherche d'un échantillon représentatif doit nous permettre de réduire le temps nécessaire à la construction du dictionnaire. En effet, l'algorithme LBG est un algorithme « statistique » qui ne repose que sur la loi des données. Dès lors, en utilisant un

échantillon de même loi, nous obtenons les mêmes performances, en terme de qualité, pour une rapidité accrue.

Les vecteurs code déterminés par l'algorithme LBG standard reposent sur la distribution des données appartenant à la base d'apprentissage. Or, ces données sont présentes dans la base d'apprentissage de façon redondante. Nous recherchons alors un échantillon qui possède la même distribution que la base, dans l'optique de préserver la qualité du dictionnaire, tout en réduisant le nombre d'opérations.

3.1.1. recherche de l'échantillon représentatif

Les vecteurs $(\vec{x}_i)_{i \in [1, \dots, N]}$ de la base \mathcal{B} sont multidimensionnels (dimension spatiale et spectrale). Ceci a pour effet d'augmenter de manière prohibitive le temps de recherche de l'échantillon représentatif. Pour réduire ce coût, nous déterminons pour chaque vecteur $(\vec{x}_i)_{i \in [1, \dots, N]} \in \mathcal{B}$ une mesure de pertinence $(\gamma_i)_{i \in [1, \dots, N]}$ qui caractérise chacun des vecteurs.

À partir de ces mesures, il est alors possible de générer un ensemble de scalaires définis de la façon suivante :

$$\Gamma = \{\gamma_i \in \mathbb{R} : \gamma_i = f(\vec{x}_i), \quad \forall i \in [1, \dots, N]\}. \quad (1)$$

où f est une transformation qui permet d'associer une mesure de pertinence γ_i à chacun des vecteurs $(\vec{x}_i)_{i \in [1, \dots, N]}$.

Le figure 4 illustre la manière dont on passe effectivement d'un vecteur à un scalaire, et inversement.

A partir de cet ensemble Γ , nous recherchons un échantillon aléatoire Λ de taille n telle que $n < N$. La sélection de

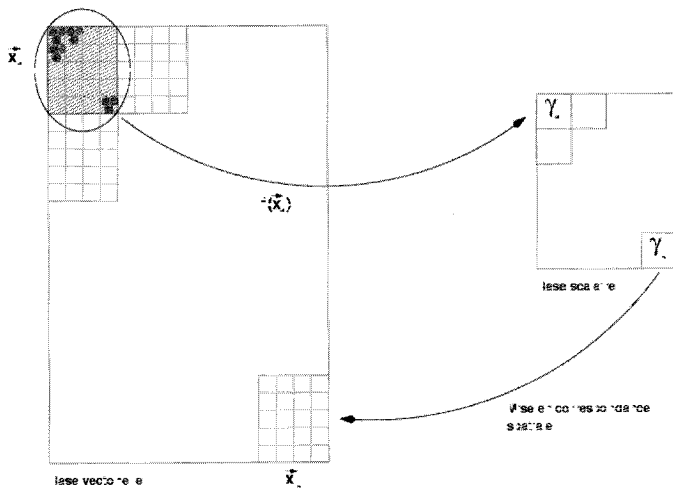


Figure 4. – Passage d'un vecteur à un scalaire, et vice-versa.

cet échantillon s'effectue alors en fonction de la fonction de répartition de l'ensemble Γ , *i.e.* la fonction de répartition de Λ doit être la même que celle de Γ . Pour obtenir un tel résultat, nous utilisons un test statistique d'homogénéité Φ permettant de déterminer la taille optimale n_0 de l'échantillon aléatoire Λ_0 . Par taille optimale, on sous-entend la plus petite taille de Λ_0 que l'on puisse obtenir via le test Φ . Ce test d'homogénéité est défini comme suit :

$$\Phi : \mathcal{P}(\mathbb{R}) \times \mathcal{P}(\mathbb{R}) \longrightarrow \mathbb{R}^+ \quad (2)$$

$$(A, B) \longrightarrow \Phi(A, B)$$

où $\mathcal{P}(\mathbb{R})$ représente une partie de \mathbb{R} .

Soit $\mathcal{F}_{\Gamma, \beta}$ défini par :

$$\mathcal{F}_{\Gamma, \beta} = \{\Lambda \in \mathcal{P}(\mathbb{R}) \mid \Delta\Gamma, \Phi(\Gamma, \Lambda) < \beta\}. \quad (3)$$

où β est un seuil réel fixé. Nous admettons que l'hypothèse $\mathcal{F}_{\Gamma, \beta} \neq \emptyset$ est vérifiée pour ce qui suit.

La taille n_0 optimale de l'échantillon Λ_0 , sous la condition du test Φ est déterminée par :

$$\Lambda_0 \in \mathcal{F}_{\Gamma, \beta} \text{ tel que } \forall \Lambda \in \mathcal{F}_{\Gamma, \beta}, \text{card}(\Lambda_0) \leq \text{card}(\Lambda). \quad (4)$$

où $\text{card}(X)$ représente le cardinal de l'ensemble X et

$$\Lambda_0 = \{\gamma_{i_j} \in \Gamma \forall j \in [1, \dots, n_0]\}. \quad (5)$$

3.1.2. le test d'homogénéité

L'hypothèse d'homogénéité peut se formuler ainsi : deux échantillons aléatoires (x_1, \dots, x_n) et (y_1, \dots, y_k) sont homogènes si la distribution de la variable aléatoire parente X est la même que celle de la variable aléatoire parente Y .

En général, la méthodologie des tests d'hypothèse demande de construire d'abord une statistique, *i.e.* une fonction de données

que l'on comparera à un seuil. Cependant, dans la plupart des situations la densité de probabilité des données n'est pas connue. C'est pourquoi il est préférable d'utiliser des tests indépendants de la loi des données (tests non paramétriques). En effet, les tests non paramétriques ne font aucune hypothèse sur la distribution de la population.

Parmi les méthodes non paramétriques d'homogénéité classiques, sont à notre disposition le test de Kolmogorov-Smirnov, le test de Wilcoxon et le test du χ^2 . Nous utilisons le test de Kolmogorov-Smirnov car il est particulièrement efficace lorsque l'on veut juger de la ressemblance de deux fonctions de distribution [16].

Pour réaliser le test de Kolmogorov-Smirnov, il faut donc tout d'abord trier l'ensemble des valeurs des deux échantillons en ordre croissant. On peut alors définir la fonction de répartition empirique de chacune des deux variables aléatoires.

Dans notre étude, nous avons considéré uniquement le cas où la base Γ est une série de variables aléatoires indépendantes $(\gamma_i)_{i \in [1, \dots, N]}$ de fonction de répartition \hat{F}_N . Puis, nous émettons l'hypothèse que l'histogramme cumulé de Γ représente la fonction de répartition de la base. Dans ce cas, l'hypothèse d'homogénéité est la suivante : les deux variables Γ et Λ suivent-elles la même densité de probabilité?

L'échantillon est sélectionné par le test de Kolmogorov-Smirnov défini par :

$$D = \text{Max} \left(|\hat{F}_N(\gamma) - \hat{F}_n(\gamma)| \right) \leq \beta_n, \quad (6)$$

où β_n est une fonction tabulée représentant le niveau de signification du test, N représente la taille de la base d'apprentissage initiale, n indique la taille du sous-échantillon aléatoire et \hat{F}_N est la fonction de distribution de Γ . Le [tableau 1] décrit les valeurs du seuil β_n du test en fonction de divers niveaux de signification.

Tableau 1. – Valeur critique de β_n en fonction de divers niveaux de signification [17].

Niveau de signification	Valeurs critiques
10%	$\beta_n \sqrt{n} = 1.22$ pour $n > 35$
20%	$\beta_n \sqrt{n} = 1.07$ pour $n > 35$

Le processus de recherche de la taille optimale n_0 de l'échantillon Λ_0 s'effectue à l'aide d'une technique de dichotomie sur la taille n de Λ chaque fois que l'hypothèse d'homogénéité est validée. Nous construisons à chacune des itérations la fonction de répartition $\hat{F}_n(\gamma)$ associée à Λ et nous appliquons le test d'homogénéité. Ce processus est itéré tant que l'hypothèse n'est pas rejetée.

