



Notions sur la compression des données dans les applications multimédias

Gérard PUIMATTO

Service de l'informatique éducative

L'exploitation de technologies multimédias pour une diffusion auprès du grand public, et notamment dans le secteur éducatif, suppose de disposer d'équipements d'utilisation simple et d'un coût raisonnable.

Quels que soient les objets envisagés – sons, images naturelles fixes ou animées, réalisations infographiques –, les volumes de données générés sont considérables ; le stockage, notamment sur un support d'édition, le transport sur les réseaux publics, mêmes dits “ à hauts débits ”, voire la simple exploitation à partir d'un disque dur, supposent, dès lors qu'il s'agit par exemple de vidéo, la mise en œuvre de débits considérablement plus importants que ceux qui sont actuellement disponibles.

Une image au standard VGA + (640 x 480, 256 couleurs) occupe un octet par pixel soit 300 ko. Dans cette définition, qui reste limitée par rapport aux standards audiovisuels, 25 images (soit une seconde de vidéo) occuperaient plus de 7 Mo, sans parler du son ! Il est aisé de comprendre que ces valeurs sont incompatibles avec la plupart des supports informatiques actuels, et même avec les contraintes des réseaux publics ou privés envisageables à moyen terme.

La conclusion s'impose d'elle même : en l'état, une telle technique de codage est inutilisable ; il est impératif de faire tenir ces informations dans un volume largement moindre, en un mot de les compresser.

Le propos n'est pas ici de proposer une présentation complète des techniques de compression, mais simplement de préciser quelques notions de base sur ce sujet. Les personnes qui doivent effectuer des choix technologiques en ce domaine, qu'il s'agisse de préoccupations éditoriales ou plus simplement de décisions d'équipements, trouveront sans doute dans ce document quelques éléments qui pourront les aider à décoder les discours commerciaux.

Sommaire

Introduction	3
Approche historique.....	4
La théorie de l'information.....	4
Évaluation de la compression	5
La compression des fichiers binaires.....	6
Les codages statistiques à longueur variable	6
Les codages par dictionnaires.....	8
Compression des images et des sons : des données fortement corrélées.....	10
Codage par répétition : le “ <i>Run Length Encoding</i> ”.....	12
Les compressions dégradantes.....	13
Les méthodes prédictives : compression différentielle et différentielle adaptative	15
Au secours : les maths reviennent !	17
Un problème spécifique : la vidéo numérique.....	26
Les images animées MPEG	27
Autres techniques utilisées pour les images animées.....	30
En guise de conclusion – provisoire – à propos des images animées.....	32
Annexes	37
Méthode de Huffman : exemple	37
Méthodes Lempel Ziv : algorithmes et exemple.....	39
Compression <i>Run Length</i> : exemple	43
Bibliographie	44

Cette Note technique est issue d'une étude réalisée pour le stage de formation des personnels du réseau CNDP “ Images numérisées ” (Caen, octobre 1993). Elle a pour objectif d'éclaircir quelque peu un domaine complexe, et d'aider les consultants en ingénierie du réseau CNDP dans leurs activités d'aide et de conseil, tant en matière d'équipement des établissements scolaires que de définition des orientations des centres. Il est, bien entendu, hors de notre propos de présenter de manière rigoureuse et exhaustive les principes et techniques de compression ; cette note ne vise qu'à donner une information de base, nécessaire pour mieux comprendre les nombreux articles et documentations y faisant référence. Une bibliographie est proposée pour ceux qui désirent aller plus loin.

Dans son fond et dans sa forme, cette note doit beaucoup aux relectures et aux points de vue de Michel Banckaert, Dominique Cavet, Michel Lavacry, Alain Maillet, Nicole Rodriguez, Pierre Antoine Taufour et Serge Torres.

La diffusion et la duplication de cette note est autorisée, et même encouragée, à la seule condition de mentionner son origine en cas d'utilisation totale ou partielle.

CNDP – Direction de l'ingénierie éducative

Contact : Gérard Puimatto, Service de l'informatique éducative

Téléphone : (1) 46 12 84 78 – (1) 46 12 84 96 (SIE)

Introduction

Les développements technologiques et les exigences des utilisateurs conduisent à des quantités de données toujours plus importantes ; ainsi, depuis les débuts des technologies de l'information, le problème de l'exploitation optimale des voies de communication et des capacités de stockage est toujours resté un sujet d'actualité.

De multiples études ont été menées sur les algorithmes de compression, dans le but de mettre les données sous un format tel qu'elles occupent moins de volume. Une fois compressées, les données ne sont plus directement accessibles, et il est nécessaire de les décompresser pour qu'elles redeviennent intelligibles. La grande variété des domaines d'exploitation, chacun ayant ses contraintes spécifiques (nature des données, capacités de traitement), conduit aujourd'hui à un très grand nombre de procédés de compression.

Dans le domaine naissant du multimédia, la plupart des données traitées sont volumineuses : une seconde de son numérisée selon le standard CD-Audio occupe environ 1,5 Mbits, et une image 640 x 480 en 16 millions de couleurs approche le megaoctet. Pourtant, c'est dans le domaine de l'image animée, et en particulier de la vidéo, que le problème est particulièrement crucial : le débit nécessaire pour une séquence composée d'images VGA 256 couleurs serait de 56 Mbits/s, et les besoins d'une transmission selon la norme de vidéo numérique CCIR 601 s'élèveraient, quant à eux, à... 166 Mbits/s !

Face aux données à transmettre, les débits couramment disponibles constituent la contrainte principale à prendre en compte. C'est naturellement le CD-ROM simple vitesse, seul disponible il y a encore peu de temps, ainsi que le réseau Numéris (RNIS), qui ont servi de référence aux principaux travaux. Pour le CD-ROM, on obtient un débit d'environ 1,5 Mbits/s, soit à peu près 3 % du flux nominal qui serait nécessaire pour transmettre des images VGA animées et moins de 0,6 % si la base retenue est le format vidéo numérique normalisé CCIR 601. Pour une transmission sur le réseau Numéris, limité aujourd'hui à 64 kbits/s, ces valeurs devraient encore être divisées par près de 20 !

Les ordres de grandeur de ces quelques valeurs montrent bien l'ampleur du problème à résoudre. Les techniques de compression, aussi élaborées soient-elles, ne peuvent à elles seules apporter une réponse à tous les besoins, et il sera nécessaire dans bien des cas de faire des sacrifices sur la définition initiale. C'est ainsi que, dans le domaine de la vidéo, on a défini le format SIF (*Source Intermediar Format*) : retenant une définition inférieure, il exploite des flux – et une qualité – sensiblement plus réduits que le CCIR 601.

De plus, une exploitation facile sur des matériels de grande diffusion suppose la disponibilité de composants électroniques peu onéreux, spécialisés dans la compression/décompression. Les investissements nécessaires à leur fabrication en grande série ne peuvent être engagés en l'absence d'un standard reconnu, qu'il soit normalisé ou non. Pour l'heure, même si quelques solutions ont fait l'objet de normalisations, et même de mise en production industrielle, aucun standard ne s'est encore véritablement imposé, en particulier pour la vidéo.

Devant cette situation encore confuse, il convient sans doute de comprendre – un peu – les principes de fonctionnement des diverses techniques mises en œuvre. La multiplicité et la complexité des solutions ne permettent pas de dire celle qui sera la bonne “ à tous coups ”. À l'heure des choix, tant pour un équipement matériel ou logiciel que pour un projet éditorial, il convient surtout de se méfier des promesses commerciales, et de privilégier, lorsque c'est possible, les options qui préserveront au mieux l'avenir.

Approche historique

La théorie de l'information

L'étude scientifique de l'information a débuté en 1924 sous l'égide de mathématiciens et physiciens (Gabor, Hartley, Nyquist, Wiener), mais elle n'a pris toute sa dimension qu'avec l'élaboration de la théorie mathématique de l'information, publiée en 1949 par Shannon et Weaver. Le développement des transmissions télégraphiques imposait, déjà, d'optimiser l'utilisation des canaux de transmission offerts, et donc d'éliminer des données à transmettre tout ce qui n'était pas indispensable à la compréhension.

Ainsi, la théorie de l'information attribue à chaque signifiant élémentaire d'un message une mesure de sa *quantité d'information* théorique, d'autant plus élevée que ce signifiant est " nouveau ", c'est-à-dire non déductible des éléments précédents. Inversement, on attribue également une *mesure de redondance* à chaque élément de message, d'autant plus faible que son poids d'information est fort.

Pour faire comprendre le principe de sa théorie, on rapporte que Shannon se livrait à un petit jeu lors des soirées mondaines : un convive ayant choisi une phrase, les autres doivent, en connaissant uniquement le début, deviner le mot suivant. Le taux de réussite de l'assistance donne une idée de ce qu'est le " poids de redondance ", alors que son complément à 1 (100 % - *taux de réussite*) illustre la notion de " poids d'information ". Bien que la " méthode " soit fort grossière, elle permet de mieux approcher des concepts, par ailleurs fort compliqués...

Les systèmes organisés comme les langues naturelles offrent des caractéristiques de redondance bien précises, qui peuvent être exploitées par les systèmes de transport de l'information pour optimiser l'utilisation des canaux disponibles. La fréquence d'utilisation de chaque lettre de l'alphabet, par exemple, constitue une première évaluation de sa probabilité d'apparition ; les caractères qui précèdent permettent également d'améliorer la " prédiction " (une voyelle, par exemple, est généralement suivie d'une consonne, et ce dans une proportion parfaitement mesurable). En faisant du caractère l'unité minimale d'information, le message étant composé dans une langue donnée, on peut donc parfaitement évaluer ses caractéristiques de redondance. De telles évaluations peuvent également être conduites en utilisant d'autres signifiants élémentaires, par exemple les mots dans le cadre de systèmes de communications utilisant un vocabulaire réduit.

Cependant, il ne faudrait pas déduire de ce qui précède que ce qui est redondant n'est qu'un " bruit " par rapport au " signal " principal, et peut donc être supprimé sans altérer le message. Seul un signifiant " totalement redondant " (situation impossible), qui serait l'illustration d'un événement certain (probabilité égale à 1), pourrait être considéré comme inutile. En outre, la redondance des langages naturels est souvent utile pour faciliter la compréhension d'un message, notamment en cas de dégradation partielle (cf. communications téléphoniques dégradées). La mesure mathématique de la quantité d'information, on l'a compris, n'a qu'un très lointain rapport avec le " poids sémantique " du message dans son système d'origine.

Shannon définit également *l'entropie* d'un message, d'une manière comparable à la notion utilisée en thermodynamique. Dans cette discipline, il s'agit d'une grandeur physique qui permet d'évaluer la dégradation de l'énergie d'un système ; on dit souvent que l'entropie mesure le degré de désordre d'un système. Dans la théorie de l'information, l'entropie d'un



message indique le poids d'information mathématique qu'il porte, et donc sa "compressibilité" théorique.

D'un strict point de vue quantitatif, on ne peut que constater que la plupart des langues véhiculaires comprennent beaucoup moins de mots que les possibilités offertes par l'alphabet, et il en est de même de données organisées devant traduire des images et des sons.

" [...] le nombre des images informatiquement possibles en 256 couleurs, au format 512 x 512 pixels, est gigantesque : $2^8 \times 512 \times 512 = 10^{704439} \times 10^{73}$. Or, 10^{73} représente le nombre de molécules dans l'univers, ce qui "contraint" la nature à ne proposer qu'un nombre infiniment plus faible [...] "[Marseau, 1992, p. 15]

Ainsi, les codages traditionnels, qui représentent ou cherchent à représenter toutes les combinaisons théoriques, se révèlent extrêmement simplistes et peu performants. Le codage ASCII sur 7 bits, par exemple, fournit plus de 4 000 milliards de possibilités pour coder les mots de 8 lettres, alors que les dictionnaires de la langue française en comptent moins de 3 000...

En tenant compte des caractéristiques d'entropie des données à traiter, il est donc possible de définir des codages plus performants. Les travaux de Shannon et de ses collaborateurs ont conduit à développer des codages statistiques fondés sur la fréquence d'apparition d'une information.

La théorie de l'information s'est attachée, dès les années vingt, à quantifier les flux nécessaires aux transferts d'information, ainsi qu'à optimiser l'utilisation des voies de communication en recherchant les codages les plus adaptés.

Shannon et Weaver ont développé une théorie mathématique qui attribue à chaque signifiant élémentaire un poids d'information d'autant plus élevé qu'il est non déductible du contexte. Le degré de complexité, ou, en référence à la thermodynamique, de désordre du message considéré est quantifié par son entropie.

Que chacun se rassure cependant, il n'est pas nécessaire d'avoir lu – et compris – les œuvres complètes de Shannon et Nyquist pour comprendre les principes fondamentaux des techniques de compression, pas plus d'ailleurs que pour utiliser un micro-ordinateur !

Évaluation de la compression

On retrouve dans les divers articles concernant la compression des évaluations de performances difficilement comparables, dans la mesure où elles ne sont pas fondées sur les mêmes principes. Il semble donc utile de préciser les moyens de mesurer la compression qui sont retenus par les spécialistes [Plume, 1993].

Le degré de réduction des données obtenu par une méthode de compression peut être évalué au moyen du *quotient de compression* défini par la formule :

$$Q_{comp} = \frac{\text{Taille initiale}}{\text{Taille après compression}}$$



Le *taux de compression*, généralement exprimé en pourcentage, est l'inverse du quotient de compression.

$$T_{comp} = \frac{1}{Q_{comp}}$$

Le *gain de compression* est également exprimé en pourcentage ; c'est le complément à 1 du taux de compression.

$$G_{comp} = 1 - T_{comp}$$

Un fichier original de 2 000 signes compressé en 800 signes présente un quotient de compression de 2,5, un taux de compression de 40 %, et un gain de compression de 60 %.

La compression des fichiers binaires

Une première possibilité de compactage est de limiter la redondance des informations, sans se préoccuper de leur origine, c'est-à-dire en considérant tous les signifiants élémentaires (bits, octets, mots...) comme *a priori* statistiquement indépendants.

Par nature indépendante des données à traiter (images, sons, textes, fichiers exécutables), la compression des fichiers binaires est impérativement une transformation entièrement réversible. Les algorithmes de compression/décompression utilisés se doivent d'être non dégradants pour garantir l'intégrité des informations traitées. À ce titre, on peut considérer qu'ils relèvent simplement d'un codage particulier d'un fichier.

Ces codages optimisés sont d'ailleurs de plus en plus souvent appelés à être utilisés directement par les systèmes d'exploitation, au niveau le plus bas. Dans certains cas, leur mise en œuvre peut même amener une amélioration des performances : la durée de la lecture sur disque varie proportionnellement avec le taux de compression, et les traitements de décompactage peuvent se révéler moins longs que le gain obtenu. Leur intervention est alors totalement transparente pour les utilisateurs et l'ensemble des applications.

La compression des fichiers binaires ne tient pas compte de la nature des données.

C'est une opération qui doit être entièrement réversible, et on ne dispose d'aucune information préalable de redondance.

Les codages statistiques à longueur variable

L'origine des techniques de codage compressif est étroitement liée à l'optimisation des voies de communications télégraphiques et téléphoniques, notamment dans le cadre d'exploitations militaires. Les études les plus importantes ont été développées durant la deuxième guerre mondiale et les années suivantes, en particulier par Shannon. Le contexte particulier explique sans doute pourquoi les problèmes de compression et de cryptage sont étroitement liés dans la littérature...

Les codages dits “ à longueur variable ” (VLC *variable length coding*) utilisent les fréquences d'apparition des éléments du message pour attribuer aux plus fréquents des codes courts, et aux plus rares des codes longs. Les fréquences d'apparition peuvent, dans certains cas, être connues à l'avance (communications structurées à vocabulaire limité), mais elles doivent généralement faire l'objet d'une analyse au cours du traitement. Dans de nombreux cas, il est nécessaire d'établir une table correspondant au message à compresser, et donc d'effectuer un traitement en deux passes (analyse, puis compression proprement dite). Outre la nécessité de relire deux fois les données, cette méthode ne permet pas la compression en temps réel. Pour éviter cette difficulté, on évalue quelquefois les redondances sur une partie du fichier, qui joue alors le rôle d'échantillon représentatif. Cependant, cette méthode est souvent peu précise, la représentativité de l'échantillon n'étant généralement pas mesurée. Les équipements modernes de communication (télécopieurs, modems) utilisent plutôt une analyse en temps réel sur des paquets de données de taille réduite.

Le code Shannon-Fano a été le premier algorithme statistique à connaître un succès important. Il a cependant rapidement été concurrencé par la méthode de Huffman, plus adaptée aux modes de traitement informatiques [Huffman, 1952].