Le résultat du test est indépendant de la répartition spatiale des données. Le choix des éléments du sous-échantillon peut être opéré de différentes façons. Dans la pratique, celui-ci est obtenu en faisant une recherche séquentielle des éléments. La seule condition à respecter est de conserver les proportions de la distribution originale.

Le test de Kolmogorov-Smirnov nous permet de déterminer un échantillon de taille minimale. Le seuil β_n , qui représente le niveau de signification du test, est un critère purement statistique. Or ce test est appliqué, il est vrai, pour vérifier l'homogénéité de l'échantillon représentatif, mais dans un contexte de compression d'image. De ce fait, on peut légitimement s'interroger sur l'évolution de la fonction β_n non plus en terme statistique mais en terme de perception. En fait, cela revient à s'interroger sur l'évolution de ce seuil en fonction de la perception visuelle. C'est pourquoi la valeur critique β_n que nous avons utilisée dans toutes les expériences a été perceptuellement ajustée.

Une fois que le sous-échantillon représentatif Λ_0 est déterminé, nous lui appliquons un algorithme de préclassification.

3.2. l'étape de préclassification

Cette étape de classification de l'échantillon représentatif Λ_0 doit nous permettre d'obtenir une meilleure reproduction des vecteurs code du dictionnaire et ainsi d'améliorer la qualité des images.

3.2.1. principe de la préclassification

Lors de l'étape du sous-échantillonnage de la base d'apprentissage Γ nous avons déterminé un échantillon représentatif Λ_0 de taille n_0 . Nous appliquons sur ce sous-ensemble, un algorithme de classification paramétrique utilisé pour les modèles de lois mélange. Nous obtenons en sortie de cet algorithme une partition $(P_i)_{i \in [1, \dots, M]}$ de réels définie par :

$$P_i = \{\gamma_i \in I_i\} \quad (7)$$

où $\bigcup_{i \in [1, \dots, M]} I_i = \{1, \dots, n_0\}$ et $I_i \cap I_j = \emptyset$ si $i \neq j$ et $|I_i| \neq 0$.

A chaque partition P_i est associée la probabilité p_i .

La partition est obtenue par une étape d'estimation des paramètres du modèle, suivie d'un processus de classification selon la règle du maximum *a posteriori*.

3.2.2. modèle de lois mélange de gaussiennes

Classiquement, une loi mélange est utilisée lorsque la nature exacte des composants est inconnue. L'étude des lois mélange de tout type (exponentielle, gaussienne, etc.) a été considérée depuis très longtemps dans la littérature. Ce modèle a été utilisé avec succès dans de nombreux domaines d'applications (biologie, médecine, image, etc.). L'idée sous-jacente est que le phénomène étudié est le résultat d'un nombre fini de causes élémentaires de même loi.

Pour caractériser la loi de l'échantillon représentatif Λ_0 , nous utilisons, dans cet article, un mélange fini de lois gaussiennes monodimensionnelles. Cette loi mélange monodimensionnelle est

alors :

$$h(x) = \sum_{j=1}^M p_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp \left[- \left(\frac{x - \mu_j}{\sigma_j} \right)^2 \right], \quad (8)$$

où les probabilités p_j vérifient $0 < p_j < 1$, et, où (μ_j, σ_j) représentent les paramètres de la loi à estimer.

Ainsi, l'image peut être définie par un ensemble de paramètres indépendamment de leur position géométrique :

$$\Phi = \{(p_j, \mu_j, \sigma_j) : j \in [1, \dots, M]\}. \quad (9)$$

3.2.3. estimation des paramètres

Les expressions des estimateurs sont non linéaires rendant de ce fait impossible leur résolution de manière analytique. Plusieurs méthodes itératives ont été utilisées pour l'estimation de ces paramètres (méthode de Newton, méthode du gradient, etc.).

Dans cet article, nous utilisons l'un des algorithmes itératifs les plus couramment utilisés, *i.e.* une version stochastique de l'algorithme dû à Dempster *et al.* [18] introduite en 1977. Il s'agit de l'algorithme SEM (Stochastic Expectation Maximization). Cet algorithme proposé par Celeux et Diebolt [19] pour améliorer les performances de l'algorithme de Dempster *et al.* a été utilisé dans un contexte images [20, 21, 22].

SEM est une variation de EM dans lequel une étape stochastique a été introduite entre l'étape E et M. Cette nouvelle étape permet de s'affranchir de la dépendance de la position initiale des paramètres et de converger vers la solution plus rapidement. Les perturbations utilisées dans l'étape S permettent d'affecter aléatoirement un individu à une classe en fonction de sa probabilité.

Le principe d'un tel algorithme, pour le modèle utilisé, est le suivant : à partir de valeurs initiales $\Phi^{(0)} = \{(p_j^{(0)}, \mu_j^{(0)}, \sigma_j^{(0)}) : \forall j \in [1, \dots, M]\}$, l'itération m est définie comme suit :

– **Etape E** : Calcul des probabilités *a posteriori* $t^{(m)}(j/x_i)$ que l'échantillon x_i appartienne à la classe P_j :

$$\left. \begin{array}{l} \forall i \in [1, \dots, n_0] \\ \text{et } \forall j \in [1, \dots, M] \end{array} \right\} t^{(m)}(j/x_i) = \frac{p_j^{(m-1)} \frac{1}{\sqrt{2\pi}\sigma_j^{(m-1)}} \exp \left[- \left(\frac{x_i - \mu_j^{(m-1)}}{\sigma_j^{(m-1)}} \right)^2 \right]}{\sum_{j'=1}^M p_{j'}^{(m-1)} \frac{1}{\sqrt{2\pi}\sigma_{j'}^{(m-1)}} \exp \left[- \left(\frac{x_i - \mu_{j'}^{(m-1)}}{\sigma_{j'}^{(m-1)}} \right)^2 \right]} \quad (10)$$

– **Etape S** : Affectation aléatoire de l'échantillon x_i à une des classes $\Phi^{(m)} = (P_j^{(m)})_{j \in [1, \dots, M]}$ selon la loi des $t^{(m)}(j/x_i)_{j \in [1, \dots, k]}$.

– **Étape M** : Calcul des estimés $(p_j^{(m)}, \mu_j^{(m)}, \sigma_j^{(m)})$ du maximum de vraisemblance, à partir de la partition obtenue :

$$\forall j \in [1, \dots, M] \begin{cases} p_j^{(m)} = \frac{\text{card}(P_j^{(m)})}{n}, \\ \mu_j^{(m)} = \frac{1}{\text{card}(P_j^{(m)})} \sum_{x_i \in P_j^{(m)}} x_i, \\ \sigma_j^{(m)} = \frac{1}{n} \sum_{j=1}^k \sum_{x_i \in P_j^{(m)}} \|x_i - \mu_j^{(m+1)}\|^2. \end{cases} \quad (11)$$

Ce processus est itéré jusqu'à convergence du log de la vraisemblance L :

$$L = \log \left(\prod_{i=1}^{n_0} \sum_{j=1}^M h(x_i, \Phi) \right). \quad (12)$$

La classification finale est le résultat de la dernière étape S.

Cet algorithme nécessite la connaissance *a priori* du nombre de classes. Dans la pratique, nous avons fixé une valeur maximale du nombre de classes ($M_{\text{Max}} = 8$). Nous avons par ailleurs également fixé la taille minimale d'une classe ($0.05 \times n_0$). En deçà de cette valeur, le nombre de classes est décrémenté d'une unité. L'algorithme est alors réinitialisé. Ceci permet d'avoir une classification non supervisée.

Dans nos expérimentations, le nombre maximal de classes n'a jamais été conservé. On observe que le nombre final de classes (M) varie entre 6 et 7.

3.3. la construction du dictionnaire

L'algorithme de construction du dictionnaire LBG est optimal (au sens du calcul des vecteurs code) lorsqu'il est appliqué sur un ensemble de variables aléatoires de loi gaussienne [8]. L'algorithme SEM permet de décomposer un échantillon de loi mélange de gaussiennes en diverses classes gaussiennes. Ainsi en construisant sur chacune des classes un dictionnaire par l'algorithme LBG, nous nous situons dans un contexte optimal pour la mesure de pertinence.

Les classes $(P_j)_{j \in [1, \dots, M]}$ sont constituées des mesures de pertinence $(\gamma_i)_{i \in [1, \dots, n_0]}$ calculées à partir des vecteurs de la base d'apprentissage \mathcal{B} . Nous sommes capables de construire la classe vectorielle $(V_j)_{j \in [1, \dots, M]}$ associée à chacune des classes scalaires $(P_j)_{j \in [1, \dots, M]}$ (cf. figure 4). Nous appliquons l'algorithme LBG sur chacune des classe V_j de façon à déterminer les vecteurs code. La taille du dictionnaire \mathcal{D}_j associé à la classe P_j est déterminée par la relation :

$$\text{card}(\mathcal{D}_j) = p_j \times n_1, \forall j \in [1, \dots, M]. \quad (13)$$

La dernière étape consiste à faire la réunion de tous ces ensembles de vecteurs code pour construire le dictionnaire final :

$$\mathcal{D} = \bigcup_{j=1}^M \mathcal{D}_j. \quad (14)$$

4. mesures de la qualité

L'évaluation de la qualité d'une image est une étape indispensable en compression d'image. Le problème d'une telle évaluation réside dans la caractérisation des différences entre les images.

Si l'utilisation de mesures objectives mathématiques pour l'évaluation de la qualité des images est très largement répandue, elle n'est pas nécessairement satisfaisante d'un point de vue purement perceptuel. En effet, dans l'état actuel des connaissances, l'être humain reste pour l'instant le meilleur juge de la qualité des images à l'issue de la chaîne de compression/décompression. La mise en œuvre d'expérimentations psychophysiques pour juger de la qualité est ainsi de plus en plus répandue [23, 24, 25]. Afin d'évaluer notre algorithme, nous nous appuyons sur ces deux démarches d'appréciation de la qualité. Nous avons donc mené parallèlement des expérimentations basées sur des mesures objectives et psychophysiques de la qualité.

4.1. mesures objectives

Les mesures objectives que nous utilisons pour déterminer la qualité des images sont : le PSNR et une mesure de qualité basée sur une combinaison de facteurs de distorsions appelée PQS couleur introduite par Kotani *et al.* [26].

4.1.1. le PSNR

L'une des mesures couramment utilisée est le rapport signal sur bruit crête – *peak signal to noise ratio* – connu sous l'étiquette PSNR. Elle mesure la distorsion qui existe entre deux images en utilisant la formule suivante :

$$\text{PSNR} = 10 \log_{10} \sum_{(x,y)} \left(\frac{(g(x,y) - f(x,y))^2}{D} \right). \quad (15)$$

où D représente l'amplitude maximale du signal, $f(x,y)$ représente la couleur du pixel de coordonnées (x,y) de l'image de référence et $g(x,y)$ représente la couleur pixel de coordonnées (x,y) de l'image ayant subi un traitement.

Bien que cette mesure de distorsion soit utilisée très fréquemment en traitement d'image comme mesure de qualité, elle n'est, cependant, basée sur aucun des principes inhérents au système visuel humain. Il est tout à fait possible d'obtenir des valeurs proches du PSNR pour deux images compressées à un même



(a) Image compressée par QV (TC = 35 :1, PSNR = 16.4 dB)



(b) Image compressée par JPEG (TC = 35 :1, PSNR = 16.4 dB).

Figure 5. – Images décodées de qualité quantitative (PSNR) approximativement identique mais de qualité visuelle différente pour un taux de compression de 35 :1.

taux par deux techniques différentes, alors qu'une des deux images (figure 5b) est jugée de qualité moindre par un observateur humain.

4.1.2. le PQS couleur

Pour déterminer les performances des deux algorithmes sur la qualité de reproduction des images, nous avons également utilisé une mesure de qualité basée sur une combinaison de facteurs de distorsions : l'échelle de qualité couleur – *color Picture Quality Scale (PQS)* – que nous appellerons dans la suite PQS couleur. Cette mesure a été introduite pour définir une échelle de qualité de l'image. Quatre facteurs (F_i) $_{1 \leq i \leq 4}$ sont définis :

- F_1 : Indice de distorsion globale de l'image,
- F_2, F_3 : Indices de distorsion induite par les textures,
- F_4 : Indice de distorsion relative aux contours de l'image.

Les valeurs de ces quatre facteurs sont déterminées en plusieurs étapes. Tout d'abord, les valeurs (R, V, B) de l'image originale et de l'image reconstruite sont converties dans l'espace couleur $Y, r - g, y - b$ [27]. Ces nouvelles composantes couleur sont

pondérées par les réponses spectrales du cortex visuel humain [28]. Ensuite, chaque nouveau triplet est transformé en un triplet (Hue, Value, Chroma) où Hue, Value et Chroma sont les trois composantes de l'espace colorimétrique de Munsell. Finalement, on calcule les différences couleur entre les triplets (H, V, C) selon la définition de distance colorimétrique introduite par la CIE [29]. Ainsi, les quatre facteurs de distorsions sont des mesures pondérées de la dégradation des images.

Pour obtenir la mesure PQS couleur, les quatre facteurs sont calculés à partir d'un grand nombre d'images reconstruites. Une Analyse en Composantes Principales est ensuite appliquée sur cet ensemble des données. Puis, une projection des données sur les différents axes principaux est appliquée. Une régression linéaire multiple est finalement effectuée entre ces nouvelles valeurs et les résultats issus de la moyenne des notes de qualité attribuées par des observateurs à chacune des images reconstruites. Ce procédé est identique à celui utilisé pour le calcul de l'échelle de qualité des images à niveaux de gris [30, 31].

Cette mesure peut se caractériser par une « mesure objective de qualité subjective des images ». Cependant, même si cette mesure donne des résultats satisfaisants, elle ne permet pas de caractériser les différences locales de qualité. Or, l'être humain est très sensible aux dégradations locales lorsqu'il juge la qualité globale d'une image.

Pour tenir compte de cette observation, nous sommes amenés à définir des tests psychophysiques de qualité.

4.2. mesures subjectives

Les mesures psychophysiques de qualité permettent de définir le seuil de détection des différences entre deux images. Ce seuil caractérise la sensibilité d'un observateur. La recherche de ce seuil fait appel à des éléments de la théorie de la détection.

Afin de déterminer la sensibilité de l'observateur, nous avons utilisé la technique du choix forcé [32].

La mesure de sensibilité d' est alors déduite par l'estimation des probabilités de détection ($p(D)$) et de fausse alarme ($p(FA)$) qui permettent de mesurer l'acuité de l'observateur. La détection consiste à déclarer que le stimulus est présent lorsqu'il est observé; la fausse alarme représente le fait de décider que le stimulus est présent alors qu'il n'est pas observé. Cette mesure de sensibilité d' représente simplement la distance entre la moyenne de la distribution de $\langle LBG \rangle$ et de la distribution $\langle LBG-SPE \rangle$. Il est toutefois utile de noter que cette sensibilité d' caractérise uniquement la paire de stimuli et non la méthode. Sous l'hypothèse que le test est non biaisé, la mesure de la sensibilité d' peut s'exprimer sous la forme suivante [32] :

$$d' = \frac{1}{\sqrt{2}} [z(p(D)) - z(p(FA))] \quad (16)$$

où $z(\cdot)$ représente la fonction inverse de la distribution normale.

5. résultats expérimentaux

Avant d'entamer la comparaison entre l'algorithme LBG-SPE et l'algorithme LBG, nous allons tout d'abord définir le cadre de notre étude.

5.1. environnement expérimental

Nous avons choisi trois taux de compression différents correspondant chacun respectivement à trois classes de qualité de reproduction : très bonne, bonne et moyenne. Ces taux de compression sont : 1 :10, 1 :20 et 1 :30. Notre base est composée de 20 images couleur dont la « complexité » varie d'une image de texture à une image de synthèse.

Les images sont transformées dans l'espace $L^*a^*b^*$ pour utiliser la distance euclidienne comme distance lors de la construction du dictionnaire dans cet espace. Cette distance définie par la CIE – *Commission Internationale de l'Eclairage* – est définie par [29] :

$$\Delta E_{ab}^* = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C_{ab}^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{k_H S_H}\right)^2}, \quad (17)$$

où

– S_L, S_C, S_H sont des fonctions de pondérations qui permettent un ajustement du calcul de la différence colorimétrique, de manière à prendre en compte la variation de l'amplitude de la différence perçue par rapport à la variation de couleur dans l'espace $L^*a^*b^*$.