La méthode de Huffman

Procédé

À partir d'une étude statistique portant sur la fréquence des éléments signifiants présents dans les données originales, l'algorithme de Huffman attribue à chacun de ces éléments un code binaire dont la taille est d'autant plus réduite que le code est fréquent. Sauf dans le cas de textes aux caractéristiques rédactionnelles précises, dans une langue donnée (la fréquence des différentes lettres peut alors être estimée *a priori*), l'application de cette méthode suppose une première lecture des données, préalable au codage proprement dit, ayant pour buts une analyse statistique et la constitution d'une table de codage. La table de correspondance du codage ainsi établie devra être transmise pour le décodage (un exemple de traitement Huffman est proposé en annexe).

Performances

Les performances de cet algorithme dépendent des caractéristiques des données à compacter, mais il est possible de connaître le taux de compressibilité sans avoir à effectuer le compactage, uniquement à partir de l'analyse statistique. Les performances peuvent aussi être améliorées en changeant la taille des motifs dont on limite la redondance, et éventuellement en scindant les fichiers à coder. Le gain de compression moyen cité par Mark Nelson [Nelson, 1993, p. 389 et suivantes] est de 31 %, et les opérations de compactage et décompactage se révèlent très lentes.

Implémentations

Le codage de Huffman est utilisé, généralement associé à un ou plusieurs autres procédés, dans diverses situations de compression. On peut citer notamment les divers formats utilisés en télécopie et en transmission de données sur ligne téléphonique (protocoles MNP), ainsi que le Photo CD de Kodak pour les hautes définitions, ou encore les procédés MPEG.

Le codage de Huffman est fondé sur une analyse des fréquences des signes dans les données à traiter. Actuellement, il est rarement utilisé seul, mais en revanche il est souvent associé à d'autres processus pour optimiser le codage final (télécopie, Photo CD, etc.).

Les codages par dictionnaires

Les codages par dictionnaires sont également fondés sur l'analyse des répétitions dans les données à traiter. Cependant, il ne s'agit plus ici de rechercher des occurrences de signifiants considérés comme élémentaires (généralement des octets), mais de “ mots ” (groupes d'octets) de longueur variable. Les mots répétés prennent place dans un dictionnaire, et chacun d'eux est remplacé, dans les données compressées, par sa seule adresse dans le dictionnaire.

Si on considère que le dictionnaire français comporte environ 2 000 mots de 6 lettres, il est possible de les coder sur 11 bits, au lieu des 30 nécessaires au minimum selon un codage alphanumérique théorique (5 bits pour un alphabet de moins de 32 caractères), et des 48 bits nécessaires pour un mode ASCII. En renouvelant ce procédé sur les mots de différentes longueurs, on peut coder les textes en émettant pour chaque mot sa longueur suivie de son adresse dans le dictionnaire. Cet exemple montre bien qu'il est largement possible d'optimiser les codages classiques. Le procédé exposé est cependant trop simpliste pour satisfaire une gamme étendue de besoins (nécessité de s'accorder sur un dictionnaire, absence de traitement des termes non prévus, etc.) ; il est cependant quelquefois utilisé dans le cas d'univers de communications limités à un vocabulaire réduit et dotés d'un dictionnaire prédéfini connu du destinataire, et ne devant donc pas être transmis.

Pour pallier ces difficultés, et produire un procédé capable de s'adapter à une grande variété de besoins, il est nécessaire de définir un dictionnaire spécifique aux données à traiter. Cette méthode a cependant pour inconvénients de nécessiter deux lectures du fichier à traiter (la première pour constituer le dictionnaire, et la deuxième pour coder effectivement les données) et d'imposer la conservation d'un dictionnaire, qui peut devenir très lourd, en tête du fichier produit.

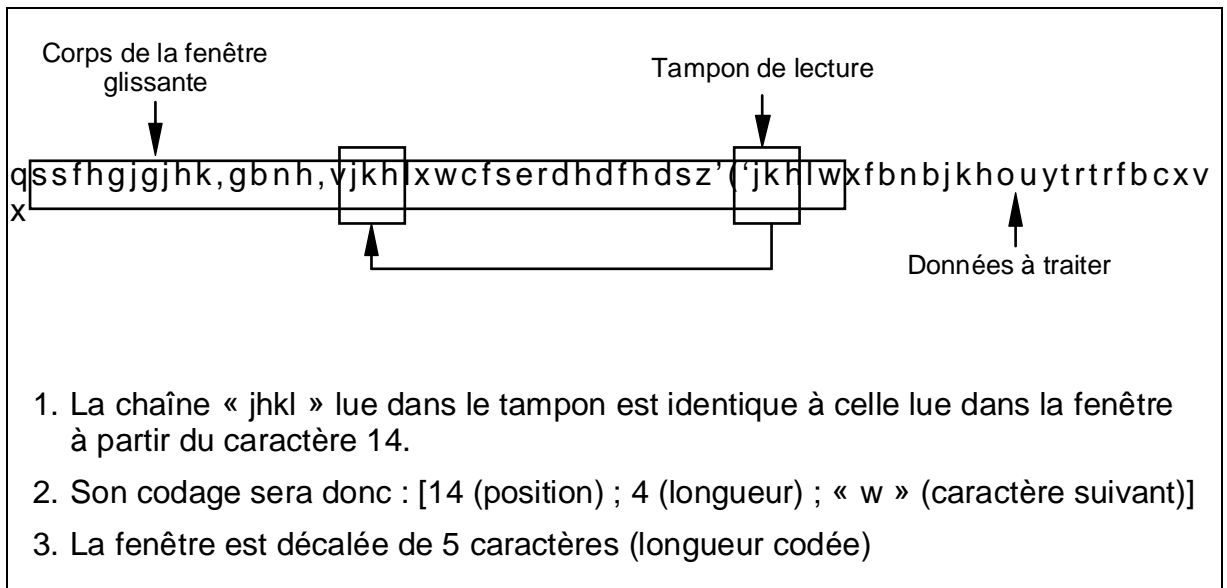
Abraham Lempel et Jacob Ziv ont publié en 1977 et 1978 des algorithmes de compression d'usage général (indépendant de la nature des données codées) fondés sur la construction d'un dictionnaire dynamique qui n'a pas à être sauvé en tant que tel, puisqu'il est reproduit automatiquement par le décompacteur en fonction des données du fichier comprimé [Ziv, 1977], [Ziv, 1978].

Les méthodes Lempel-Ziv (LZ, LZx)

Les algorithmes de compression et décompression LZ ont des fonctionnements symétriques : leur principe est fondé sur l'indexation de chaînes dans un dictionnaire qui, dans les deux cas, est construit durant le traitement.

Le dictionnaire est défini comme un tableau de chaînes de tailles variables, repérées par leur adresse ; la taille du tableau est également variable, limitée par le mode de codage des adresses.

Cette méthode, plus performante que celles utilisées auparavant, souffre cependant de quelques difficultés. Sa programmation est assez complexe (gestion de pointeurs “ glissants ” sur des fenêtres, gestion de tableaux de longueur variable composés d'objets de longueur variable), et le programme peut conduire à des délais de traitement longs, notamment en compression. Le traitement s'effectuant sur une fenêtre, c'est sa taille qui détermine les performances du dispositif : lors du codage, les chaînes du tampon de lecture sont comparées à toutes les positions dans la fenêtre, et une taille réduite conduit à ne pas prendre en compte des chaînes répétées au delà de cette distance ; en revanche, une taille plus importante impose des traitements considérables, en multipliant le nombre de comparaisons nécessaires. De plus, s'il n'y a pas de correspondance, les caractères doivent être transmis individuellement, avec une longueur de séquence à zéro, ce qui peut conduire à une augmentation de la taille des données.



Principe de fonctionnement de LZ77.

En 1978, Lempel et Ziv ont publié une nouvelle version de leur algorithme [Ziv, 1978], abandonnant le concept de fenêtre glissante. Le dictionnaire se construit dynamiquement, tout au long du traitement. Chaque signe transmis par le codeur comprend un index dans le dictionnaire, ainsi que le caractère suivant à coder. La longueur de la chaîne n'a plus à être transmise, puisqu'elle est conservée dans le dictionnaire ; la concaténation de la chaîne répétée avec le caractère suivant est également placée dans le dictionnaire, et devient ainsi disponible pour la totalité du codage ultérieur. Cette nouvelle version permet de dépasser les limites de LZ77, et améliore largement les performances, surtout sur les fichiers longs. Cependant, une nouvelle difficulté apparaît : la taille du dictionnaire est limitée par le mode de codage de ses index (souvent sur 16 bits, soit 65 536 entrées), et il est donc nécessaire de gérer l'évènement " dictionnaire plein ". Une possibilité est de laisser le dictionnaire en l'état, et de simplement l'utiliser, mais cela conduit à une détérioration des performances dans le cas de longs fichiers, qui peuvent présenter de grands changements dans la nature des données ; une autre solution est de supprimer le dictionnaire existant, et d'en reconstruire un autre, mais, dans le cas de données homogènes, cela peut conduire à une détérioration des performances. La solution généralement retenue est d'analyser le taux de compression obtenu avec le dictionnaire existant, et de le reconstruire seulement si on constate une détérioration.

Terry Welch (le " W " de LZW) a publié en 1984 de nouvelles améliorations [Welch, 1984]. Contrairement au dictionnaire de LZ78, qui ne contient au départ qu'une chaîne vide repérée par l'index " 0 ", celui de LZW comprend au départ les codes ASCII, repérés de 0 à 255 ; tous les symboles peuvent ainsi être directement codés selon le dictionnaire, permettant ainsi d'éviter la détérioration des performances pour les fichiers ne présentant que peu de répétitions. De plus, Welch a également défini les principes de communication entre codeur et décodeur : le caractère autoadaptatif du procédé est amélioré par la possibilité d'adapter dynamiquement les paramètres au cours du codage, les modifications étant communiquées au décodeur selon un codage spécifique. Il s'agit par exemple de l'augmentation de la taille des adresses, de la purge partielle ou totale du dictionnaire, de la création temporaire d'un autre dictionnaire, et plus généralement de tout changement de technique de compression en cours de traitement.

La plupart des compacteurs du marché utilisant LZ et ses variantes (Arj, Pkzip...) exploitent conjointement un algorithme statistique limité (issu de Huffman ou de Shannon-Fano), mais en évitant l'analyse préalable de l'ensemble du fichier, trop pénalisante en termes de vitesse



d'exécution. Dans la plupart des cas, l'analyse est limitée à une fenêtre, dont la taille varie en fonction des mesures de représentativité statistique des résultats obtenus. D'autres variantes prévoient d'effectuer le compactage par " paquets " de taille fixe, ce qui permet des codages statistiques faciles, tout en ménageant des possibilités d'accès direct et de traitement " au vol " (modems).

Le plus cité des algorithmes LZ, le LZW (perfectionnements divers apportés par Welch en 84) est sans doute aussi le moins utilisé, car il a fait l'objet d'un brevet déposé par Unisys pour une exploitation directe par le système Unix. Dans la plupart des cas, les algorithmes mis en œuvre par les logiciels actuels sont des évolutions de LZ77 ou LZ78, exploitant quelques uns des principes mis en place par Welch, comme le codage des caractères ASCII ou l'adaptation de taille des dictionnaires.

Performances

Les codages Lempel Ziv sont à ce jour ceux qui permettent les meilleurs taux de compactage non destructif sur la plupart des fichiers. Les possibilités de paramétrage et d'association à d'autres techniques (statistiques notamment) offrent une infinité de variations permettant d'optimiser le rapport entre taux de compression et vitesse de traitement. Les taux obtenus sur un jeu d'essai [Marseau, 1992] mettent en évidence des compressions deux fois plus efficaces que celle de Huffman. Les tests effectués par Mark Nelson [Nelson, 1993] montrent des gains de compression moyens compris entre 45 et 50 % pour les différentes variantes.

Implémentations

De très nombreux programmes de tous types utilisent les algorithmes LZ. Outre les utilitaires spécialisés (*Arc, Pkzip, Lharc, Arj*), on peut citer les logiciels de sauvegarde (*QIC-122* pour les sauvegardes à bandes, *PC Backup, Norton Backup, MS Backup, Sytos*), les protocoles de transmission haute vitesse par modem (*V42bis*), ainsi que diverses implémentations intégrées aux systèmes d'exploitation (*Compress* sous Unix, *Dblspace* de DOS 6). L'ensemble des compacteurs " temps réel " (*Dblspace, Stacker...*) utilisent ces techniques. Les variantes mises en œuvre sont souvent issues de LZ 77, qui se révèle plus performant sur les fichiers courts (le dictionnaire, et donc l'efficacité de LZ 78 et des versions postérieures croissent avec la taille du fichier) ; en outre, il a l'avantage d'être dans le domaine public.

Les algorithmes Lempel Ziv sont les processus génériques les plus efficaces actuellement. Il existe d'innombrables variantes de ces algorithmes, dont le LZW, qui est sans doute le plus connu, mais aussi le moins utilisé, car il est protégé par brevet. Les processus de compactage et de décompactage sont symétriques, et procèdent à la construction d'un dictionnaire durant le traitement du fichier.

Compression des images et des sons : des données fortement corrélées

Une image informatique peut être assimilée à un tableau de pixels, organisé en lignes et colonnes, dont chaque élément a une valeur. La définition de cette valeur est obtenue par une ou plusieurs données numériques (généralement une pour les images monochromes, trois pour les images couleurs en représentation RVB, YUV ou YCC), ou bien par une adresse dans une table de couleurs prédéfinies, la palette (*Color Look Up Table*, ou CLUT).

Cependant, on constate que la couleur d'un pixel est souvent très proche, voire identique à celle des pixels contigus. Statistiquement, il y a toujours une forte corrélation entre les valeurs des pixels d'une même zone d'image, et cette spécificité peut être mise à profit pour compacter les données. Cette caractéristique est souvent désignée par le terme de “ redondance spatiale ”.

Dans le cas de phénomènes variant en fonction du temps, comme le son ou les images animées, une autre corrélation peut être mise en évidence entre des mesures successives, et elle est d'autant plus importante que la fréquence de mesure est élevée. C'est la redondance temporelle.

La quantification de ces corrélations, tant au niveau spatial pour les images fixes que temporel pour les sons et les images animées, constitue la mesure de redondance des informations. Cette notion, directement issue des travaux de Shannon, doit être prise au sens de la théorie de l'information, et non au sens commun : chaque information élémentaire se voit attribuer une quantité d'information, d'autant plus élevée qu'elle est “ nouvelle ”, c'est-à-dire non déductible du contexte. Cette quantité, évaluée en pourcentage, permet de définir également la mesure de redondance associée, qui est le complément à 1 de la quantité d'information.

Alors que, au sens commun, la redondance est liée au caractère plus ou moins superflu des informations, elle est, pour Shannon, une simple mesure mathématique indiquant sa compressibilité théorique. Aucune donnée d'une image ou d'un son n'est totalement redondante, car cela signifierait qu'elle peut être à coup sûr déduite du contexte. La “ suppression des redondances ” comme description d'une technique de compression n'a donc aucun sens, puisque aucune des informations élémentaires ne présente de redondance totale.

En fait, l'ensemble des techniques de compression des images exploitent la corrélation entre des pixels voisins pour produire une “ valeur probable ” du prochain pixel à coder. Le produit de cette prédiction étant statistiquement proche de la valeur réelle, il est alors possible d'effectuer le codage de manière plus économique, par exemple simplement en codant uniquement la différence avec la valeur prédite. Les ordres de grandeur des valeurs à transmettre sont alors statistiquement plus faibles, ce qui permet d'utiliser un nombre plus faible de bits, soit en codage de longueur fixe, soit en longueur variable (méthode statistique).



La couleur ou la luminosité de deux pixels contigus sont statistiquement très voisines, et il en est de même de deux échantillons successifs d'un enregistrement sonore numérique. La forte corrélation entre ces valeurs donne une idée de ce qu'est la redondance des informations au sens de Shannon.

Codage par répétition : le “ *Run Length Encoding* ”

Le procédé “ *Run Length* ” ne relève pas d'une théorie mathématique très complexe. Il s'agit simplement de remplacer des éléments signifiants successifs identiques par un seul d'entre eux, suivi du nombre de répétitions (un exemple de traitement RLE est donné en annexe). Ce procédé peut paraître simpliste et peu performant si on cherche à l'appliquer, par exemple, à un texte : même dans notre belle langue française, les répétitions nombreuses de lettres n'apporteraient qu'une compression dérisoire ! En revanche, si on l'applique à une image, en particulier d'origine infographique, il est aisé de s'apercevoir que les plages de couleur homogènes sont souvent importantes, surtout si le nombre de couleurs est faible, et l'image limitée à la colorisation de quelques centaines de pixels, sur un fond uniforme...

Particulièrement simple à mettre en œuvre, c'est un procédé qui a été largement utilisé par les logiciels de dessin dans les années passées, éventuellement associé à un autre algorithmes plus complexes. Dans certains cas, le RLE est utilisé pour les images animées, sans aucune exploitation de la redondance temporelle.