– k_L, k_C, k_H sont des facteurs paramétriques qui permettent de compenser les variations des composantes de différence de couleurs perçues, en fonction des variations dues aux conditions expérimentales.

Des travaux antérieurs ont montré que cet espace de représentation est le plus performant en terme de qualité d'image, pour les taux de compression testés [31, 33]. La base d'apprentissage a été créée en décomposant l'image originale en vecteurs de pixels de dimension 4×4 afin d'exploiter les corrélations spatiales existantes entre des pixels voisins [34].

Lors des tests psychophysiques, les images sont affichées sur un écran SUN CRT piloté par une carte graphique de type GC14/SX. La résolution spatiale de cet écran est de 1152 par 900 pixels avec un codage des couleurs de 8 bits par canal. Il est nécessaire de pouvoir afficher une image avec un contrôle précis de la luminance, du contraste, etc. afin de ne pas biaiser l'expérimentation psychophysique. Nous avons donc déterminé une table de correction gamma pour corriger la réponse non linéaire du moniteur [35]. Le moniteur utilisé nécessite la correction gamma suivante : $R_\gamma = 2.6, V_\gamma = 2.5, B_\gamma = 2.5$.

5.2. choix d'une mesure de pertinence

La fonction f définie dans la section 3.1.1 permet d'associer à un vecteur à coder, une mesure de pertinence scalaire.

5.2.1. définition des mesures de pertinence

La base de scalaires Γ a été générée à partir de mesures de pertinence des vecteurs à coder. Nous avons utilisé la mesure couleur moyenne définie comme la norme euclidienne du vecteur moyenne couleur \hat{m} :

$$\hat{m} = \|\hat{x}\| = \sqrt{(\hat{m}_R)^2 + (\hat{m}_V)^2 + (\hat{m}_B)^2}, \quad (18)$$

où $(\hat{m}_i)^2$ représente la moyenne de chacun des canaux R, V et B.

Nous avons également utilisé la norme euclidienne du vecteur écart-type comme mesure de l'écart-type couleur $\hat{\sigma}$:

$$\hat{\sigma} = \|\hat{\sigma}\| = \sqrt{(\hat{\sigma}_R)^2 + (\hat{\sigma}_V)^2 + (\hat{\sigma}_B)^2}, \quad (19)$$

où $(\hat{\sigma}_i)^2$ représente la variance sur chacun des canaux R, V et B. Cette mesure représente le contraste du vecteur.

Nous avons également testé une combinaison des moments d'ordre trois et quatre qui représentent respectivement l'asymétrie δ et le kurtose ξ du vecteur. L'asymétrie d'un vecteur est mesuré par :

$$\hat{\delta} = \|\hat{\delta}\| = \sqrt{(\hat{\delta}_R)^2 + (\hat{\delta}_V)^2 + (\hat{\delta}_B)^2}, \quad (20)$$

où $(\hat{\delta}_i)^2 = \hat{\mu}_3(\cdot)/\hat{\sigma}^3(\cdot)$ représente l'asymétrie sur chacun des canaux R, V et B. $\hat{\mu}_3(\cdot)$ est défini, pour chacun des canaux, par :

$$\hat{\mu}_3(\cdot) = \frac{1}{\text{card}(\vec{x}(\cdot))} \sum_{i=1}^{\text{card}(\vec{x}(\cdot))} (x_i(\cdot) - \hat{m}(\cdot))^3.$$

où i représente les pixels constituant le vecteur $\vec{x}(\cdot)$. $\hat{\delta}$ représente l'asymétrie de l'histogramme.

De même, le kurtose est défini par :

$$\hat{\xi} = \|\hat{\xi}\| = \sqrt{(\hat{\xi}_R)^2 + (\hat{\xi}_V)^2 + (\hat{\xi}_B)^2}, \quad (21)$$

où $(\hat{\xi}_i)^2 = \hat{\mu}_4(\cdot)/\hat{\sigma}^4(\cdot)$ représente le kurtose sur chacun des canaux R, V et B. $\hat{\mu}_4(\cdot)$ est défini, pour chacun des canaux, par :

$$\hat{\mu}_4(\cdot) = \frac{1}{\text{card}(\vec{x}(\cdot))} \sum_{i=1}^{\text{card}(\vec{x}(\cdot))} (x_i(\cdot) - \hat{m}(\cdot))^4.$$

où i représente les pixels constituant le vecteur $\vec{x}(\cdot)$. $\hat{\xi}$ représente la concentration de l'histogramme.

De plus, nous avons testé une cinquième mesure de pertinence $\hat{\eta}$ définie comme une combinaison linéaire de la couleur moyenne \hat{m} et de l'écart-type $\hat{\sigma}$:

$$\hat{\eta} = \alpha \hat{m} + (1 - \alpha) \hat{\sigma}, \quad 0 \leq \alpha \leq 1. \quad (22)$$

Nous avons créé les images correspondant aux cinq différents indices de pertinence définis en section 5.2.1. En effet, l'extraction de l'échantillon est effectué à partir de ces images de pertinences.

La figure 6 illustre les différents résultats obtenus en utilisant ces indices sous forme d'images. Nous pouvons remarquer que seules deux des quatre images représentant les valeurs des indices sont visuellement pertinentes. Il s'agit de la figure 6b et de la figure 6c qui représentent respectivement l'image obtenue à partir du calcul de la moyenne du vecteur \hat{m} et celle obtenue par l'écart-type $\hat{\sigma}$. Dans l'image issue de la moyenne \hat{m} , nous distinguons nettement les zones homogènes tandis que la deuxième porte l'information sur les contours des objets de l'image. Les deux autres figures (figure 6d et 6e) n'apportent pas, d'un point de vue détail visuel, d'informations supplémentaires par rapport aux deux images précédentes. Toutefois, on note que l'information la plus intéressante est celle contenue dans la figure 6f. Nous observons une représentation de l'image en régions homogènes avec une préservation des contours des objets. Cette image a été générée en utilisant une combinaison linéaire de la moyenne \hat{m} du vecteur et de l'écart-type $\hat{\sigma}$ associé.

La valeur du paramètre $\alpha = 0.6$ a été déterminée en utilisant des test psychophysiques. La valeur obtenue correspond à l'optimum choisi par les individus dans une procédure de choix forcé.

De manière à sélectionner le meilleur indice de pertinence, nous les comparons selon un critère objectif et subjectif de distorsion.

5.2.2. mesures objectives

Nous avons utilisé la mesure de qualité définie par le PSNR pour sélectionner le meilleur indice de pertinence. La figure 7 illustre le PSNR des images décodées en fonction des indices utilisés et du taux de compression. Toutes les images reconstruites ont été traitées en utilisant une valeur critique $\beta_n \sqrt{n} = 1.22$ en tant que seuil du test de Kolmogorov-Smirnov. Cette valeur de $\beta_n \sqrt{n}$ correspond à un taux d'erreur acceptable de 10%. Nous remarquons facilement que pour les trois taux de compression testés (1 :10, 1 :20 et 1 :30), l'indice de pertinence η donne de meilleurs résultats que les autres indices.

Pour confirmer les résultats obtenus par la mesure du PSNR, nous avons effectué des tests psychophysiques sur la qualité des images reconstruites.

5.2.3. mesures psychophysiques

Pour mener nos tests, nous avons utilisé la même valeur critique que celle employée lors de la mesure de distorsion par le PSNR. L'observateur doit classer par ordre décroissant de qualité les images issues des cinq traitements représentant les diverses mesures de pertinence. Les sujets possèdent tous une vision normale des couleurs.

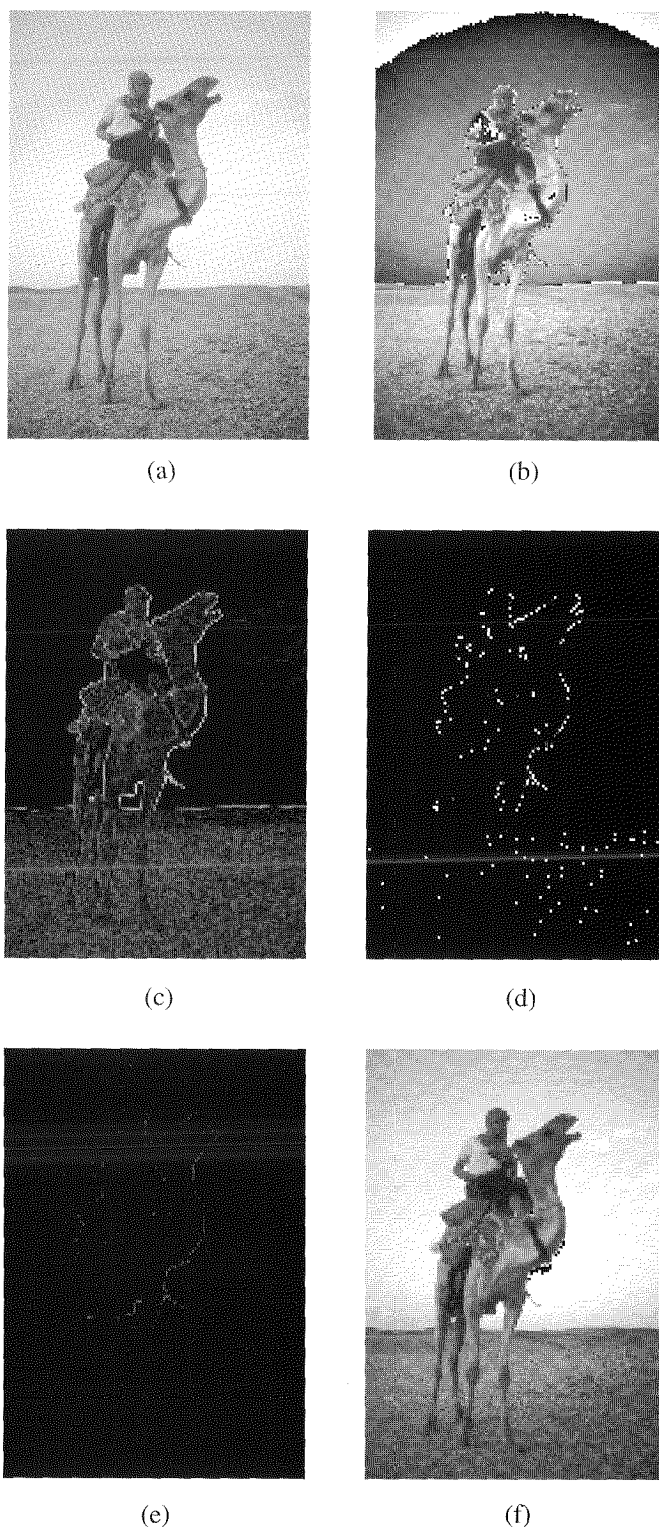


Figure 6. – Images des différents indices de pertinence (redimensionnées à la taille de l'image originale). (a) Image originale, (b) Image de l'indice de pertinence \hat{m} , (c) Image de l'indice de pertinence $\hat{\sigma}$, (d) Image de l'indice de pertinence $\hat{\delta}$, (e) Image de l'indice de pertinence $\hat{\xi}$, (f) Image de l'indice de pertinence $0.6\hat{m} + 0.4\hat{\sigma}$.

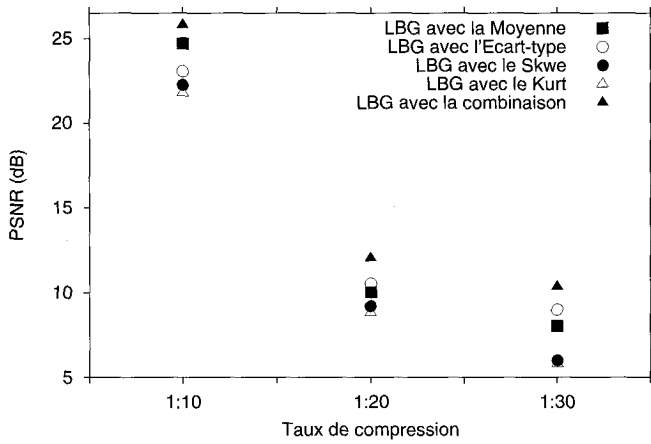


Figure 7. – Evolution du PSNR des indices de pertinence suivant la valeur critique $\beta_n\sqrt{n} = 1.22$.

La figure 8 représente les résultats obtenus pour les tests psychophysiques. Pour un taux de compression égal à 1 :10 (figure 8a), les images codées avec l'indice de pertinence $\hat{\eta}$ ont une meilleure qualité avec un pourcentage de 21%. La deuxième position revient aux images compressées via l'indice moyenne \hat{m} avec un pourcentage de 19.7%. Ces différences ne sont néanmoins statistiquement pas significatives.

Le classement est conservé lorsque le taux de compression augmente (figure 8b et figure 8c). De plus, nous remarquons que la différence de pourcentage de classement augmente avec les niveaux de compression. Au taux de compression de 1 :20, la méthode proposée utilisant l'indice de pertinence $\hat{\eta}$ donne la meilleure qualité d'image avec un pourcentage de 48%, alors que la deuxième méthode (algorithme avec \hat{m}) n'atteint que 14%. Lorsque le taux de compression est égal à 1 :30, le pourcentage entre la première et la deuxième position varie de 68% (algorithme avec $\hat{\eta}$) à 9% (algorithme avec \hat{m}).

Les images compressées par la technique développée en se basant sur l'indice de pertinence $\hat{\eta}$ ont une qualité, au décodage, meilleure que celle obtenue avec les autres indices testés.

Nous utilisons donc l'algorithme LBG-SPE basé sur la mesure de pertinence $\hat{\eta}$ pour effectuer la comparaison avec l'algorithme LBG standard.

5.3. évaluation de la qualité

Avant de commencer la comparaison de l'algorithme LBG-SPE avec l'algorithme LBG standard, il est nécessaire d'ajuster la valeur du seuil $\beta_n\sqrt{n}$ selon des critères visuels.

5.3.1. ajustement perceptuel de la valeur critique $\beta_n\sqrt{n}$

Nous avons vu que la valeur critique $\beta_n\sqrt{n}$ utilisée dans le test de Kolmogorov-Smirnov est un indice du niveau à partir duquel

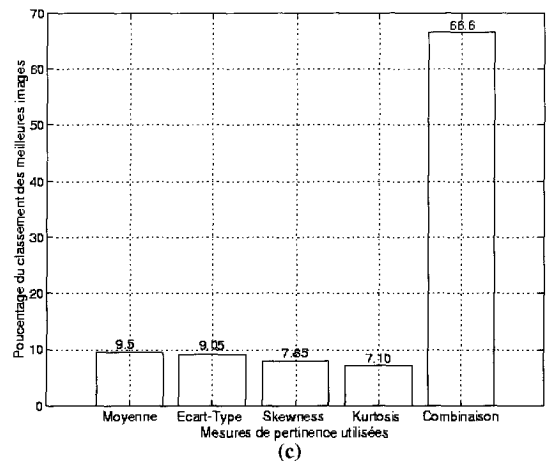
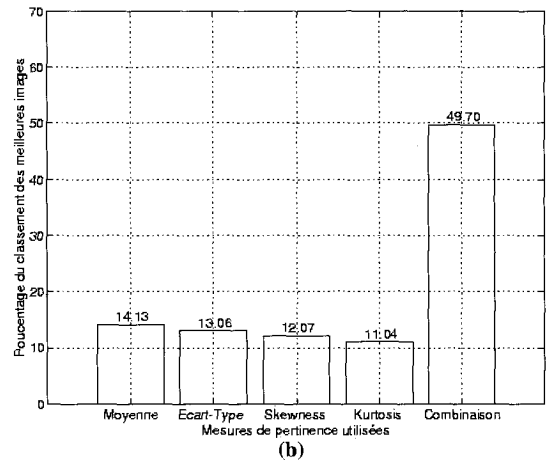
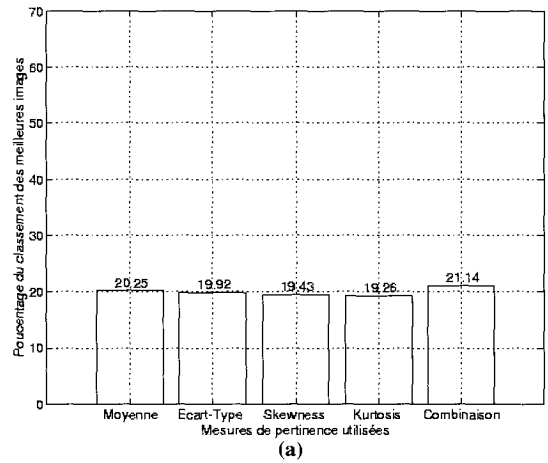