Procédé

Si n octets successifs sont dans un même état, il est aisé de transmettre l'octet répété et le nombre de répétitions. On pourra ainsi, dans la plupart des cas, coder sur 3 octets les n octets composant le signal initial. Dans le cas de textes, voire de fichiers binaires, cette analyse exclusivement réalisée au niveau des octets successifs n'apporterait qu'une faible amélioration ; en revanche, dans le cas d'images bit map (codées pixel par pixel), et particulièrement pour les dessins réalisés “ à main levée ”, les plages de répétition sont considérables (zones de couleurs homogènes), et les résultats beaucoup plus probants.

S'il est relativement simple de coder l'octet à répéter, suivi du nombre de répétitions dans l'octet suivant, cette méthode peut se révéler très pénalisante pour certains fichiers : à la limite, si deux octets consécutifs sont toujours différents, le volume du fichier “ compressé ” sera le double de celui du fichier initial ! Pour éviter cet inconvénient, les versions les plus avancées du codage *Run Length* utilisent un code discriminant pour indiquer le début d'une séquence “ *octet à répéter + nombre de répétitions* ”, les octets isolés restant codés sous leur forme initiale.

Performances

On le comprendra aisément, un tel codage ne peut apporter des performances considérables, mais il présente l'avantage de la simplicité. Cette méthode convient mieux à des dessins bit map, mais peut apporter des résultats comparables pour des images naturelles en 256 couleurs. Au delà (codages sur 4 096 couleurs et plus), les résultats obtenus se révèlent plus décevants, les pixels identiques étant moins fréquents.

En pratique, on obtient des gains de compression de 30 à 50 % dans la plupart des cas pour les dessins, sensiblement moins pour les images naturelles.

Implémentations

Le format PCX de Z-soft, repris à son compte par Microsoft avec les différentes implémentations de *Paintbrush*, sans doute un des plus répandus des formats graphiques,



utilise ce mode de compression. Après un en-tête de 127 octets contenant diverses informations sur le codage (taille de l'image, nombre de plans, nombre de bits par pixel, ainsi éventuellement que des informations de palette), on rencontre les données compressées selon ce principe. Dans le cas où l'image est codée selon des plans de couleur (RVB par exemple), le codage se fait ligne par ligne, successivement dans les différents plans (par exemple, en VGA 16 couleurs, ligne 1 dans les quatre plans, puis ligne 2 dans les quatre plans, etc.).

Le codage *Run Length* est également largement utilisé dans les différents modes de télécopie développés par le CCITT (télécopie groupes 3 et 4 en particulier), généralement associé à un codage statistique de Huffman.

Les extensions multimédias de *Windows* reconnaissent également les fichiers RLE, qui sont une version compressée *Run Length* du format BMP. Ce format n'a guère connu de succès auprès des éditeurs, qui préfèrent en utiliser d'autres plus répandus (PCX ou TIFF par exemple), ou directement le format BMP non compressé.

Le CD-I de Philips, parmi d'autres formats graphiques, utilise deux formats *Run Length* codés respectivement sur deux octets avec une palette sur sept bits (CLUT 7) et sur un octet avec une palette sur trois bits (CLUT 3).

La méthode *Run Length* se révèle inutile lorsque les systèmes d'exploitation ou des utilitaires de bas niveau (*Dblspace* de DOS 6.0, *Stacker*...) mettent eux mêmes en œuvre des dispositifs plus performants (généralement LZx). L'utilisation de ces outils, qui travaillent directement au stade des écritures sur disques rend la méthode *Run Length* totalement inutile : on ne constate pratiquement aucune différence de taille après compactage entre les fichiers soumis non compressés et ceux qui ont été traités par *Run Length* au préalable.

Les méthodes Run Length consistent à coder les octets répétés sous forme d'un seul octet, suivi du nombre de répétitions. Cette méthode non dégradante est très simple, mais donne néanmoins de bons résultats pour certains types d'images (dessins au trait, images monochromes). Les codages Run Length sont surtout utilisés actuellement sur des matériels disposant de faibles capacités de traitement (télécopie, CD-I).

Les compressions dégradantes

Les procédés évoqués plus haut ne produisent pas de dégradation, et relèvent à ce titre du compactage plus que de la compression. Dès que l'on accepte de perdre des informations, fussent-elles jugées superflues, on entre dans le domaine de la compression dégradante. Se pose alors la question de la limite acceptable à la perte d'informations, de la sélection des critères opportuns et des paramètres à retenir pour estimer la qualité de restitution. Le contexte sémantique n'étant pas sans importance, l'appréciation est largement subjective. Dans le cas des images, par exemple, il est difficile d'évaluer objectivement la pertinence des informations, et l'appréciation ne peut que faire intervenir des éléments subjectifs. On comprendra aisément que les impératifs de séparation des couleurs, notamment, ne sont pas les mêmes sur un tableau de Van Gogh et sur une photo architecturale, et qu'aucun algorithme ne peut faire cette différence !

Plusieurs modes d'appauvrissement sont envisageables, qu'il faudra choisir en fonction de l'image à traiter et des besoins. Le nombre de niveaux de couleurs peut être réduit, éventuellement en utilisant une table d'indexation, selon des méthodes très simples... ou très complexes. Le résultat peut se révéler très convaincant, mais au prix de temps de calcul parfois très longs. Également, le nombre de points de la définition initiale devra être adapté au

mode de restitution utilisé, et aux besoins liés à l'image concernée (par exemple, remplacement de blocs de 2×2 pixels par leur valeur moyenne).

Rappelons que la numérisation du son est obtenue en découpant le signal en fines tranches de temps (échantillonnage, généralement à 11, 22 ou 44 kHz) puis en l'évaluant par rapport à une échelle numérique de référence (quantification) [Dos. Ing., n°13, 1993]. La numérisation d'une image fixe, quant à elle, s'effectue au travers d'un échantillonnage en zones géographiques sur la surface de l'image, suivi d'une quantification par rapport à une échelle de représentation du signal lumineux ; cette représentation peut se faire par rapport aux couleurs primaires RVB, comme c'est généralement le cas en informatique, ou en séparant les informations de luminance et de chrominance, selon des techniques couramment pratiquées en vidéo (Y/C, YUV, YCC) [Cavet, 1994].

Dans bien des cas, l'information obtenue après numérisation peut se révéler trop fine pour la chaîne de restitution envisagée. Par exemple, les dispositifs de visualisation ne disposent pas tous des 24 bits généralement utilisés pour coder les couleurs, et la qualité des haut-parleurs utilisés rend souvent tout à fait superflue une définition d'enregistrement de type CD-Audio. Une analyse soignée de la pertinence des informations conservées, tenant compte de l'ensemble de la chaîne jusqu'à la restitution finale, permet de réduire considérablement la quantité des données conservées, sans dégradation lors de la restitution. Pourtant, une telle opération consiste déjà en une altération de la source, et relève donc bien d'une compression dégradante. En outre, ce type de traitement ne peut se concevoir que dans le cas de l'élaboration d'un support de diffusion, adapté à une chaîne de restitution précise ; dans le cas d'un archivage, en revanche, il sera nécessaire de prévoir une définition sensiblement plus fine, pour prévenir des besoins futurs.

D'autre part, nombre d'informations, si elles sont technologiquement restituables, sont peu ou pas perceptibles à l'oreille ou à l'œil humains. La prise en compte de la sensibilité de nos organes perceptifs permet, là encore, de minimiser les flux d'informations. Cette méthode a été utilisée très tôt en téléphonie, mais également en vidéo : l'œil étant plus sensible à la luminance qu'à la chrominance, les codages YCC et YUV notamment codent beaucoup moins finement les couleurs que l'intensité lumineuse elle-même (sous-échantillonnage de la chrominance).

Dans le cas d'images animées, on sait, depuis les origines du cinéma, que le mouvement est rendu par une succession d'images fixes à une fréquence suffisante pour la perception de l'œil humain (généralement 25 images par seconde). Chacune de ces images doit donc être traitée de la même manière qu'une image fixe lors de la numérisation. Pourtant, là encore, la quantité de données dépendra de la technologie choisie. Les paramètres retenus pour la numérisation de chaque image devront entrer en ligne de compte, mais aussi le nombre d'images par seconde (25 avec les procédés vidéo européens, 30 en NTSC), et la technique de restitution (affichage par trames entrelacées par exemple).

Cependant, s'il est possible d'adapter les paramètres de numérisation ou de codage lorsqu'on connaît les caractéristiques de l'ensemble de la chaîne de restitution, ainsi que les utilisations qui sont envisagées (démarche éditoriale), cette démarche est beaucoup risquée lorsque le but de l'opération est l'archivage. Dans ce cas, en effet, on ignore quelles seront les performances des matériels utilisés dans quelques années, et, surtout, il est probablement tout à fait déraisonnable d'abandonner des informations alors que l'on ignore le type d'exploitation qui sera fait, quelquefois des années plus tard. Ainsi, le Photo CD conserve des images numérisées dans une définition seize fois plus fine que celle qui est nécessaire pour un téléviseur, pour garantir les possibilités d'utilisations futures ; et, dans bien des cas, on considère que le meilleur moyen de conserver des films ou des séquences vidéo reste les supports traditionnels argentiques ou magnétiques, tous encore analogiques.

Avant même d'envisager un algorithme de traitement, capable de compresser de manière optimale une grande variété d'images ou de sons, c'est à une adaptation des données de numérisation que l'on doit se livrer. Les capacités techniques de la chaîne de restitution, les capacités sensorielles de l'oreille ou de l'œil humain et les exigences de qualité qui peuvent varier selon que l'objectif est l'archivage ou l'édition, sont des éléments à prendre en compte dès le stade de la détermination des paramètres de numérisation.

Les méthodes prédictives : compression différentielle et différentielle adaptative

Dans le cas de sons numérisés, on peut constater que les niveaux de deux échantillons successifs présentent statistiquement une différence faible. On dira qu'il y a une forte corrélation entre le signal obtenu sur un échantillon donné et à l'échantillon précédent, ou encore, au sens de la théorie de l'information, que ses informations présentent une forte redondance.

De même, dans le cas des images, il est aisé de constater qu'il y a une forte corrélation spatiale entre un pixel et la zone qui l'entoure : dans la très grande majorité des cas, les images, naturelles ou de synthèse, présentent de larges zones dont les couleurs sont très proches (ciel, mer, constructions, etc.).

Pour les images animées, quiconque a regardé par transparence un morceau de film a pu constater que les images successives sont très souvent quasi semblables. À la corrélation spatiale évoquée plus haut s'ajoute donc une corrélation temporelle, sans doute encore plus forte.

Les méthodes différentielles : procédés

Méthode différentielle simple

Une première étape pour réduire la taille des données est de transmettre non plus la valeur de l'échantillon, mais celle de la différence avec l'échantillon précédent, qui est statistiquement inférieure et peut donc être codée sur un nombre de bits plus réduit. L'utilisation de ce principe pour un codage non dégradant (entièrement réversible) suppose cependant un codage des différences entre deux échantillons avec un codage à longueur variable, à la manière de la méthode de Huffman. Or, une telle méthode, qui suppose une analyse statistique préalable de l'ensemble des données, n'est pas adaptée à ce type d'utilisation : d'une part, elle ne peut être appliquée en temps réel (analyse et compression en deux passes), et, d'autre part, elle nécessite des traitements de décodage plus complexes qui peuvent se révéler incompatibles avec les matériels de restitution grand public envisagés.

La restitution du son pouvant souvent s'accommoder de quelques pertes d'informations, la solution généralement retenue est un codage de toutes les données selon un nombre identique de bits, plus réduit que la définition originale de la quantification. Par exemple, dans le cas de sons numérisés sur 8 bits, le codage différentiel sera effectué sur seulement 4 bits, ce qui divise le flux de données par deux. En revanche, les sauts d'amplitude supérieure à 4 bits entre deux échantillons successifs ne pourront être fidèlement restitués, mais ils sont statistiquement peu fréquents.

Exemple



Dans le cas d'une numérisation du son sur 8 bits (256 niveaux), à 22 kHz, un codage différentiel sur 4 bits (16 niveaux) occasionnera au maximum un retard de 16 échantillons pour transmettre une variation d'amplitude maximale, maintenue durant une durée égale au moins à 16 échantillons. Un tel évènement est fort improbable, et le retard moyen constaté (et donc la perte de qualité) se révèle proche de zéro¹. Les données transmises sont équivalentes en volume à un codage à 11 kHz sur 8 bits, avec une qualité de restitution généralement bien supérieure.

Dans le cas des images fixes, on peut également utiliser une méthode différentielle, par ligne, ou par colonne. Chaque pixel sera alors codé à partir de la différence avec le pixel précédent, et on peut alors appliquer les méthodes Huffman ou LZ aux données obtenues. Ce codage, largement utilisé par le passé, suppose néanmoins des traitements assez lourds (calcul des différences et codage LZW) et se révèle de moins en moins performant lorsque le nombre de couleurs augmente. Dans ce cas, il est nécessaire de réaliser un codage en 3 plans monochromes (RVB ou YCC) pour obtenir de bons résultats.

On peut également citer le format DYUV exploité par le CD-I, qui ne code totalement que le premier pixel de chaque ligne, les autres étant codés par différence avec l'élément précédent.

Méthode différentielle adaptative

Le procédé peut être amélioré en réalisant une "prédiction" plus précise que la simple prise en compte de la valeur de l'échantillon précédent. Cela peut par exemple, dans le cas du son, être réalisé par extrapolation de la vitesse de variation du signal sur les échantillons précédents. La valeur délivrée par le prédicteur est alors celle de l'échantillon précédent, corrigée en tenant compte de la moyenne des variations constatées sur quelques échantillons passés. Le codage ainsi obtenu permet, avec un même nombre de bits de codage, des altérations largement plus réduites qu'avec la méthode différentielle simple.

La compression d'images peut également être réalisée en calculant une valeur théorique probable pour un pixel à partir des pixels voisins, et en codant la différence entre l'état réel du pixel et cette prédiction. Les méthodes classiques de compactage, comme celles de Huffman ou de Lempel-Ziv, permettent alors de réduire la taille des données obtenues. Dans le cas le plus simple, la prédiction est réalisée à partir de la moyenne des pixels de l'image, le coefficient de corrélation des pixels consécutifs, et l'unique valeur du pixel précédent.

L'algorithme obtenu est disymétrique : le codage, qui comprend une analyse préalable de la totalité de l'image, se révèle largement plus complexe que le décodage. Des procédés plus élaborés peuvent être utilisés en étendant l'environnement à un plus grand nombre de pixels (souvent 8), ainsi qu'en découpant l'image de base en plusieurs images plus réduites (souvent 8x8 pixels), ce qui permet de réduire l'écart type.

Pour les images animées, les procédés d'extrapolation utilisés pour le son peuvent permettre de prédire le mouvement, mais il est souvent plus efficace de le traduire sous forme d'un vecteur obtenu par interpolation entre deux images distantes. Cependant, dans ce cas, les images doivent être traitées dans un ordre différent de celui de la restitution, ce qui contribue à rendre les traitements plus complexes, notamment en ce qui concerne le codage.

Performances

Dans le cas de la compression de données sonores, les codages différentiels et différentiels adaptatifs (DPCM et ADPCM) offrent des résultats satisfaisants. La qualité de restitution est directement fonction des paramètres de numérisation, en particulier la fréquence d'échantillonnage et le nombre de valeurs disponibles sur l'échelle de quantification (8 bits :

¹ Il est cependant possible de trouver des sons qui s'adaptent très mal à cette manipulation.



256 valeurs ; 16 bits : 65 536 valeurs). Un codage DPCM ou ADPCM sur un nombre de bits divisé par 2 offre un encombrement équivalent à une fréquence d'échantillonnage divisée par 2, avec une qualité largement supérieure.

Dans tous les cas, le rapport de compression envisageable sans dégradation sensible ne peut dépasser 10. En effet, le signal sonore est unidimensionnel et présente une corrélation beaucoup moins forte que les images.

La compression d'images selon ces méthodes donne des gains de compression temporelle très variables, selon la nature des données (entre 20 et 80 %), mais qui se détériorent lorsque le nombre de couleurs augmente. Les algorithmes prédictifs introduisent la notion de traitement dissymétrique, le codage étant beaucoup plus complexe (et long) que le décodage.

Implémentations

Sons : les méthodes DPCM et ADPCM sont largement utilisées en téléphonie, pour laquelle elles ont été élaborées. Elles sont reprises dans le cadre des spécifications *Windows Multimedia* (DPCM), CD-ROM XA et CD-I (ADPCM).