Figure 8. – Résultats des tests subjectifs avec $\beta_n\sqrt{n} = 1.22$, pour les trois taux de compression testés. a) Taux de compression égal à 1 :10. b) Taux de compression égal à 1 :20. Taux de compression égal à 1 :30.

l'hypothèse de similarité des ensembles sera rejetée. Pour un taux d'erreur de 10%, $\beta_n\sqrt{n} = 1.22$ et pour 20% $\beta_n\sqrt{n} = 1.07$. Ces valeurs ne sont pas optimisées d'un point de vue perceptuel.

Afin d'ajuster perceptuellement la valeur du seuil $\beta_n\sqrt{n}$ du test de Kolmogorov-Smirnov, nous avons généré cinq images, en

utilisant l'algorithme LBG-SPE, correspondant à des valeurs de $\beta_n\sqrt{n}$ comprises entre 1.07 et 1.22. Nous avons mené des tests psychophysiques, durant lesquels le sujet doit classer, toujours par ordre décroissant de qualité, les cinq images reconstruites en fonction des cinq seuils testés.

La figure 9 présente les résultats obtenus à l'issue de ces tests. Nous avons appliqué une interpolation linéaire entre les résultats obtenus en fonction des valeurs $\beta_n\sqrt{n}$ du test. Nous remarquons que, pour une valeur critique supérieure à 1.15, la qualité des différentes images reconstruites sont peu discernables pour les observateurs. Pour des valeurs inférieures à 1.15, la qualité des images se dégrade rapidement. Pour une valeur de $\beta_n\sqrt{n} = 1.11$, les images ont la meilleure qualité dans 18% des cas, tandis que pour $\beta_n\sqrt{n} = 1.07$, on obtient un pourcentage de 10%. La valeur critique pivot, en terme de qualité des images reconstruites, est alors donnée par la valeur 1.15.

En utilisant cette valeur, nous appliquons le test de Kolmogorov-Smirnov pour obtenir l'échantillon Λ_0 optimal et ainsi avoir la meilleure qualité d'image. Cette valeur représente un optimum perceptuel du seuil du test.

5.3.2. comparaison des algorithmes

Pour mener à bien cette comparaison, nous utilisons, à l'instar de la sélection de l'indice de pertinence, une mesure objective de distorsion entre l'image originale et l'image compressée. Nous utilisons également des tests psychophysiques de la qualité afin de confirmer les résultats obtenus avec les mesures objectives de qualité.

a. Mesure objective de la qualité

Nous utilisons la mesure du PQS couleur définie en section 4.1.2. La figure 9 illustre les résultats obtenus pour les images encodées par l'algorithme LBG-SPE et l'algorithme LBG standard, et ce pour les trois taux de compression testés.

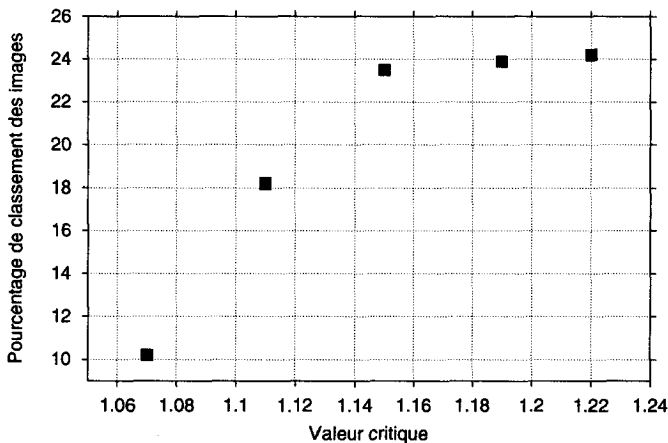


Figure 9. – Evolution de la valeur critique $\beta_n\sqrt{n}$ en fonction du pourcentage des meilleures images obtenues.

Nous voyons sur les figures 10a, 10b et 10c que l'algorithme LBG-SPE est nettement supérieur en terme de qualité des images reconstruites, à fort taux de compression. En effet, on remarque

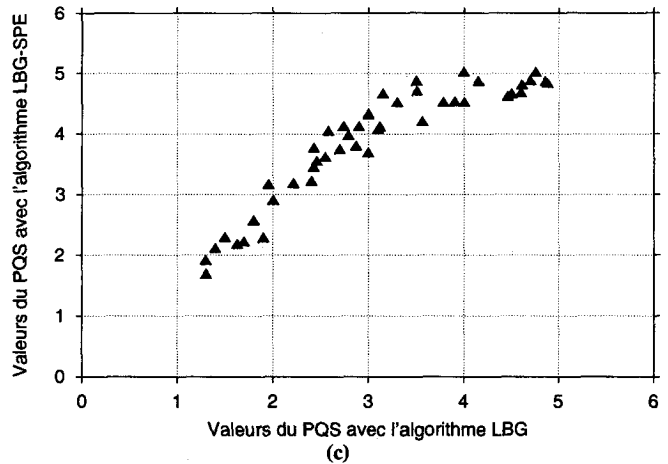
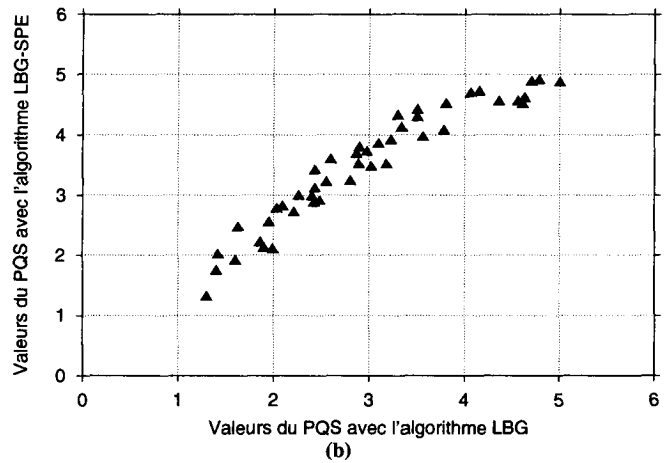
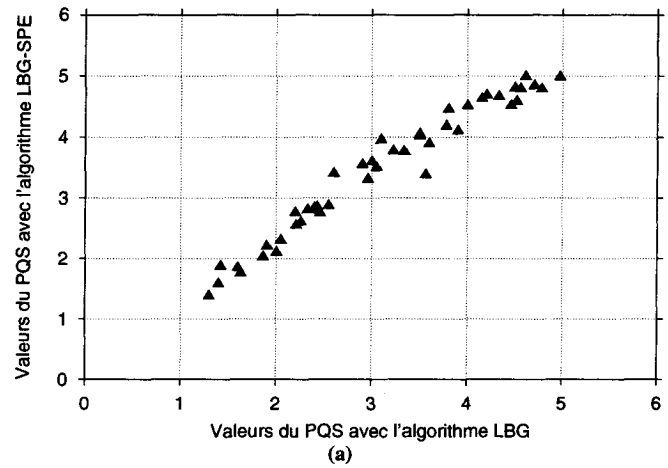


Figure 10. – Résultats de la comparaison de l'algorithme LBG standard et de l'algorithme LBG-SPE selon le critère de qualité PQS couleur.