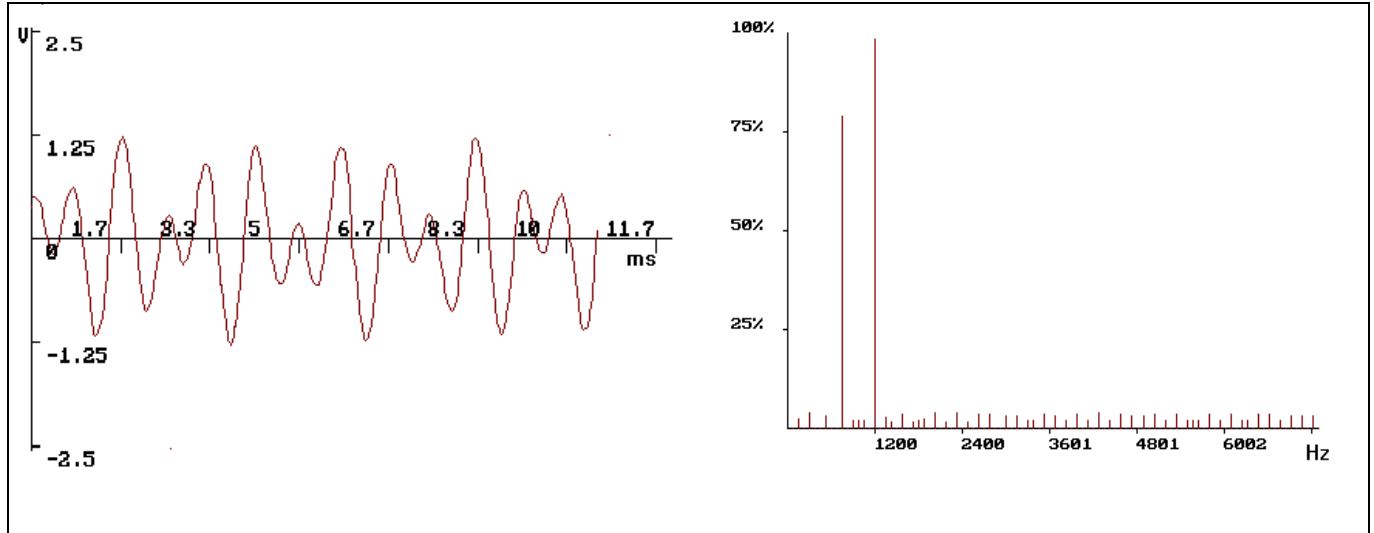
En ce qui concerne les images, le format GIF, développé par *Comuserve*, utilise la méthode différentielle, associée aux méthodes de compactage traditionnelles LZ et Huffman.

Pour réduire la quantité d'informations à transmettre, les procédés différentiels codent non plus chaque valeur, mais la différence avec une valeur probable, calculée par un processus de prédiction.

Selon le degré de complexité du procédé, la valeur prédite peut être obtenue très simplement (l'échantillon précédent, ou le premier pixel de la ligne peuvent servir de référence), ou en tenant compte de diverses informations, comme les caractéristiques globales de corrélation liées au type des données, ou les valeurs et variations enregistrées sur les données "voisines".

Au secours : les maths reviennent !

L'utilisation de transformations mathématiques permet de remplacer les valeurs produites par la numérisation par des données de nature différente qui peuvent se révéler plus faciles à compresser.



Signal vocal de numérotation téléphonique (touche 4) et sa décomposition spectrale obtenue par transformation de Fourier²

Parmi les transformations mathématiques les plus connues, la transformation de Fourier permet de traduire une variation de signal par rapport au temps dans le domaine fréquentiel (décomposition spectrale). On démontre en effet que tout phénomène oscillatoire peut être décomposé en une combinaison de fonctions trigonométriques simples (décomposition en série de Fourier) ; chacune d'entre elles est alors caractérisée par des coefficients (amplitude et fréquence), qui peuvent être représentées par un diagramme en batonnets des différentes fréquences présentes.

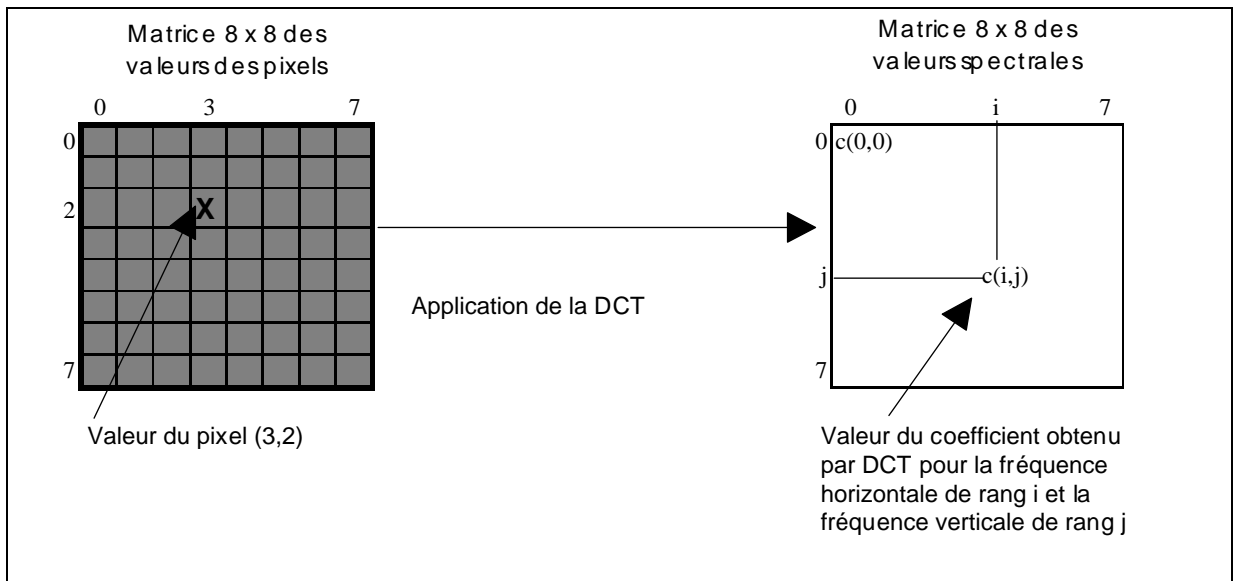
Concernant des signaux numérisés, qui sont des représentations discrètes (le nombre de valeurs par unité de temps est fini et connu) de phénomènes oscillatoires, on peut utiliser la transformation de Fourier discrète (*Fast Fourier Transform* ou FFT). Elle exploite les mêmes principes que la décomposition en série de Fourier, mais, contrairement à elle, produit une décomposition en une somme finie de valeurs fréquentielles, ce qui lui donne la propriété d'être entièrement réversible. En pratique, on définit aussi la transformation inverse (IFFT ou *Inverse Fast Fourier Transform*) qui restitue fidèlement les données initiales. En outre, les calculs nécessaires à cette transformation sont une succession de multiplications et d'additions, très rapidement exécutables par un processeur simple.

La transformation cosinus discrète, ou DCT

Dans le cas d'images fixes, il s'agit d'analyser un signal ne variant plus en fonction du temps, mais bien en fonction de la position du pixel dans le pseudo-plan de l'écran ; c'est une transformation de même nature que la transformée de Fourier qui est invoquée, mais opérant en fonction des coordonnées X et Y du pixel. La Transformation Cosinus Discrète (DCT ou *Discrete Cosine Transform*) permet de décomposer le signal lumineux composant l'image (X et Y pour les coordonnées du pixel, Z pour sa définition lumineuse) selon une combinaison de fonctions trigonométriques, identifiables par leur fréquence et leur amplitude³.

² D'après le logiciel *Spector*, développé par le groupe Évariste (distribution Micrelec).

³ Encore ne s'agit-il pas ici de fréquences par rapport au temps, mais par rapport à la variation de X et Y.



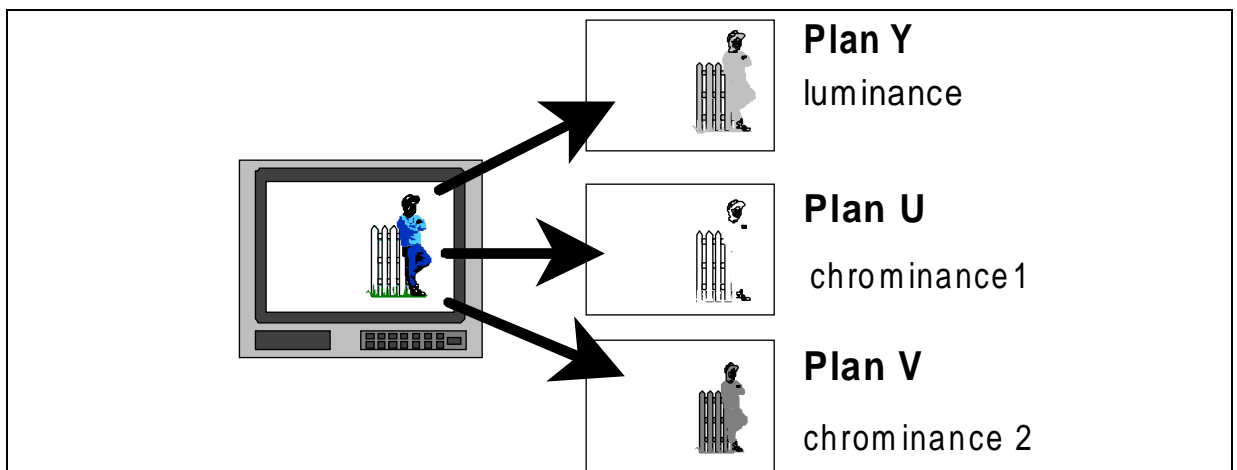
Application de la DCT à un pavé de 8 pixels de coté.

Elle traduit cette information spatiale en une information fréquentielle (spectrale), X et Y représentant alors les fréquences du signal dans ces deux dimensions (horizontale et verticale).

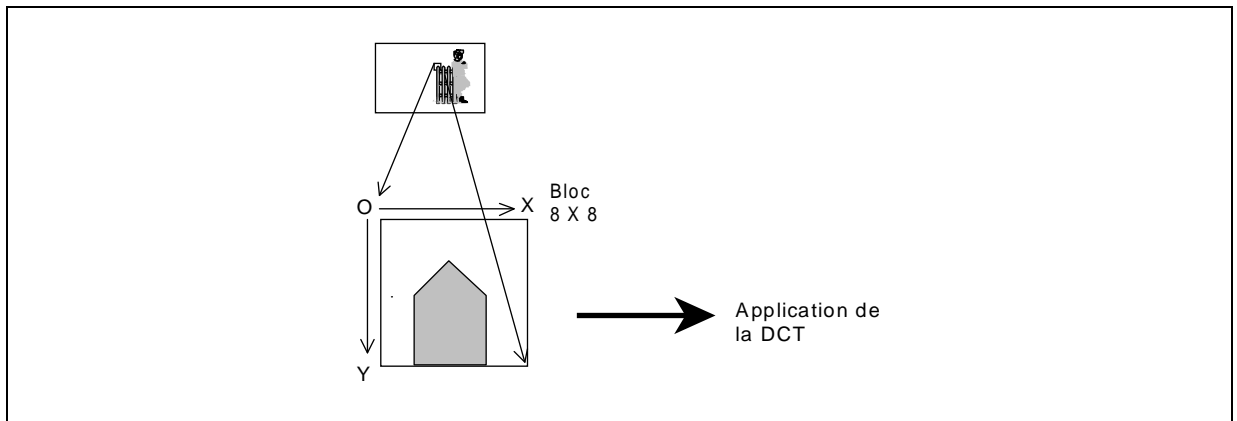
Comme la FFT, la DCT est une transformation entièrement réversible (IDCT, inverse de la DCT restitue fidèlement les données initiales), et son application n'est à l'origine d'aucune perte d'information.

Les transformations mathématiques les plus utilisées dans les méthodes de compressions sont des décompositions fréquentielles (ou spectrales) du type de la transformation de Fourier. Dans le cas des images, il s'agit de la DCT, qui établit une décomposition selon les fréquences verticales et horizontales.

En pratique, l'image est généralement codée en YUV, et traitée comme trois "images" monochromes ou plans. La DCT est alors appliquée successivement aux données du plan de luminance (Y) et des 2 plans de chrominance (U, V). Pour limiter le nombre d'opérations à effectuer, la transformation est appliquée à des zones rectangulaires de l'écran, généralement de 8 x 8 pixels.



1 : décomposition d'une image en 3 plans YUV

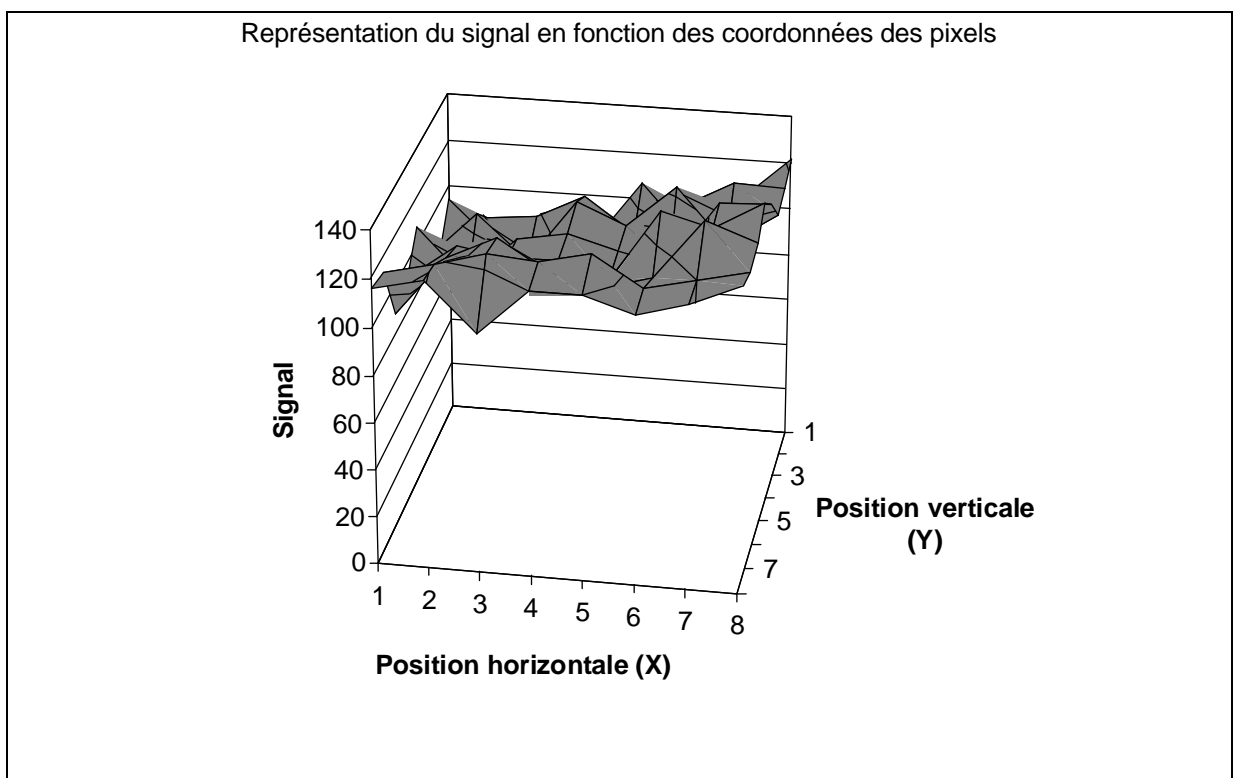


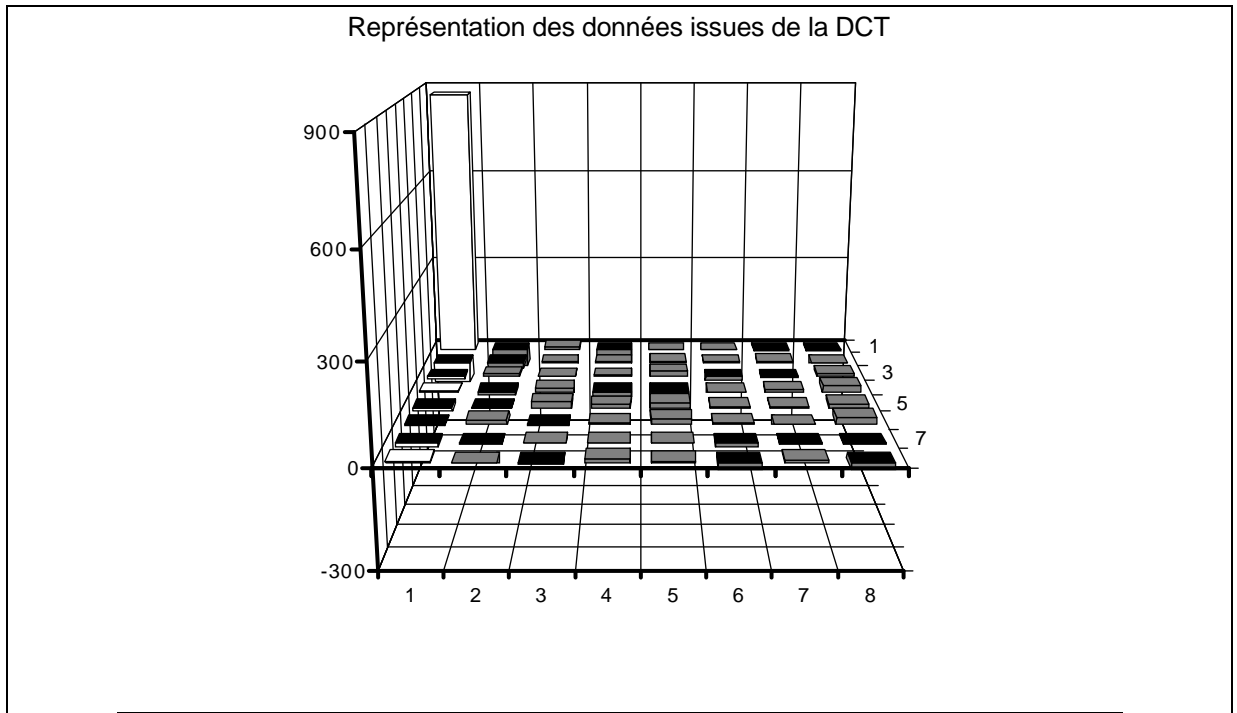
2 : découpage de chaque plan en pavés de 8 x 8 pixels et application de la DCT

La DCT produit elle même une matrice de 64 coefficients (8 x 8), rangés selon les fréquences croissantes selon les axes horizontal et vertical. Généralement, la plus grande partie de l'énergie est concentrée dans les basses fréquences du spectre qui sont exprimées par les coefficients les plus proches du coin supérieur gauche de la matrice.

94	74	88	80	107	84	103	122	872	-59	9	-19	1	0	-7	-4
78	96	98	108	93	107	118	117	-64	-17	2	3	6	3	6	1
86	106	87	114	105	123	100	114	-8	9	1	4	18	-10	-5	11
107	86	105	109	103	98	126	127	3	-12	10	-13	-10	0	6	20
94	109	106	106	99	130	124	136	-11	-3	19	9	17	4	4	11
88	113	123	110	105	111	136	128	-2	14	-3	4	18	4	3	22
114	119	125	123	128	115	120	125	-10	-3	1	3	1	-9	-3	-4
116	121	101	120	120	113	119	128	6	1	-4	11	3	-20	8	-17

Matrice des valeurs de luminance des pixels et sa transformée DCT [Terrasson, 1992]





*Représentations d'une matrice de pixels et de sa transformée DCT.
On remarque clairement l'importance de la valeur supérieure gauche de la matrice,
qui correspond à la moyenne des pixels de la zone.*

$$DCT(i, j) = \frac{1}{\sqrt{2N}} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} Pixel(x, y) \cos\left[\frac{(2x+1) i \Pi}{2 N}\right] \cos\left[\frac{(2y+1) j \Pi}{2 N}\right]$$

$$Pixel(x, y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i) C(j) DCT(i, j) \cos\left[\frac{(2x+1) i \Pi}{2 N}\right] \cos\left[\frac{(2y+1) j \Pi}{2 N}\right]$$

$$\left(\text{où } C(u) = \frac{1}{\sqrt{2}} \text{ si } u = 0 ; C(u) = 1 \text{ si } u \neq 0 \right)$$

(Pixel (x,y) désigne la valeur du pixel de coordonnées (x,y) et
DCT(i,j) le coefficient repéré par la ligne i et la colonne j dans la matrice DCT.

*La DCT existe : je l'ai rencontrée !
Transformations DCT et IDCT appliquées à un pavé de N x N pixels*

La compression sera réalisée ultérieurement sur cette matrice en remplaçant progressivement les coefficients les moins significatifs (ceux proches du coin inférieur droit de la matrice) par des approximations de plus en plus grossières, selon des échelles de quantification bien définies ; pour des gains supérieurs, on ignorera les valeurs les plus proches du coin inférieur droit de la matrice. Le décodeur utilisant alors un intervalle de valeur et non la valeur exacte, la précision des valeurs du bloc de pixel reconstruit par l'IDCT est réduite.

Le coefficient supérieur gauche de la matrice (coordonnées 0,0) représente une fréquence horizontale et verticale nulle ; il est proportionnel à la valeur moyenne des pixels du bloc traité et permet donc à lui seul une prédiction du bloc de 8 x 8, voire la reconstruction d'une image de prévisualisation par pavés de 8 x 8 (compression théorique du bloc N x N : 1/N²). Ainsi, la DCT permettra d'effectuer aisément une compression dégradante en éliminant progressivement les informations les moins pertinentes.



Les transformations fréquentielles sont entièrement réversibles (au moyen de leur transformation inverse), et ne produisent donc par elles-mêmes aucune compression. Celle-ci est réalisée dans un deuxième temps par une quantification sur les fréquences.

Les transformations FFT et DCT sont particulièrement adaptées pour un usage temps réel car elles ne sont constituées que d'opérations simples (multiplication, addition, division). Cependant, le nombre d'opérations élémentaires nécessaires au calcul d'une matrice DCT croît largement plus vite que le nombre de pixels du pavé à traiter. Il est donc utile de limiter l'importance des zones traitées, généralement 8 x 8 pixels.

Vers d'autres transformations mathématiques

Après la FFT et DCT, les ondelettes !

Si les transformées de Fourier et autres DCT rendent de considérables services, elles sont à l'origine plus adaptées aux traitements de signaux continus stationnaires (par exemple les signaux utilisés en radio), et seules des versions dérivées appauvries sont utilisées en traitement numérique. Dans le secret des laboratoires de recherche, on travaille aujourd'hui à de nouvelles transformations qui décomposent le signal non pas en fonctions sinusoïdales simples, mais en des fonctions moins régulières, les “ fonctions en ondelettes ”. Il s'agit ici de décomposer le signal en une combinaison de morceaux d'ondes, ce qui équivaut à un filtrage en sous-bandes, utilisé dans les techniques de compression audio.

Bien plus adaptée au traitement des images numériques, les transformations en ondelettes permettront sans doute dans un avenir proche d'obtenir de bien meilleurs taux de compression, avec une moindre dégradation. Cependant, l'utilisation de telles méthodes suppose dans bien des cas la réalisation de processeurs spécialisés ; compte tenu de l'antériorité des méthodes fondées sur la DCT, et de la banalisation des processeurs correspondants, on peut se poser des questions sur l'avenir de méthodes alternatives, aussi prometteuses soient-elles.

Fractales : le retour !

La décomposition fractale des images est une autre piste souvent évoquée par les chercheurs. Il s'agit ici de déterminer point par point une ou des fonctions mathématiques particulières (les fonctions fractales) dont les valeurs approcheront au mieux l'image initiale. Si le traitement initial est important (il s'agit de trouver point par point la fonction qui approche le mieux le signal), la décompression est de nature toute différente puisqu'il s'agit alors “ seulement ” d'appliquer la fonction trouvée.

Essentiellement dissymétriques, les algorithmes fractals semblent pouvoir offrir des gains de compression considérables, sous réserve de disposer de capacités de calcul suffisantes (la compression d'une image superVGA peut prendre jusqu'à une dizaine de minutes avec un 486 DX 33).

Les premiers logiciels et matériels de compression fractale sont commercialisés aux Etats Unis, et même importés, quoiqu'un peu confidentiellement, en France. Ils sont aujourd'hui capables de traiter une image 640 X 480, 16 millions de couleurs en quelques minutes pour la compression, alors que l'affichage se fait plus rapidement que pour les images JPEG équivalentes (moins d'une seconde). Un taux de compression de 1 % (!), vérifiable par la taille du fichier (de 900 ko à 9 ko), permet d'afficher une image sans dégradation notable, alors qu'une compression JPEG dans les mêmes proportions ne permet la restitution que d'un vague damier.



Encore convient-il de préciser que ce procédé ne convient pas à tous types d'images. Une particularité de cette méthode est qu'elle produit un codage entièrement vectorisé, et ne conduit donc à aucun effet de pixélisation. Il est même possible de restituer l'image à une définition supérieure à celle de départ (généralement 2 fois plus fine) ! Bien entendu, les points intermédiaires sont produits par calcul, à partir de la fonction fractale obtenue, sans rapport direct avec l'image de départ. Si l'image traitée est suffisamment "régulière", le résultat peut être étonnant, mais il peut également se révéler décevant dans le cas de contours très finement ciselés.

Le logiciel *Images Incorporated*, de la société Iterad System Inc, est disponible auprès de la société Image etc.⁴ (environ 3 500 F HT), qui diffuse également une boîte à outils de développement (*Color Box Compression SDK*). Une carte d'extension, permettant une compression plus rapide, est également proposée.

Ce procédé a été notamment utilisé par Microsoft pour l'encyclopédie *Encarta*. Il permet de stocker un très grand nombre d'images sur un support (par exemple, 10 images 640 X 480 en 16 millions de couleurs sur une disquette), mais requiert un logiciel spécifique pour la décompression. L'absence de normalisation de cette solution reste cependant une limite à son exploitation.

Des transformations mathématiques d'une autre nature que les décompositions fréquentielles peuvent également conduire à des compressions performantes. On cite, par exemple, les fonctions fractales ou "en ondelettes". Cependant, ces nouvelles techniques se heurtent à la généralisation des transformations spectrales (DCT), qui ont déjà donné lieu au développement de circuits spécialisés.

Les processeurs de traitement du signal

Tous les appareils numériques, même les plus simples, nécessitent un traitement de nature informatique pour soumettre un signal à un périphérique de restitution qui est le plus souvent... analogique (haut-parleur, moniteur). Ainsi, les lecteurs de disques laser et autres consoles de jeux font appel à des circuits spécialisés, les processeurs de traitement du signal ou DSP (*Digital Signal Processor*).

Comme son nom l'indique, le processeur de signal numérique intervient pour traiter un signal issu, pour tous les dispositifs de mesure physique, d'un capteur qui fournit une variation de tension, donc de nature analogique. Avant de l'exploiter de manière numérique, il est donc nécessaire de le coder à travers un convertisseur ADC (*Analogic Digital Converter*), puis de le transmettre au DSP. Tous les calculs nécessaires à un traitement de signal numérique pourraient être effectués par un processeur traditionnel, mais avec des performances catastrophiques. En effet, une simple multiplication demande près de 20 cycles d'horloge sur un processeur de type Intel i386 ou Motorola 68 030, alors qu'un multiplieur câblé n'utilisera que deux cycles. De tels dispositifs existent dans les coprocesseurs arithmétiques, intégrés ou non aux processeurs principaux (i486, Pentium), mais ils sont alors conçus pour des calculs à virgule flottante, alors que les signaux numériques n'utilisent que des entiers. Des composants spécialisés ont donc été développés et se sont rapidement répandus avec la diffusion considérable des équipements numériques (CD-Audio, consoles de jeux, etc.). Leur application la plus connue est le traitement du son dans les CD-Audio, mais ils sont souvent également utilisés en télécommunications (télécopie, transfert de données, liaison Numéris) et de plus en plus pour les applications informatiques exploitant des données numérisées.

⁴ Image etc., BP 322, 33695 Mérignac Cedex. Tél : 56 34 79 34 ; Fax : 56 13 01 80.



Les diverses fonctions que peut avoir un DSP relèvent généralement de l'analyse et du filtrage du signal, et éventuellement de sa compression, lorsqu'elle n'est pas entièrement logicielle ou confiée à un autre processeur spécialisé. La plupart des calculs de filtrage ou d'analyse du signal ne sont pas effectués sur le signal lui-même, mais sur sa décomposition spectrale, obtenue par des transformations de type FFT ou DCT. Les DSP sont donc conçus pour s'adapter au mieux à ce type de transformation, et calculent aujourd'hui une FFT de 10 à 20 fois plus vite qu'un processeur traditionnel. En particulier, ces traitements faisant appel à de multiples séquences de multiplications et additions d'entiers, les DSP sont dotés d'une unité arithmétique de conception spécifique, le MAC (*Multiply And Accumulate*) capable d'effectuer un produit et de l'ajouter au résultat précédent en un seul cycle d'horloge.

De même, la formule de la DCT citée plus haut montre à l'évidence que son exécution (mais c'est le cas pour bien d'autres traitements confiés au DSP) doit être réalisée par de nombreuses boucles, chaque itération étant constituée d'un nombre réduit d'opérations. La gestion du compteur de boucle, qui constitue une partie non négligeable du traitement, est confiée à un module spécialisé qui permet de décharger le MAC de cette tâche.

Pour accéder à ces diverses fonctions, les DSP de dernière génération sont dotés d'un jeu d'instructions spécifique : ce sont des processeurs programmables (tout au moins à un niveau élémentaire), et il devient ainsi possible de les utiliser pour effectuer des traitements variés, de les adapter à l'exécution de multiples algorithmes. Aujourd'hui réservés à des équipements informatiques particuliers (cartes sonores, équipement de digitalisation/restitution), ils équiperont demain directement la carte mère des micro-ordinateurs, à l'image des Macintosh AV. Plus tard, progressivement, ces circuits seront directement intégrés au processeur principal.

Les processeurs spécialisés dans les traitements de l'image comme le i750 d'Intel ou le C450 de C-Cube sont en fait de supers DSP, intégrant l'équivalent de plusieurs DSP classiques, mais aussi des convertisseurs ADC/DAC⁵, et des circuits logiques câblés pour la mise en œuvre d'algorithmes préprogrammés. Les plus récents disposent même de larges possibilités de programmation pour exploiter de nouveaux algorithmes.

Le traitement du signal est caractérisé par une grande quantité d'opérations simples ; les processeurs des ordinateurs, trop complexes, sont peu adaptés pour effectuer ces traitements qui sont de plus en plus souvent confiés à des processeurs spécialisés, les DSP.

Les méthodes modernes : les images fixes JPEG / JBIG

Le CCITT et l'ISO ont réuni dans les années quatre-vingt le groupe JPEG (*Joint Photographic Experts Group*) qui a permis l'adoption en 1992 de la norme ISO/CEI 10 918, plus connue sous le nom de norme JPEG. Dans le même temps, des travaux comparables conduisaient à l'adoption de la norme ISO/CEI 11 544, plus connue sous le nom de JBIG.

JPEG et JBIG sont des ensembles de spécifications et algorithmes de compression des images fixes. Leur statut de norme officielle constitue une évolution essentielle puisque cela permettra sans doute la réalisation de processeurs spécialisés à bas prix, fabriqués à grande échelle, qui pourront être intégrés dans de nombreuses machines.

Chacune de ces normes est en fait composée de plusieurs techniques de compression, incluant notamment des spécifications pour un codage conservatif (entièrement réversible) et pour un autre dégradant. Dans le second cas, il s'agit de codages adaptables, dont le comportement peut varier en fonction des images à traiter, mais aussi des contraintes liées au matériel utilisé.

⁵ Convertisseurs digital/analogique et analogique/digital.



Le cahier des charges défini à l'origine des travaux du comité JBIG ciblait davantage les besoins d'un puissant algorithme pour la transmission des images, notamment en télécopie. Ce procédé est donc particulièrement adapté pour les images monochromes, ainsi que pour les images codées sur un petit nombre de couleurs. En outre, ce codage prévoit d'une part une restitution séquentielle, ligne à ligne, particulièrement adaptée à la télécopie, et d'autre part une décompression progressive de qualité croissante, par pavés (imagelette, définition écran, ou haute définition pour l'impression).

JPEG propose également un mode progressif et un mode séquentiel, mais il est originellement prévu pour des images en 16 millions de couleurs (24 bits par pixel). Il a été élaboré pour permettre aussi bien une exécution logicielle que la réalisation de circuits spécialisés.

Les travaux des comités ISO ont montré que les algorithmes JBIG donnent de meilleurs résultats que JPEG lorsque chaque pixel est codé sur moins de 8 bits.

Dans les deux cas, l'image est découpée en un damier, chaque zone obtenue étant ensuite traitée indépendamment. Chaque pavé ainsi obtenu (généralement 8×8 pixels) est considéré comme une matrice de pixels sur laquelle peut être appliquée la transformée DCT.

La technique conservative combine les techniques de codage prédictif/adaptatif avec les méthodes statistiques de Huffman, qui produit une compression conservative optimale (taux de 50 % environ). Les travaux du groupe JPEG sont pourtant essentiellement connus pour les compressions non conservatives.

Procédé

Les algorithmes JPEG et JBIG de compression non conservative sont fondés sur les propriétés de la transformation DCT.