- a) Comparaison des algorithmes pour un taux de compression de 1 :10.
- b) Comparaison des algorithmes pour un taux de compression de 1 :20.
- c) Comparaison des algorithmes pour un taux de compression de 1 :30.

que la courbure de l'axe principal du nuage des réponses devient de plus en plus prononcée au fur-et-à-mesure que le taux de compression augmente. Pour un taux de compression de 1 :10, lorsque l'on a une valeur du PQS égale en moyenne à 2 pour l'algorithme LBG, on obtient une valeur du PQS égale en moyenne à 2.2 pour l'algorithme LBG-SPE. De même, pour un taux de compression égal à 1 :20, une valeur du PQS égale en moyenne à 2 pour l'algorithme LBG correspond à une valeur du PQS égale en moyenne à 2.5 pour l'algorithme LBG-SPE. Pour un taux de compression de 1 :30, une valeur du PQS égale en moyenne à 2 (algorithme LBG) correspond à une valeur du PQS égale en moyenne à 3 (algorithme LBG-SPE).

Lorsque les valeurs du PQS, pour les deux méthodes et pour tous les taux de compression testés, sont proches de 5, nous avons sensiblement la même qualité, avec tout de même une légère préférence pour l'algorithme LBG-SPE (cette constatation est identique pour une valeur du PQS proche de 1).

Nous pouvons noter que plus le taux de compression est élevé et plus la qualité des images décodées obtenue par l'algorithme LBG-SPE est supérieure à celle obtenue par l'algorithme LBG.

De façon à valider cette hypothèse, nous utilisons des tests subjectifs de qualité.

b. Mesure subjective de la qualité

Les tests psychophysiques ont été utilisés de façon à évaluer les différences de qualité (si elles existent) des images reconstruites par l'algorithme LBG-SPE et l'algorithme LBG et à les quantifier. Ces tests ont été réalisés pour chacun des taux de compression définis auparavant.

Tableau 2. – Réponses moyennes des observateurs.

Stimuli	Meilleur à gauche	Meilleur à droite
<LBG-SPE, LBG>	0,8	0,2
<LBG-LBG, SPE>	0,18	0,82

a) Réponse moyenne des observateurs pour le taux de compression égal à 1 :10.

Stimuli	Meilleur à gauche	Meilleur à droite
<LBG-SPE, LBG>	0,83	0,17
<LBG-LBG, SPE>	0,08	0,92

b) Réponse moyenne des observateurs pour le taux de compression égal à 1 :20.

Stimuli	Meilleur à gauche	Meilleur à droite
<LBG-SPE, LBG>	0,88	0,12
<LBG-LBG, SPE>	0,09	0,91

c) Réponse moyenne des observateurs pour le taux de compression égal à 1 :30.

Les observateurs doivent évaluer une paire d'images composée de l'image codée par l'algorithme LBG-SPE et de l'image compressée via la technique du LBG, en fonction de leur qualité. La question posée aux sujets est : « Quelle est l'image de meilleure qualité? ». En utilisant l'équation 16 et à partir des données du tableau 2, nous avons mesuré la sensibilité de l'observateur pour les trois taux de compression. Le tableau 3 récapitule les valeurs des sensibilités calculées en fonction des différents niveaux de compression.

Tableau 3. – Résultats du calcul de la valeur de sensibilité de l'observateur.

Taux de compression	Sensibilité d'
1 :10	1.24
1 :20	1.67
1 :30	1.78

Nous notons que ces valeurs d' sont supérieures à 1.0, ce qui signifie que dans la majorité des cas, l'observateur a désigné comme méthode restituant les images avec la meilleure qualité la méthode LBG-SPE.

Ces résultats nous permettent de conclure que l'algorithme LBG-SPE génère un dictionnaire de qualité supérieure à celui obtenu à l'aide de l'algorithme LBG standard. Plus le taux de compression est élevé, plus la différence de qualité est visible.

c. Temps de calcul

Les résultats des expérimentations précédentes démontrent que la qualité d'une image codée par l'algorithme LBG-SPE est supérieure à celle obtenue à partir de l'algorithme LBG standard.

Puisque la construction du dictionnaire s'effectue sur un sous-ensemble de la base d'apprentissage, l'algorithme proposé doit être plus rapide que l'algorithme LBG classique (et ce, même si nous utilisons une étape de préclassification dans l'algorithme LBG-SPE). En effet, le gain obtenu en complexité par l'élimination de la redondance statistique de la base, permet d'accroître la complexité dans le choix des éléments du dictionnaire.

Nous comparons également le temps de construction du dictionnaire avec l'algorithme LBG-SPE privé de cette étape, avec le temps de construction en utilisant l'algorithme LBG standard et l'algorithme LBG-SPE. Cela nous permet alors de mesurer le gain de temps obtenu.

Nous voyons l'évolution du temps de calcul pour les trois algorithmes (figure 11). Nous remarquons nettement que l'étape de construction du dictionnaire est plus rapide avec l'approche du LBG-SPE. Pour générer un dictionnaire de taille 256, le temps moyen nécessaire à l'algorithme LBG-SPE est de 4 s alors que le temps moyen de calcul de l'algorithme LBG classique est de 7.5 s, et de 9.4 s pour l'algorithme LBG-SPE privé de l'étape de sous-échantillonnage. Ce dernier est, bien entendu, plus lent que

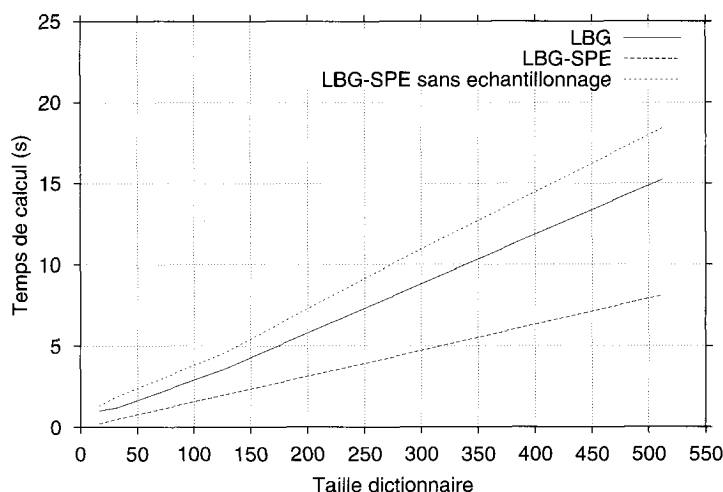


Figure 11. —Evolution du temps d'exécution de l'étape de construction du dictionnaire par l'algorithme LBG-SPE, l'algorithme LBG-SPE privé de l'étape d'échantillonnage et l'algorithme LBG classique (tous les calculs ont été effectués sur une station SUN SPARC 20).

l'algorithme LBG classique. En effet, la différence réside principalement dans la plus grande complexité du choix des vecteurs code.

Plus la taille du dictionnaire à construire est grande et plus la différence de temps moyen de calcul augmente : pour un dictionnaire de 300 vecteurs code, la différence de temps moyen est de 4.10 s pour l'algorithme LBG, et de 6.04 s pour l'algorithme LBG-SPE privé de l'étape de sous-échantillonnage.

L'algorithme LBG-SPE de construction du dictionnaire proposé dans cet article permet non seulement d'améliorer la qualité des images reconstruites mais aussi d'accélérer le processus de construction.

6. conclusion

Dans cet article nous avons proposé un algorithme de construction du dictionnaire pour la quantification d'images couleur. Cet algorithme procède comme suit. Chaque élément de la base d'apprentissage est associé à un indice de pertinence scalaire. A partir de cette nouvelle base d'apprentissage, nous sélectionnons un sous-ensemble statistiquement et perceptuellement représentatif. Celui-ci est alors classifié à l'aide d'un algorithme stochastique d'estimation des paramètres d'une loi mélange de gaussiennes. Le résultat de cette classification est alors projeté dans la base d'apprentissage originelle. L'algorithme LBG standard est ensuite appliqué à chacun des sous-ensembles ainsi constitués. La taille de chacun des dictionnaires associés est proportionnelle à la probabilité estimée de chacune des classes. Le dictionnaire final est constitué de l'union des dictionnaires.

En comparant cette technique basée sur un « sous-échantillonnage » de la base et une préclassification associée avec l'algorithme LBG classique, nous avons montré que la qualité des images reconstruites est supérieure. Ces résultats ont été obtenus en utilisant des mesures objectives de la qualité ainsi que des tests psychophysiques. De plus, nous avons une économie de temps de calcul avec l'algorithme proposé lors de la construction du dictionnaire représentatif de l'image.

La méthode qui a été développée dans cet article est basée sur un indice de pertinence associé à chacun des vecteurs de la base d'apprentissage issue de l'image à compresser. Cet indice a été validé par des tests psychophysiques. Il s'agit d'une combinaison du vecteur couleur moyenne \hat{m} et de l'écart-type $\hat{\sigma}$ du vecteur définie par $\hat{\eta} = 0.6\hat{m} + 0.4\hat{\sigma}$.

Nous avons donc introduit une nouvelle approche pour la construction du dictionnaire, se traduisant par un gain de temps dans sa construction et par une amélioration de la qualité des images décodées.

BIBLIOGRAPHIE

- [1] A. Das and A. Gersho, « Variable dimension spectral coding of speech at 2400 bps and below with phonetic classification », in *ICASSP*, vol. 1, (Detroit, Michigan), p. 492–495, may 1995.
- [2] W. Jiang, X. Wu and W.-Y. Ng, « Vq index coding for high-fidelity medical image compression », in *ICIP*, (Santa-Barbara, California), p. 678–681, oct. 1997.
- [3] M. T. Orchard, « A fast nearest-neighbor search algorithm », in *ICASSP'91*, vol. 4, (Toronto, Canada), p. 2297–3000, apr. 1991.
- [4] C. M. Huang, Q. Bi, G. S. Stiles and R. W. Harnix, « Fast full search equivalent encoding algorithms for image compression using vector quantization », *IEEE Trans.*, vol. 1, p. 413–416, July 1992.
- [5] H. Monawer, « Image vector quantisation using a modified LBG algorithm with approximated centroids », *Electronics Letters*, vol. 31, p. 174–175, feb. 1995.
- [6] V. R. Algazi, G. E. Ford, R. R. Estes and A. El-Fallah, « A progressive analysis based perceptually transparent coder for still images », in *SPIE, Very High Resolution and Quality Imaging*, vol. 2663, (San José, CA), p. 155–166, 1996.
- [7] P. C. Cosman, R. M. Gray and M. Vetterli, « Vector quantization of image subbands : A survey », *IEEE Transactions on Image Processing*, vol. 5, p. 202–225, feb. 1996.
- [8] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1991.
- [9] A. Trémeau, C. Charrier and H. Cherifi, « A vector quantization algorithm based on the nearest neighbor of the furthest color », in *International Conference on Image Processing*, vol. 3, (Santa Barbara), p. 682–685, IEEE, oct. 1997, ICIP'97.
- [10] B. Ramamurthi and A. Gersho, « Classified vector quantization of images », *IEEE Transactions on Communications*, vol. COM-34(11), p. 1105–1115, 1986.
- [11] K. L. Oehler and R. M. Gray, « Combining image compression and classification using vector quantization », *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, p. 461–473, may 1995.
- [12] P. C. Cosman, K. L. Oehler, E. A. Riskin and R. M. Gray, « Using vector quantization for image processing », in *IEEE Proceedings*, vol. 81, p. 1326–1341, sept. 1993.
- [13] B. S. Everitt and D. J. Hand, *Finite Mixture Distributions*, Chapman and Hall, 1981.

- [14] H. Bouyanzer, R. Grisel, H. Cherifi and R. Debrie, «Segmentation d'images couleur, approche scalaire et vectorielle», in *Actes du 13ème colloque GRETSI*, (Antibes), sept. 1991.
- [15] S. P. Lloyd, «Least squares quantization in PCM», *IEEE Transactions on Information Theory*, vol. IT-28, p. 129–137, mar. 1982.
- [16] H. Cherifi, «Evaluation de tests d'hypothèse», in *Onzième Colloque GRETSI*, vol. 1, (Nice, France), p. 81–84, June 1987.
- [17] D. Freedman, R. Pisani and R. Purves, *Statistics*, W. W. Norton & Company, 1978.
- [18] A. P. Dempster, N. M. Laird and D. B. Rubin, «Maximum likelihood from incomplete data via the EM algorithm», *Royal of Statistical Society*, vol. 39, p. 1–38, 1977.
- [19] G. Celeux and J. Diebolt, «the SEM algorithm : A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem», *Computational Statistics Quarterly*, no. 2, p. 73–82, 1985.
- [20] W. Qian and D. M. Titterton, «Estimation of parameters in hidden markov models», *Trans. Roy. Soc.*, p. 407–428, 1991.
- [21] H. Cherifi and R. Grisel, «Filter estimation maximization algorithm for image segmentation», in *International Conference on Acoustics, Speech and Signal Processing*, vol. 5, (Adelaide, Australia), p. 133–136, 1994.
- [22] B. Benmiloud and W. Pieczynski, «Estimation des paramètres dans les chaînes de markov cachées et segmentation d'images», *Traitement du Signal*, vol. 12, p. 433–454, 1995.
- [23] L. Bédard, A. Saadane and D. Barba, «Représentation et quantification psychovisuelles d'images couleur», in *GRETSI*, (Grenoble), p. 965–968, sept. 1997.
- [24] P. C. Cosman, R. M. Gray and R. A. Olshen, «Evaluating quality of compressed medical images : SNR, subjective rating and diagnostic accuracy», *IEEE Proceedings*, vol. 82, p. 919–932, June 1994.
- [25] C. Charrier and H. Cherifi, «Evaluation des espaces couleur pour la compression par quantification vectorielle», in *CORESA'97*, p. 48–55, 26–27 mars 1997.
- [26] K. Kotani, Q. Gan, M. Miyahara and V. R. Algazi, «Objective picture quality scale for color image processing», in *IEEE International Conference on Image Processing*, (Washington DC), p. 133–136, 1995.
- [27] P. Lennie and M. D'Zmura, «Mechanisms of color vision», *CRC Critical Reviews in Neurobiology*, vol. 3, no. 4, p. 333–400, 1988.
- [28] R. W. G. Hunt, «Revised colour-appearance model for related and unrelated colours», *Color Research and Application*, vol. 16, p. 146–165, June 1991.
- [29] CIE, «Industrial colour-difference evaluation», Tech. Rep. 116, Bureau Central de la CIE, 1995.
- [30] N. Avdhanam and V. R. Algazi, «Prediction and measurement of high quality in still image coding», in *SPIE, Very High Resolution and Quality Imaging*, vol. 2663, p. 100–109, feb. 1996.
- [31] C. Charrier, K. Knoblauch and H. Cherifi, «VQ-coded image quality optimized by color space selection», in *Picture Coding Symposium*, vol. 143, (Berlin, Germany), p. 195–200, sept. 1997.
- [32] N. A. Macmillan and C. D. Creelman, *Detection theory : a user's guide*, Cambridge University Press, 1990.
- [33] C. Charrier, K. Knoblauch and H. Cherifi, «Perceptual distortion analysis of color image VQ-based coding», in *SPIE, Very High Resolution and Image Quality II*, vol. 3025, (San-Jose, California), p. 134–143, feb. 1997.
- [34] Y. S. Feng and N. M. Nasrabadi, «Dynamic address-vector quantization of RGB colour images», *IEEE Proceedings*, vol. 138, p. 225–231, aug. 1991.
- [35] R. S. Berns, R. J. Motta and M. E. Gorzynski, «CRT colorimetry, part I : Theory and practice», *Color Research and Application*, vol. 18, p. 299–314, oct. 1993.

Manuscrit reçu le 16 février 1998.

LES AUTEURS

Christophe CHARRIER



Docteur de l'Université Jean Monnet de Saint-Etienne. Il est actuellement en stage postdoctoral au Laboratoire de Radiocommunication et de Traitement du Signal du département Génie Electrique et Génie Informatique de l'Université Laval de Québec. Ses activités de recherche portent sur la compression des images couleur, sur la mesure de qualité, ainsi que sur la détection et la reconnaissance de cible par radar.

Hocine CHERIFI



Docteur de l'Institut National Polytechnique de Grenoble. Maître de conférence habilité, il dirige l'opération Compression et Diagnostic au sein de l'Equipe Ingénierie de la Vision du Laboratoire Traitement du Signal et Instrumentation de l'Université Jean Monnet. Ces travaux de recherche concernent la modélisation et la compression d'image.