Les images couleur sont généralement codées comme trois images monochromes. Le codage de base des données représentant chaque pixel ne fait l'objet d'aucune spécification, mais c'est le YUV qui est généralement utilisé. L'image est tout d'abord découpée en zones de dimensions réduites (8×8 pixels), puis chaque pavé est traité par la DCT. La compression, selon un taux laissé au choix de l'utilisateur, est effectuée par quantification des coefficients issus de cette décomposition spectrale. Cette étape est la seule à l'origine des dégradations. Enfin, les données obtenues sont codées sous une forme elle-même compressée au moyen des algorithmes conservatifs décrits plus haut (Huffman essentiellement).

Performances

Les compressions/décompressions successives sont toujours dégradantes, avec un effet cumulatif, et il convient donc de limiter ces opérations. On estime généralement qu'une image JPEG ne contenant aucune information superflue (définition en nombre de points et de couleurs conforme au dispositif de restitution) tolère sans dégradation notable un taux de compression de l'ordre de 1 pour 10 ou 13. L'affichage d'une image "plein écran" peut prendre environ 1 seconde sur un Macintosh 68 040 ou un PC Intel i486. En outre, il faut noter que l'extension multimédia *Quicktime* du système d'exploitation du Macintosh supporte le codage JPEG.

Cependant, les images couleur sont généralement codées en "vraies couleurs"⁶ sur 24 bits par pixel, alors que les dispositifs d'affichage ne sont pas aussi performants (mémoire vidéo insuffisante notamment) ; l'affichage en un nombre de couleurs plus réduit (4 ou 8 bits) se fait au moyen de trames coûteux en termes de temps de traitement, qui rendent la restitution plus laborieuse.

Implémentations

Une fraction croissante des logiciels de traitement d'images exploitent actuellement le format JPEG, en importation comme en exportation. Près d'une dizaine de processeurs spécialisés sont également

⁶ On désigne ainsi un affichage en 16 millions de couleurs, codé sur 24 bits par pixel (*true color*).



disponibles sur le marché. L'extension *Quicktime* du Système 7 d'Apple assure le traitement JPEG directement au niveau du système d'exploitation.

Il faut également signaler des versions dérivées de JPEG adaptées au codage de la vidéo animée, connue sous le nom de M-JPEG (*Moving JPEG* ou *Motion JPEG*). La méthode retenue consiste en un codage de chaque image selon le procédé JPEG, avec un taux de compression optimal, compte tenu de la bande passante disponible. Ces procédés souffrent cependant d'une absence de normalisation.

Les codages JPEG et JBIG sont aujourd'hui normalisés. Des processeurs spécialisés ont été construits, et le format JPEG sera progressivement traité directement par les systèmes d'exploitation.

JPEG et JBIG sont en fait chacun un ensemble d'algorithmes permettant aussi bien des compressions conservatives que dégradantes. Ils sont fondés sur les transformations fréquentielles calculées sur des blocs de 64 pixels.

Un problème spécifique : la vidéo numérique

Les flux de données nécessaires à la vidéo numérique sont considérables. Ainsi, le codage 4.2.2 nécessite 166 Mbits/s, pour une définition de 720 x 576 sur 16 bits, à 25 images par seconde. L'avis CCIR 601, qui fixe les paramètres de ce codage, retient un débit nominal, comprenant son et données, de 216 Mbits/s, ou encore 270 Mbits/s avec un codage sur 20 bits. De tels débits ne sont bien sûr envisageables que sur des équipements spécialisés de production extrêmement coûteux : un disque dur de 3 Go est nécessaire pour seulement 20 minutes de programme, et encore faut-il que son débit soit suffisant (de l'ordre de 20 Mbits/s après compression).

Cependant, il y a une très forte corrélation spatiale et temporelle entre les données qui composent une séquence. Lorsqu'il n'y a pas de changement de plan, la redondance spatiale et temporelle moyenne d'une séquence de vidéo animée peut être estimée entre 98 et 99 % [Sonovision, Janvier 92, p.18].

Si ce chiffre peut laisser envisager des taux de compression théoriques de l'ordre de 1 à 2 %, il reste bien entendu tout à fait illusoire de les approcher sans dégradation, voire destruction de la structure même de la séquence. D'une part, une partie des informations redondantes reste toujours indispensable à l'interprétation du contexte, et, de plus, les algorithmes mis en œuvre ne seront jamais parfaits. D'autre part, les contraintes de l'exploitation numérique en temps réel (conservation de l'isochronie) rendent souvent impossible l'utilisation de procédés de reprise sur erreur habituellement utilisés en informatique ; la conservation d'informations redondantes permet alors de tolérer un faible taux d'erreurs de transmission.

Les images animées MPEG

Parallèlement aux travaux du JPEG, le comité MPEG (*Motion Picture Expert Group*) a porté ses efforts sur la compression de la vidéo animée. Comprenant des spécialistes de la vidéo, de l'informatique, mais aussi des télécommunications (forte implication des experts du CCITT), le comité MPEG a travaillé sur une norme de compression adaptée à l'exploitation d'images vidéo sur les matériels informatiques. Les contraintes posées concernaient principalement les débits d'information, une exploitation éditoriale supposant une possibilité de diffusion sur CD-ROM. Les travaux ont donc été principalement fondés sur un débit de l'ordre de 1,5 Mbits/s, soit à peu de chose près le débit d'un CD-ROM en mode 2 (170 ko/s).

Le résultat est la norme ISO/CEI 11172, plus connue sous l'appellation MPEG 1, et adoptée par l'ISO en novembre 1993.

Les paramètres ayant servi de base aux travaux du groupe sont cependant actuellement très en retard par rapport à l'évolution de la technique ; la quasi-totalité des lecteurs de CD-ROM disponibles sur le marché propose un débit au moins égal à 300 ko/s, au moyen d'un doublement de la vitesse de rotation. Le comité MPEG poursuit donc ses travaux, et travaille actuellement sur l'évolution MPEG 2 adaptée à des débits de 2 à 10 Mo/s. L'orientation retenue semble retenir une structure beaucoup plus ouverte, capable d'intégrer à l'avenir d'autres algorithmes et de s'adapter aux évolutions du contexte technologique. Parallèlement, un sous-comité travaille également à l'élaboration d'une norme (provisoirement connue sous le terme de MPEG 4) adaptée aux usages en faible débit (visiophonie, par exemple).

Procédé

MPEG 1 traite les images vidéo comme une succession d'images individuelles, dont chacune est représentée comme une matrice de pixels.



Format de numérisation

Compte tenu du débit retenu, de l'ordre de 150 ko/s, la définition du format de numérisation retenu (*SIF : Source Intermediar Format*) est de 360 x 188 pixels. Le codage est réalisé en YCC, soit une information de luminance et de deux de chrominance. Cependant, selon des usages désormais classiques en vidéo numérique, la luminance de chaque pixel est codée à pleine résolution, alors que la chrominance, moins perceptible à l'œil, est codée par pavés de 2 x 2 pixels, soit une définition en chrominance de 180 x 144.

Codage prédictif

L'exploitation de la redondance statistique des informations constituant une séquence vidéo permet de limiter largement les informations transmises. En se fondant sur les valeurs codées auparavant, le codeur et le décodeur peuvent souvent prédire la valeur du pixel suivant. Seule la différence entre le pixel et la valeur prédite doit alors être codée, et elle est généralement proche de 0 (peu de variations entre les pixels). Ainsi, ce n'est pas seulement la corrélation spatiale qui est utilisée, comme pour un codage JPEG, mais également la corrélation temporelle entre des images successives.

Compensation du mouvement

La compensation du mouvement est obtenue par le déplacement d'un bloc de pixel voisin de l'image précédente vers un bloc de l'image en construction, selon un vecteur de mouvement dont les coordonnées sont calculées à partir de la succession des images précédentes. Sur une courte succession d'images, on constate en effet que la plupart des pixels restent souvent inchangés d'une image à l'autre, alors que les autres subissent de légers déplacements par blocs. Les premières stations de compression demandaient une intervention manuelle pour ce codage, rendant l'opération coûteuse, interdisant un traitement en temps réel, et conduisant à un résultat largement dépendant du savoir-faire de l'opérateur. Les systèmes proposés aujourd'hui garantissent un traitement automatique, pratiquement en temps réel, en recherchant chaque pavé de 8 x 8 de l'image précédente dans une zone plus large de l'image suivante (généralement un carré de 32 à 64 pixels de coté).

Transformation spectrale

Appliquée à des blocs de 8 x 8 pixels, la transformation DCT permet d'obtenir une matrice 8 x 8 de coefficients de fréquences horizontales et verticales. Cette étape n'apporte en soi aucun gain en volume, mais permet de "classer" les informations selon des fréquences croissantes, de moins en moins perceptibles pour l'œil humain. La compression sera réalisée ultérieurement au moyen d'un remplacement progressif des coefficients les moins significatifs (définissant les informations les moins perceptibles à l'œil) par des valeurs approchées de plus en plus grossières, codées sur un nombre de bits plus réduit.

Codage de longueur variable

Grâce à un codage statistique dérivé de Huffman, des mots de longueur variable sont utilisés pour coder les signaux, en utilisant les plus courts pour les données les plus fréquentes.

Trois types d'images

MPEG 1 utilise 3 types d'images :

- les images *Intra* (I) sont codées entièrement, sans référence aux autres ni prédiction temporelle ou compensation du mouvement (comparables à des images JPEG). Les images I servent de référence pour le décodage des autres images, qui exploitent la corrélation temporelle ; leur taille moyenne est d'environ 19 ko ; il y a au moins une image I au début de chaque plan, puis toutes les 12 images.



- Les images *Prédictives* (P) sont codées avec une prédiction temporelle à partir des images déjà décodées, éventuellement avec compensation du mouvement. Le codage utilisé est de type différentiel. La taille moyenne d'une image P est d'environ 10 ko.
- Les images *Bidirectionnelles* (B) sont obtenues par interpolation et sont les plus compressées. Elles sont codées à partir de deux images complètes (I ou P), situées dans le temps avant et après l'image B à décoder, à l'aide de vecteurs de déplacement. La taille moyenne des données correspondant à une image B est d'environ 3 ko, mais le décodage n'est possible que si les deux images de référence sont disponibles. Les données codées doivent alors être organisées selon un ordre qui ne correspond plus au déroulement normal des images.

Performances

Les premières réalisations MPEG, et notamment les films sur support CD-I réalisés par Philips pour son procédé de restitution *Full Motion Video (Digital Video)*, ont montré qu'il était possible d'obtenir un niveau de qualité satisfaisant [Banckaert, 1994]. Quoique incomparable avec une qualité vidéodisque, le codage MPEG est capable de produire des images de qualité supérieure à celles d'un magnétoscope VHS.

Les codages MPEG réalisés, tant en mode automatique que manuel, n'utilisent jamais la totalité des possibilités du codage. La qualité obtenue dépend directement du soin apporté à l'opération, et, souvent, à la compétence de l'opérateur. Il n'y a donc pas de tarif de référence pour un codage vidéo MPEG : il dépend directement de la qualité du travail demandé. Dans bien des cas, la numérisation proprement dite est précédée par un prétraitement en régie vidéo numérique, pour réduire la dynamique des images et en simplifier le codage, en adaptant le flux d'informations à la bande passante disponible. À terme, cela induit de nouvelles contraintes lors du tournage qui devront être prises en compte par les vidéastes.

Les spécialistes de l'audiovisuel souhaitent l'avènement d'un MPEG 2 de plus haute qualité et compatible avec les attentes en matière de haute définition, alors que la rapide évolution des technologies informatiques rend actuellement caduque la limitation du débit à 1,5 Mbit/s (les lecteurs de CD-ROM actuels ont un débit de 3 Mbit/s, quelquefois même 6 Mbits/s).

Le codage MPEG 2 sera sans doute sensiblement différent, organisé autour d'un profil principal défini pour le service le plus important, la diffusion de vidéo. D'autres profils secondaires, organisés selon un mode hiérarchique, permettront d'assurer d'autres services, éventuellement avec une qualité dégradée.

Pour éviter les difficultés rencontrées avec la contrainte de débit du CD-ROM retenue comme pivot de MPEG 1, la nouvelle norme sera conçue ouverte et évolutive.

Implémentations

De nombreux constructeurs ont annoncé leur intention d'adopter le codage MPEG en lecture, principalement pour les matériels destinés au grand public. Quelques dizaines de titres sur support CD-I (*Digital Video*) sont disponibles, et le Vidéo CD semble promis à un bel avenir. Quelques cartes de décompression sont également disponibles pour les micro-ordinateurs, mais les produits susceptibles de les utiliser tardent à arriver sur le marché.

L'utilisation de MPEG impose une rapidité d'exécution incompatible avec d'éventuelles solutions logicielles. La société C-Cube a développé un circuit spécialisé, le C-450, qui est utilisé par divers matériels. L'extension *Full Motion Video* du CD-I utilise ce circuit, ainsi que les cartes MPEG dédiées aux stations Amiga et PC (*Real magic*). La diffusion de ce circuit à grande échelle devrait permettre une baisse des prix de ces matériels.

Les équipements de numérisation et compression restent encore d'un prix élevé, exigent une station musclée et un opérateur qualifié. Les derniers matériels présentés présentent néanmoins une évolution sensible à la baisse, avec notamment plusieurs cartes de compression d'un prix compris entre 20 et 30 000 F, utilisables sur un micro-ordinateur de type PC ou Macintosh ayant une configuration adaptée (vitesse de traitement, mémoire, taille des disques). Le prix de revient d'une station de compression complète est au minimum d'environ 150 000 F ; les difficultés de mise en œuvre conduisent pratiquement systématiquement à avoir recours pour le codage à des prestataires de service spécialisés.

La norme MPEG (version 1) a été approuvée en 1993. Elle a été adoptée pour le CD-I et le Vidéo-CD, et des matériels de décompression sont disponibles pour les micro-ordinateurs. L'expérience acquise par les opérateurs de compression et l'apparition de nouveaux matériels de compression largement automatisés et perfectionnés permet à présent de réaliser des documents MPEG de qualité équivalente au VHS, voire supérieure.

Autres techniques utilisées pour les images animées

Parallèlement aux travaux de MPEG, le secteur informatique a développé ses propres solutions pour amener la vidéo sur les écrans des micro-ordinateurs. Les possibilités d'affichage et de traitement permettaient, dès la fin des années quatre-vingt, d'afficher des images (fixes) de haute qualité, et de créer des animations élémentaires.

Quicktime

Apparu en 1991, *Quicktime* a été intégré au Système 7 des Macintosh. Il s'agit d'un environnement de développement et d'exécution qui permet d'associer à des données classiques des fichiers représentant des séquences sonores ou vidéo.

Quicktime comprend essentiellement des formats de données standardisés, des procédés de compression/décompression, et une interface utilisateur spécifique. L'extension système utilisée est fondée sur le principe du maintien de l'isochronie des données, et introduit donc le temps comme élément principal du système d'exploitation. Les fichiers *Quicktime* sont principalement de deux types : une extension du format PICT est destinée à coder les images fixes, avec de nouvelles possibilités de compression et prévisualisation, et le nouveau format MOOV permet d'accéder par un système de pointeurs aux données vidéo, sonores et autres, qui doivent être synchronisées et stockées par ailleurs.

Quicktime est composé de trois modules principaux :

- Le *Movie Manager* assure la lecture et la synchronisation des données exploitées et décide du mode de compression à mettre en œuvre.
- L'*Image Compression Manager* effectue la compression des images selon divers algorithmes. Deux algorithmes propriétaires permettent de s'adapter à diverses contraintes (exploitation directe à partir d'un CD ou sur disque dur, capacité d'affichage et de traitement de la machine). La norme JPEG est également exploitable, en mode matériel ou logiciel selon les possibilités de la machine. *Quicktime* a aussi adopté les algorithmes *Indeo* développés par Intel. L'intégration de MPEG devrait faire l'objet d'annonces prochaines, sous réserve de la commercialisation à grande échelle des extensions matérielles nécessaires.
- Le *Component Manager*, enfin, se charge de l'interface avec les différentes applications qui peuvent faire appel aux services de *Quicktime*.



Quicktime est surtout conçu comme un ensemble de spécifications très ouvert, capable d'intégrer facilement un grand nombre d'évolutions matérielles et logicielles, sans remettre en cause les applications existantes. Enfin, il permet la restitution de séquences sur un Macintosh, sans extension matérielle, pourvu qu'il soit au minimum équipé d'un processeur 68 020 (le nombre d'images par seconde étant fonction des capacités de traitement disponibles).

L'image, longtemps limitée à une petite vignette sautillante au format timbre poste, s'agrandit progressivement avec les nouvelles versions du logiciel et les nouveaux matériels : limitée à un quart d'écran à 15 images/s sur un Quadra 700, elle sera bientôt disponible en plein écran sur les stations Power PC.

Vidéo pour Windows

Vidéo pour Windows est un ensemble logiciel commercialisé par Microsoft et permettant la capture et la restitution de vidéo animée sur PC.

L'exécution de séquences vidéo est possible sans matériel spécifique, alors que la capture doit être faite au moyen d'une carte de numérisation. Une large compatibilité est permise grâce à la définition de spécifications permettant d'intégrer des matériels aux fonctionnalités différentes. Ainsi, à travers un ensemble de pilotes, *Vidéo pour Windows* sera capable de supporter des périphériques aux fonctionnalités diverses, tout en assurant, autant que possible, les fonctions manquantes au niveau logiciel.

Pas plus que *Quicktime*, *Vidéo pour Windows* n'est pas un algorithme de compression. Il s'agit plutôt d'une interface standardisée entre le matériel et les procédés de codage et de compression, qui offre des API (interfaces de programmation) relativement indépendantes du matériel. Cependant, comme Apple, Microsoft a également défini des algorithmes de compression adaptés à différentes situations (*Microsoft Vidéo 1*, *Microsoft RLE compressor*), et intègre ceux proposés par des sociétés tierces, comme *Indeo* d'Intel.

Plus qu'une technique de compression, *Vidéo pour Windows* fournit en fait une plateforme commune sur laquelle pourront s'articuler divers procédés de codage. Actuellement, en l'absence de dispositif spécifique, la qualité est limitée à un format de 320 x 200 à 15 images secondes (processeur Pentium).

On peut penser que, comme cela a été le cas pour les extensions multimédia de *Windows 3.0*, *Vidéo pour Windows* sera intégré aux prochaines versions du système. *Windows NT* est déjà livré avec *Vidéo pour Windows* en standard.

DVI

Intel et IBM ont développé, un processeur spécialisé dans le traitement des images, le i750. Prévu dès la conception pour intégrer de larges possibilités de programmation, il est capable de s'adapter à de nouveaux algorithmes.

Dès 1992, Intel a commercialisé le système DVI, comprenant des cartes de visualisation et acquisition (cartes Actionmedia, codéveloppées avec IBM), et un ensemble d'algorithmes adaptés directement aux contraintes informatiques. Le système est conçu dès l'origine selon un mode hiérarchique : deux types de codages sont prévus, exploitant des algorithmes différents.

Le PLV (*Production Level Video*), fondé sur des principes proches de MPEG, produit des images en 256 x 240 en 25 ou 30 images par seconde, mais nécessite des temps de calcul considérables pour la phase de compression (de 50 à 100 fois la durée du programme en traitement, sur un système informatique complexe et coûteux). La numérisation et la compression devait être confiées à une société spécialisée (en France TeleTota) pour un coût supérieur à 1 000 F la minute.



Le RTV (*Real Time Video*) permet la capture d'images en temps réel avec une qualité sensiblement plus faible.

Intel a actuellement abandonné la diffusion des matériels DVI et privilégie des solutions permettant une décompression logicielle, à travers les algorithmes *Indeo*.

Le procédé *Indeo*

Depuis la commercialisation de DVI, Intel a développé les algorithmes *Indeo* permettant la restitution, soit à travers un équipement de type Actionmedia, soit entièrement par logiciel. Intégrés à *Vidéo pour Windows* et à *Quicktime*, ils permettent une amélioration sensible de la qualité de restitution dans chacun de ces environnements.

Intel commercialise actuellement la carte Smart Video Recorder, entièrement consacrée à la capture de séquences vidéo et à leur compression en temps réel, et exploitant une nouvelle version d'*Indeo* (*Indeo* version 3). La décompression de séquences *Indeo* est à présent entièrement confiée au processeur principal et la qualité de restitution (taille des images, nombre de trames par seconde) est directement liée au type de matériel utilisé.

Ordinateur	format 160 x 120	format 320 x 240
486 SX 25	15 images par seconde	4 images par seconde
486 DX 33	20 images par seconde	5 images par seconde
486 DX2/66	30 images par seconde	8 images par seconde

Performances de restitution d'Indeo 3 en fonction du processeur utilisé

La nature de la société Intel, plus grand fabricant de processeurs destinés à la micro-informatique, également impliqué dans le développement de nouvelles architectures de machines (bus PCI par exemple) laisse clairement apparaître sa stratégie en ce domaine. Le fonctionnement actuellement proposé, même sur les machines de haut de gamme, est peu satisfaisant, mais les nouvelles générations de machines permettront une meilleure prise en compte de la vidéo.

Les analyses de marché conduites par Intel montrent en effet que ni les constructeurs ni les utilisateurs ne sont prêts à intégrer à leur machines des équipements vidéo coûteux. Proposer dès maintenant une vidéo, fût-elle misérable, ne constitue-t-il pas une excellente incitation à... changer de matériel bien vite ?

Indeo est un algorithme de codage et compression destiné à l'exploitation de la vidéo sur micro-ordinateur. Indeo a été conçu par Intel et adopté par de nombreux constructeurs et éditeurs, au premier rang desquels Apple et Microsoft.

En guise de conclusion – provisoire – à propos des images animées

La multiplicité des solutions annoncées et – quelquefois – utilisées laisse apparaître le caractère mal stabilisé de ce secteur. D'une manière générale, on ne peut que constater le manque de finalisation de tout ce qui concerne l'exploitation de données fondées principalement sur le paramètre temps. Les micro-ordinateurs et leurs systèmes d'exploitation ne sont pas encore adaptés à une exploitation " temps réel ", et la prise en compte de ces problèmes relève de solutions spécifiques, non standardisées. On peut cependant noter une évolution majeure avec la sortie des Macintosh AV et Power Macintosh : au-delà de tous les gadgets mis en avant par la grande presse, ils intègrent une gestion avancée en temps réel.

Comme le constate Norbert Paquel [Paquel, 1993], le monde de la vidéo numérisée est aujourd'hui le lieu d'une grande confusion : multiplicités de produits et de standards, rapidité des évolutions, absence de normalisation effective. Le seul fil conducteur permettant d'ordonner – un peu – le chaos est une distinction plus fine des secteurs d'activités.

La transmission de télévision “ haute définition ”

C'est dans le domaine de la télévision numérique que la compression a sans doute le plus grand rôle à jouer dans les années à venir. Sans technique de compression efficace et standardisée, il n'a aucune possibilité de développer un réseau qui, par nature, doit avoir une durée de vie longue (de dix à vingt ans).

Des consultations et travaux de normalisation sont en cours pour ce secteur, dont les implications économiques sont gigantesques.

Si la compression reste actuellement un enjeu considérable pour les sociétés qui proposent des systèmes adaptés à la télédiffusion, pratiquement plus personne ne doute que des procédés efficaces vont apparaître et seront disponibles avant la mise en service des réseaux et autres “ autoroutes ”. Aux États Unis, on parle déjà de 300 à 500 chaînes numériques bientôt disponibles ; même si les solutions technologiques ne sont pas encore en place, la confiance dans leur réalisation prochaine est si grande que l'essentiel des débats se développe autour du thème “ comment les utiliser ”.

Il reste, bien entendu, plus délicat de faire aujourd'hui des choix stratégiques en ce domaine, mais cela ne concerne guère le secteur éducatif. Partie prenante dans le secteur de la télédiffusion, il reste un partenaire mineur et devra s'aligner sur les choix effectués à un autre niveau.

Le choix d'une technique de compression est indispensable à l'avènement de la télédiffusion numérique. Pour l'heure, il n'est pas encore effectué.

La production vidéo ou télévision actuelle

La production de télévision analogique de qualité suppose une image d'excellente définition au départ, et des possibilités de traitement image par image. Dans bien des cas, même si la prise de vues et la diffusion finale se font en analogique, on a recours au numérique pour les traitements intermédiaires. Restant l'apanage d'équipements professionnels pour lesquels on se soucie encore peu d'interopérabilité, les solutions sont variées et souvent propriétaires. Les équipements de “ production studio ”, y compris les matériels de montage numérique, travaillent souvent “ en linéaire ”, sans compression, ou avec un simple compactage conservatif (Huffman, *Run length*, LZxx). Sony et Ampex exploitent le mode 4.2.2, qui suppose déjà une compression par sous-échantillonnage de la chrominance. Pour les versions bas de gamme des matériels de montage virtuel, on utilise également des procédés de compression image par image de type JPEG (variantes Motion-JPEG, non normalisées).

D'une manière générale, les manipulations de production pouvant être nombreuses, un mode de compression dégradant sera à l'origine d'altérations importantes dans le cas de multigénération, indispensable en postproduction.

Pour ce qui concerne les équipements de montage virtuel et de postproduction, la conformité avec une norme n'est pas une nécessité, puisque la compression n'est utilisée que lors du traitement, le produit final étant généralement fourni sous forme analogique. Les choix faits dans ce cadre sont donc sans conséquences sur les autres équipements, et peuvent être effectués en tenant compte uniquement des performances du système lui-même.



La production vidéo et télévision actuelle n'a recours au numérique que pour des traitements intermédiaires, le produit final étant encore toujours diffusé sous forme analogique. Les choix effectués à ce niveau sont sans conséquences sur les autres équipements.

La distribution de produits éditoriaux en vidéo numérique

Qu'il s'agisse de produits utilisant comme support le disque compact ou d'une distribution en réseau (qui suppose un réseau capillaire à débit suffisant jusqu'à l'utilisateur final), les faibles débits actuels exigent une forte compression, forcément dégradante. C'est le domaine de MPEG 1. En support CD, c'est le format CD-I qui semble devoir s'imposer, à travers les extensions *Full Motion Video*, ainsi que le Vidéo-CD pour les titres qui ne demandent pas une forte interactivité, mais cette tendance reste encore fragile, compte-tenu de l'étroitesse du marché.

Les traitements nécessaires à la compression MPEG doivent encore, dans la plupart des cas, être confiée à des sociétés spécialisées, mais des stations de compression sur micro-ordinateur approchant le traitement en temps réel sont à présent disponibles (de 100 000 à 300 000 F). De nombreuses annonces font état de la forte baisse de tarif des cartes spécialisées, mais les micro-ordinateurs nécessaires, même s'il s'agit de matériels largement diffusés (Mac ou PC) restent des configurations coûteuses : Power Mac ou Pentium, disques de forte capacité (1 à 2 Go), mémoire importante (16 Mo), etc.

Il faut également citer le standard concurrent du CD-I, développé par 3DO et adopté notamment par Panasonic, qui exploite la technologie double vitesse des CD-ROM, et permet d'envisager une qualité accrue. Longtemps annoncé, ce système est à présent disponible, mais connaît une diffusion limitée aux Etats Unis et inexistante en Europe. Son marché cible semble actuellement se limiter aux jeux.

Kaleida, la filiale commune d'Apple et IBM chargée des formats de base du multimédia, développe pour sa part le système ScriptX, en fait un super langage de description comparable à Postscript, capable de créer des programmes fonctionnant à l'identique sur de multiples plates-formes (*Windows*, Mac, *OS/2* et CD-I à venir).

Pour les productions multimédias destinées au grand public et devant utiliser la vidéo plein écran, la seule solution ayant une existence commerciale à des coûts acceptables est, pour l'instant, le procédé MPEG 1, notamment exploité par le CD-I et le Vidéo-CD.

Le choix d'un lecteur de CD-I doté de sa cartouche FMV comme station dans un établissement scolaire peut se justifier dans la mesure où des titres pertinents en nombre suffisant ont pu être trouvés. Pour l'heure, les programmes éducatifs disponibles sont encore peu nombreux.

D'une manière générale, le choix d'un équipement de restitution de vidéo numérique ne peut être fait qu'en tenant compte des titres que l'on envisage d'utiliser.

L'intégration de vidéo à des usages informatiques

Il y a un vaste débouché éditorial pour des CD-ROM documentaires, des encyclopédies, des programmes éducatifs, des jeux, etc. exploitant la vidéo numérique. Dans ce domaine, les grands acteurs semblent avoir fait leur choix, ce qui permet de mieux deviner ce que sera l'évolution future. La société Intel, notamment, est passée, en moins de dix-huit mois, de DVI



à *Indeo 3* utilisant la carte *Smart Video Recorder*, des produits contradictoires dans leur définition : si DVI était fondé principalement sur une restitution de qualité à travers un équipement matériel, *Indeo 3* prévoit une restitution entièrement logicielle, la carte *Smart Video Recorder* n'étant nécessaire que pour la capture de séquences.

L'ensemble des algorithmes *Indeo* a été très rapidement retenu par Microsoft (*Vidéo pour Windows 1.5* inclura *Indeo 3*), ainsi que par Apple pour *Quicktime*, ce qui lui donne aujourd'hui une position de standard de fait. De nombreux fabricants et éditeurs ont également conclu des accords de partenariat avec Intel sur *Indeo*. Il est vrai que certains l'avaient déjà fait pour DVI, et qu'ils disposent à présent de magnifiques produits... invendables, faute pour l'acheteur potentiel de pouvoir acquérir le matériel correspondant ! Actuellement, on peut cependant penser que la lignée *Indeo* devrait faire l'objet de larges développements : les nouveaux processeurs (Intel Pentium, Power PC) sont capables d'effectuer des traitements plus rapides, et les architectures des machines évoluent dans le sens d'une meilleure prise en compte des difficultés (bus PCI). Déjà, la société Teletota propose un service de numérisation *Indeo* en 320 x 240 à 24 images par seconde, c'est-à-dire une définition équivalente à MPEG 1⁷.

D'autres systèmes de cette nature existent néanmoins ; on peut citer *CinePack* (SuperMac) et *Captain Crunch* (Media Vision). Le Motion JPEG, qui donne aujourd'hui les meilleurs résultats, est généralement réservé aux bornes interactives, car il suppose des équipements matériels coûteux et non normalisés pour la restitution.

Il faut également citer les matériels de décompression MPEG sur micro-ordinateur, comme la carte *Real Magic*, disponible pour PC. Quoique d'un prix modéré (2 500 à 4 000F), ces matériels ont du mal à s'imposer car il n'existe pas une bibliothèque de programmes suffisante. Il faut cependant noter qu'un accord a été conclu avec Philips pour l'exploitation du Vidéo CD.

Les éléments de vidéo intégrés à une application informatique sont généralement prévus pour être utilisés sous Quicktime (Macintosh et quelquefois PC) ou Vidéo pour Windows. Les décodeurs utilisés, s'ils ne sont pas disponibles en standard dans les systèmes, sont toujours fournis avec le produit (module d'exécution de Vidéo pour Windows ou de Quicktime pour PC, extension à de nouveaux algorithmes comme Captain Crunch ou Cinepack).

Les systèmes demandant des matériels spécifiques pour la décompression sont souvent réservés aux bornes interactives, qui sont des ordinateurs dédiés à un nombre limité d'applications.

Compression et station multimédia

Pour l'instant, et sans doute encore pour longtemps, il n'existe aucune solution généraliste couvrant l'ensemble du secteur du multimédia.

Les choix doivent s'effectuer en tenant compte de nombreux paramètres, au premier plan desquels le choix entre une simple station de restitution et une station de production, même modeste.

⁷ Ce service devra cependant faire l'objet d'une évaluation et d'une analyse comparée avec MPEG au niveau des coûts et performances.



Dans le cas d'une station d'exécution, le choix ne peut être fait qu'en tenant compte des produits que l'on souhaite utiliser. La compression est alors un paramètre parmi d'autres, qui déterminera la compatibilité entre matériel et applications.

En revanche, pour une station de production, même modeste (production de documents légers pour un usage personnel), l'offre est très large, et la facture peut être très élevée. Il convient avant tout de bien connaître les contraintes du développement multimédia. Dès lors qu'on envisage de sortir du cadre de la PréAO, et surtout utiliser ses productions sur d'autres machines, les difficultés peuvent devenir considérables. Une bonne connaissance des logiciels de composition qui seront utilisés et des systèmes est nécessaire, en particulier en ce qui concerne la prise en compte des différents codages, et du matériel nécessaire pour effectuer les traitements correspondants.

De plus, si les systèmes, à travers leurs extensions multimédias (*Quicktime, Vidéo pour Windows*), intègrent progressivement les principaux codages, les logiciels doivent encore, dans bien des cas, assurer directement les fonctions de décompression⁸. Les possibilités d'utilisation sur des machines non dotées de l'application utilisée doit expressément être vérifiée.

Pour ce qui concerne les périphériques d'acquisition (notamment vidéo), il convient de distinguer ceux qui utilisent une compression logicielle en temps différé (*Video Blaster*) de ceux qui assurent un codage compressif en temps réel (*Smart Video Recorder*). Au delà de quelques dizaines de secondes, les premiers se révèlent inutilisables (plus de 40 Mo par minute sont nécessaires).

Pour une station d'exécution, la compatibilité avec les applications utilisées doit être le critère déterminant du choix.

En revanche, pour les stations de composition et développement, les choix sont beaucoup plus complexes, et il convient de vérifier avec soin les compatibilités des différents éléments, logiciels et matériels.

L'exécution sur un ordinateur différent de celui qui a servi à la composition peut se révéler impossible, et doit faire l'objet de la plus grande attention.

⁸ La décompression JPEG, notamment, est prise en charge par *Quicktime* sur Macintosh, mais doit être traitée par les applications sur les PC.

Annexes

Méthode de Huffman : exemple

compression de la chaîne : "MODESTE CITATION VAUT MIEUX QUE BRILLANTE PARAPHRASE" (d'après [Marseau, 1992] p. 389 et suivantes).

Les caractères utilisés font partie de {espace,A,B,C,D,E,H,I,L,M,N,O,P,Q,R,S,T,U,V,X}.

A chacun de ces caractères on associe sa fréquence d'apparition. On obtient :

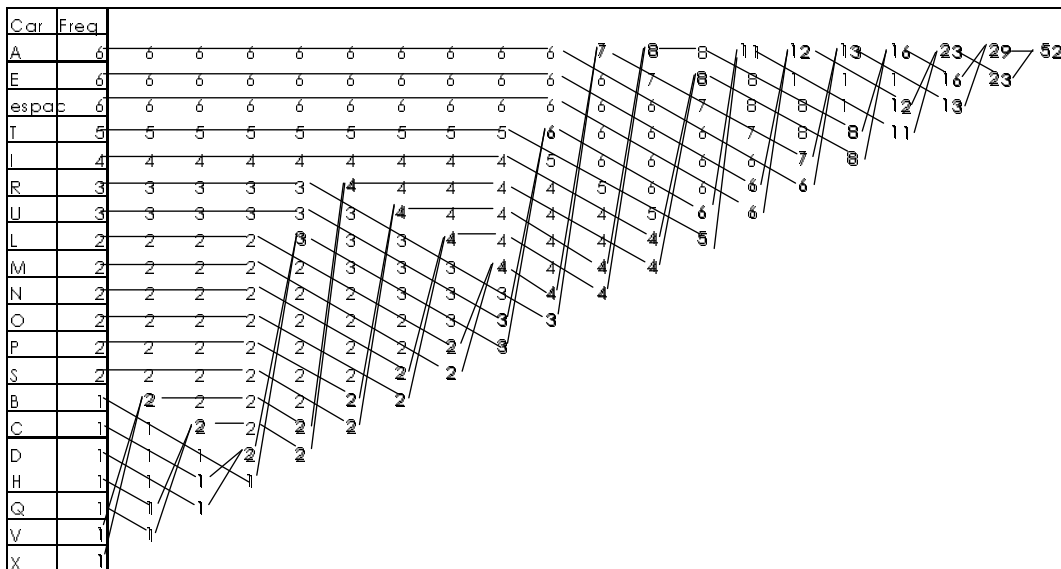
car.	esp	A	B	C	D	E	H	I	L	M	N	O	P	Q	R	S	T	U	V	X
freq.	6	6	1	1	1	6	1	4	2	2	2	2	2	1	3	2	5	3	1	1

Après classement par fréquences décroissantes, on procède au regroupement des deux symboles de plus faible probabilité, en reclassant les regroupements obtenus. La répétition de ce processus jusqu'à ne plus disposer que d'un seul groupe produit un arbre dit "arbre de Huffman".

Pour obtenir le codage final, il suffit de parcourir l'arbre à partir du signal à coder jusqu'au dernier regroupement. A chaque regroupement rencontré, on attribue le bit 0 si on se trouve sur la branche supérieure, et le bit 1 sinon. Les bits ainsi obtenus sont rangés du poids le plus faible au poids le plus fort.

Pour pouvoir décoder un message, il faut disposer de la table de codage qui doit généralement être transmise avec les données codées. En outre, lors de la relecture, il faut être capable de décomposer la suite de bits en caractères, ce qui ne va pas de soi, le codage de Huffman utilisant des "mots" de longueur variable et aucun séparateur.

On démontre aisément que la structure de l'arbre de Huffman produit des codes séparables, ce qui signifie qu'il n'y a qu'une seule manière de relire la chaîne codée. En effet, chaque suite de bits obtenue décrit un chemin de l'arbre. Toute séquence commençant par la même suite de bits reviendrait à décrire le même chemin ; il n'est donc pas possible d'obtenir une autre séquence débutant par la même suite.



Exemple d'arbre de Huffman



A	001	I	0010	M	01001	S	00001	H	001110
E	100	R	0101	N	00010	B	11011	Q	001111
esp	101	U	1100	O	00011	C	110100	V	001100
T	111	L	01000	P	00000	D	110101	X	001101

Table de codage obtenue

La chaîne est donc codée de la manière suivante :

```

01001 00011 110101 100 00001 111 100 101 110100 0010 111 011 111 0010 00011
M   O   D   E   S   T   E   ‘   C   I   T   A   T   I   O
00010 101 001100 011 1100 111 101 01001 0010 100 1100 001101 101 001111 1100
N   ‘   V   A   U   T   ‘   M   I   E   U   X   ‘   Q   U
100 101 11011 0101 0010 01000 01000 011 00010 111 100 101 00000 011 0101 011
E   ‘   B   R   I   L   L   A   N   T   E   ‘   P   A   R   A
00000 001110 0101 011 00001 100
P   H   R   A   S   E

```

Le codage produit a une longueur de 198 bits.

Codée en ASCII, la même chaîne occuperait 52 octets, soit 416 bits ; en utilisant un codage optimisé pour 20 caractères, chaque caractère doit être codé sur 5 bits, ce qui donne une longueur totale de 260 bits. Il faut cependant noter qu'il est également nécessaire, tant pour le codage à longueur fixe sur 5 bits que pour le codage Huffman, de transmettre la table de correspondance, qui occuperait 40 octets... Dans ce cas précis, la chaîne compactée serait donc plus longue que la chaîne originale, mais cette situation s'inverse rapidement avec la taille des fichiers (la table de codage a toujours une longueur inférieure à 768 octets).



Méthodes Lempel Ziv : algorithmes et exemple

Les algorithmes et exemples donnés sont issus de [Plume, 1993]. Les algorithmes de compactage et décompactage utilisent un dictionnaire de taille limitée (1024 entrées). Les codes sont donc toujours de même taille (sur 10 bits). Certaines variantes utilisent des dictionnaires de taille variable, les codes étant eux-mêmes de longueur croissante au long du fichier. Un code réservé indique alors que le nombre de bits de codage est incrémenté.

```
/* Algorithme de compactage LZ*/
début
  /* initialisation */
  Dictionnaire = table ASCII
  Code_latent = " " /* vide */
  Maxdim = 1024 /* taille maximale du dictionnaire (optionnel) */
  Taille_dico = 257
  /* Traitement */
  lire l'octet 1 du fichier
  c=octet1
  Tant que c<>EOF /* fin de fichier = faux */
    si Code_latent = " " /* vide */
      alors
        Code_latent = c
        Ancien_rang = ASCII(c) /* sauve le rang de c */
      sinon
        Chaîne = latent + c /* concaténation */
        si chaîne appartient au dictionnaire
          alors
            Code_latent = Chaîne
            Ancien_rang = rang de chaîne dans dictionnaire
          sinon
            Emettre (Ancien_rang)
            Ajouter chaîne au dictionnaire
            Taille_dico = Taille_dico + 1
            si Taille_dico = Maxdim
              alors /* réinitialisation du dictionnaire (optionnel) */
                Emettre code 257
                Taille_dico = 257
            fin si
            Code_latent = c
            Ancien_rang = ASCII(c)
          fin si
    fin si
  Lire(c)
Fin Tant que
Emettre Ancien_rang /* indice dernière chaîne */
Emettre 256 /* code fin de traitement */
Fin traitement
```

Processus de compactage



Code lu	Ajout au dictionnaire	Code_latent	Code émis
D	rien	D	rien
U	DU (258)	U	D
' '	U' ' (259)	' '	U
C	' 'C (260)	C	' '
O	CO (261)	O	C
D	OD (262)	D	O
A	DA (263)	A	D
G	AG (264)	G	A
E	GE (265)	E	G
' '	E' ' (266)	' '	E
A	' 'A (267)	A	' '
U	AU (268)	U	A
' '	rien	U' '	rien
D	UD (269)	D	259
E	DE (270)	E	D
C	EC (271)	C	E
O	rien	CO	rien
D	COD (272)	D	261
A	rien	DA	rien
G	DAG (273)	G	263
E	rien	AG	rien
' '	GE' ' (274)°	' '	265
D	' 'D (275)	D	' '
'	D' (276)	'	D
I	'I (277)	I	'
M	IM (278)	M	I
A	MA (279)	A	M
G	rien	AG	rien
E	AGE (280)	E	264
FIN	rien	rien	E

Exemple : codage de la chaîne
“ DU CODAGE AU DECODAGE DES IMAGES ”

Jusqu'à la première paire présente dans le dictionnaire, le codage est traditionnel. Pour toute paire répétée, on ne transmet que son adresse; l'absence de codage d'un caractère peut être localisée par le code transmis immédiatement après qui est supérieur à 257. Un code spécifique (non exploité ici) peut indiquer que les adresses seront codées avec un bit de plus à partir de ce rang lorsque la taille du dictionnaire n'est pas limitée. Ainsi, la taille des adresses peut ne pas être prédéfinie, et il en est de même, par voie de conséquence, de la taille du dictionnaire. Dans l'exemple cité, la taille du dictionnaire est fixée à 1024 entrées, ce qui permet un codage fixe des adresses sur 10 bits. Ces différentes possibilités de paramétrage permettent d'adapter l'algorithme à différentes situations, et notamment de l'optimiser en fonction de la taille des fichiers. Dans tous les cas, le dictionnaire est reconstruit par le décompacteur durant son action, et n'a donc pas à être conservé.

Des améliorations et variantes peuvent être apportées, notamment en conservant un certain nombre d'adresses du dictionnaire dans le fichier compacté pour des fonctions particulières, comme l'augmentation de la taille des adresses, les changements éventuels de technique de compression, la purge de séquences rarement utilisées du dictionnaire, la réinitialisation du dictionnaire, ou la création d'un deuxième dictionnaire



```
/*Algorithme de décompactage LZ*/
Début
  /* initialisation */
  Dictionnaire = table ASCII
  Code_latent = ''
  Code_fin_compression =256
  /* traitement */
  Lire Code
  Tant que Code <> 256 /* fin de compression */
  si Code_latent = ''
    alors
      Code_latent = Code
      Emettre Code
    sinon
      si Code < 255
        Alors
          Emettre Code
          Chaîne = Code_latent + Code
          Ajouter Chaîne au dictionnaire
          Code_latent = Code
        sinon
          Chaîne = Séquence (code)
          /* chaîne correspondant à ce code dans le dictionnaire */
          Emettre Chaîne
          Chaîne_tempo = Code_latent + premier caractère de Chaîne
          Ajouter Chaîne_tempo au dictionnaire
          Code_latent = Chaîne
      fin si
    fin si
  fin Tant que
fin traitement
```

Processus de décompactage



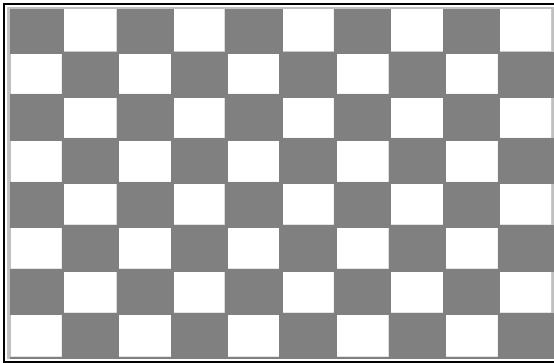
Code reçu	Code_latent	Texte final	Ajout au dictionnaire
D	rien	D	rien
U	D	U	DU (258)
''	U	''	U'' (259)
C	''	C	''C (260)
O	C	O	CO (261)
D	O	D	OD (262)
A	D	A	DA (263)
G	A	G	AG (264)
E	G	E	GE (265)
''	E	''	E'' (266)
A	''	A	''A (267)
259	A	U''	AU'' (268)
D	U''	D	U''D (269)
E	D	E	DE (270)
261	E	CO	EC (271)
263	CO	DA	COD (272)
265	DA	GE	DAG (273)
''	GE	''	GE'' (274)
D	''	D	''D (275)
'	D	'	D' (276)
I	'	I	I' (277)
M	I	M	IM (278)
264	M	AG	MA (279)
E	AG	E	GAE (280)

Exemple : décodage de la chaîne codée dans l'exemple précédent

Compression *Run Length* : exemple

Compression d'une image composée d'un damier noir et blanc.

Dans le cas d'un damier noir et blanc de 10 carrés sur 10, représenté sur un écran VGA monochrome 640 x 480, correspondant à des carrés de 64 x 48, chaque ligne serait codée en regroupements de 64 bits, soit 8 octets.



Les pixels blancs étant traduits par un bit à 0 et les noirs à 1, une séquence de 8 pixels noirs sera codée "11111111", soit par l'octet FFh en hexadécimal, et une séquence de 8 pixels blancs sera codée "00000000", soit par l'octet 00h en hexadécimal.

Pour différencier le codage de plages de pixels répétés des pixels isolés, on utilise un code discriminant réservé à cet usage, généralement le caractère hexadécimal 14h (cf. format PCX).

Dans le cas d'un octet non répété dont le code est celui de l'indicateur de répétition (14h), le codage sera obtenu par la représentation du code 14h répété une fois, soit "14 - 14 - 1". Le "compactage" conduit ici à un triplement de la taille (3 octets au lieu d'un), mais cette dégradation est heureusement rare.

On obtient alors "14 - FF - 08" pour 64 pixels noirs (14h indique la répétition ; FFh est l'octet à répéter ; 08 est le nombre de répétitions), et "14 - 00 - 08" pour 64 pixels blancs (14h indique la répétition ; 00h est l'octet à répéter ; 08 est le nombre de répétitions), soit finalement "14-FF-08-14-00-08-14-FF-08..." pour une ligne commençant par un carré noir, et "14-00-08-14-FF-08-14-00-08-..." pour une ligne commençant par un carré blanc.

Le codage obtenu est de 30 octets par ligne, soit 14 400 octets au lieu de 38 400 octets. Evidemment, s'il était possible d'appliquer encore la même méthode sur des groupes de 8 octets, ou même des lignes, les résultats seraient tout autres, mais le *Run Length* ne va pas jusque-là. Il est facile de comprendre que le procédé LZW donnera dans pratiquement tous les cas de meilleurs résultats, mais au prix de traitements largement plus complexes.

Des variantes de ce procédé analysent les répétitions de bits (images monochromes, notamment issues de scanners) ou de pixels (séquences de 3 octets pour les images sur 24 bits). Dans le cas d'images organisées en plans de bits (4 plans pour les images EGA/VGA 16 couleurs), la compression sera plus importante en travaillant plan par plan, les composantes stables se révélant plus nombreuses.

Bibliographie

Delassale, B., “ *La Déferlante Numérique* ”,
Sonovision, N°355, janvier 1992.

Huffman, D. A., “ *A method for the construction of minimum redundancy codes* ”,
Proceedings of the IRE, Vol. 40, Numéro 9, Septembre 1952.

Marseau X., “ *Compression et cryptage en informatique* ”,
éditions Hermès, Collection Traité des nouvelles technologies, série informatique, 1992.

Nelson M., “ *La compression des données* ”,
éditions Dunod, 1993.

Paquel N., “ *Multimédia et télévision interactive* ”,
Mémoires optiques et systèmes, N° 117, septembre 93.

Plume P., “ *Compression des données* ”,
éditions Eyrolles, 1993.

Terrasson J., “ *Les outils du multimédia* ”,
éditions Armand Colin, 1992.

Welch, T., “ *A technique for high performance data compression* ”, IEEE Computer, Vol. 17,
juin 1984.

Ziv, J. et Lempel, A., “ *A universal algorithm for sequential data compression* ”, IEEE
Transactions on Information Theory, Vol. 23, mai 1977.

Ziv, J. et Lempel, A., “ *Compression of individual sequences via variable rate coding* ”,
IEEE Transactions on Information Theory, Vol. 24, septembre 1978.

Documents CNDP

Note technique “ *Numérisation des images* ”,
Cavet D., CNDP, , 1994 (à paraître).

Note technique “ *Le système Photo CD de Kodak* ”,
Cavet D., CNDP, 1992.

Note technique “ *CD-I Digital Video et VHS : essais comparatifs* ”,
Banckaert M., Gasser P., Lavacry M., CNDP, 1994.

Dossiers de l'ingénierie éducative, “ *De l'analogique au numérique* ”,
CNDP, Les cartes vocales, N° 13, juin 1993, p. 16-18.

Normes

MPEG 1 : ISO/CEI 11 172 (parties 1, 2 et 3).

MPEG 2 : ISO/CEI 13 818 (parties 1, 2 et 3).

JPEG : ISO/CEI 10 918.

JBIG : ISO/CEI 11 544.

Ces normes sont disponibles en français auprès de
AFNOR, Tour Europe, Cedex 7, 92049 Paris La Défense