

Les standards H.264/AVC et HEVC

Marco Cagnazzo

Département Traitement du Signal et des Images
TELECOM ParisTech

24 mai 2013

Plan

- 1 Introduction
- 2 H.264/AVC
 - Définitions et schéma
 - Nouveaux modes et outils
 - Profils, niveaux, performances
 - Network abstraction layer
- 3 HEVC
 - Définitions
 - Coding tools
 - Stream

Plan

1 Introduction

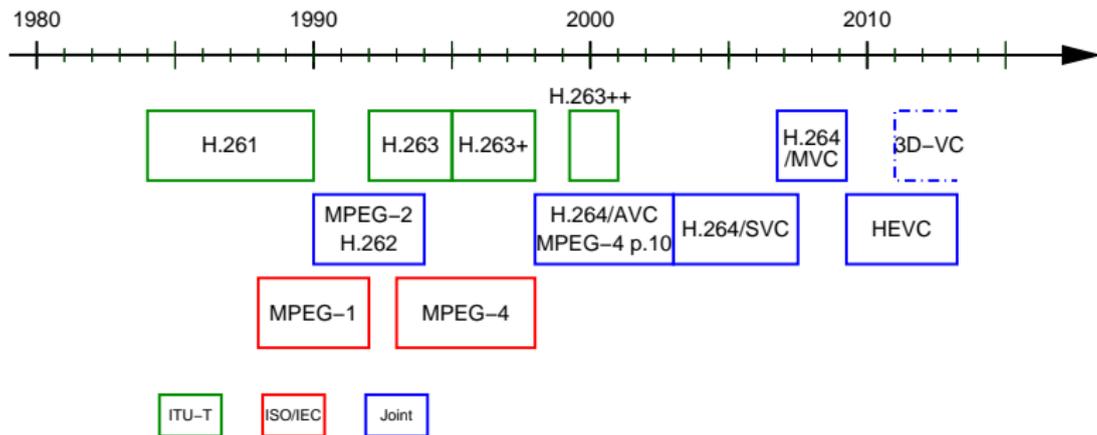
2 H.264/AVC

- Définitions et schéma
- Nouveaux modes et outils
- Profils, niveaux, performances
- Network abstraction layer

3 HEVC

- Définitions
- Coding tools
- Stream

Les standards vidéo : chronologie



Les standards vidéo

Groupes de standardisation

Organismes de standardisation :

ISO International Standardization Organization

IEC International Electrotechnical Commission

ITU International Telecommunication Union

Groupes de travaille

- MPEG (1988) : ISO/IEC Moving Picture Expert Group
- VCEG (1997) : ITU Video Coding Expert Group
- *Joint Video Team*: H.264 et MPEG-4/Part 10 (JVT);
extension scalable de H.264 (SVC)

Le standard H.264/AVC

- Développé en 1998-2003
- Standard approuvé comme ITU-T Recommendation H.264 et comme ISO/IEC International Standard 14496-10 (MPEG-4 part 10) Advanced Video Coding (AVC).
- **Codeur hybride**
- Seulement la partie vidéo

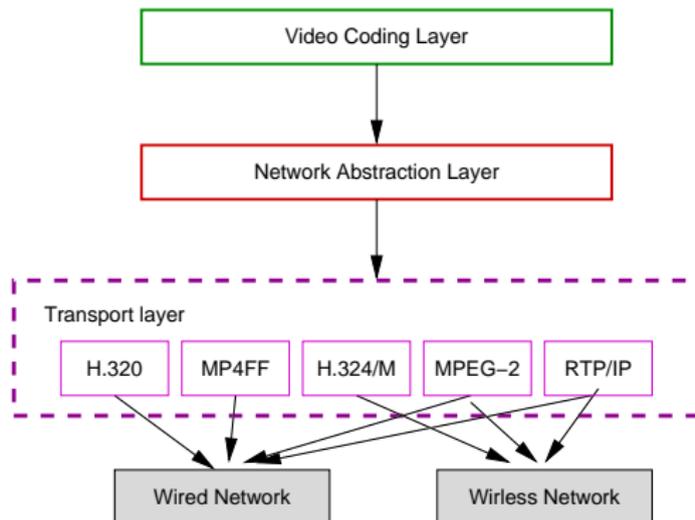
Objectives

- But : optimiser les performance débit-distorsion
- Jusqu'à -60% de débit pour la même qualité par rapport à MPEG-2
- Applications :
 - Broadcast sur câble, satellite, xDSL, TNT, etc.
 - Stockage sur support de mémoire (optique, magnétique, *solid state*, etc.)
 - *Conversational services* sur Ethernet, LAN, WiFi, xDSL, etc.
 - Video-on-Demand et streaming multimédia sur Ethernet, LAN, WiFi, xDSL, etc.
- On doit penser à l'intégration avec le niveau de transport/stockage

Structure de H.264

- L'efficacité de la transmission dans des environnements différents dépend :
 - de l'efficacité de la compression ;
 - et de l'intégration avec les protocoles de transport.
- Seulement l'intégration entre adaptation au réseau et compression assure les meilleures performances
- C'est pour cela que H.264 est constitué de deux couches (*layers*) conceptuelles
 - Video Coding Layer (VCL)
 - Network Adaption Layer (NAL)

Structure de H.264



VCL et NAL

- VCL fournit des outils performants pour la compression
 - Intra-prediction, variable-size ME/MC, in-loop filtering, CABAC, etc.
 - Profiles et niveaux
- NAL permet l'adaptation au différents types de transport
 - *Packet switched* transport (RTP/IP, TCP/IP, ...) vs. *circuit switched* transport (MPEG-2, H.320, ...)
 - Streaming vs. storage

Plan

1 Introduction

2 H.264/AVC

- Définitions et schéma
- Nouveaux modes et outils
- Profils, niveaux, performances
- Network abstraction layer

3 HEVC

- Définitions
- Coding tools
- Stream

Représentation de la vidéo

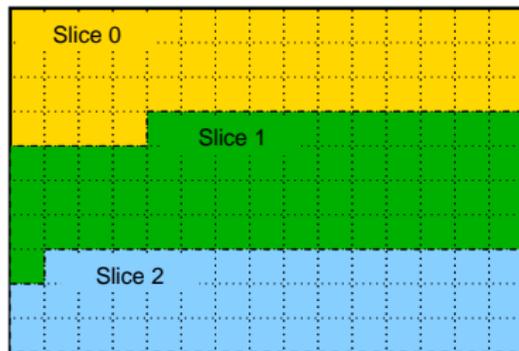
- Entrée : vidéo numérique en format YCbCr (équivalent à YUV)
- *Luma* et *Chroma*
- Échantillonnage : 4 : 2 : 0
 - Cela correspond à la sensibilité différente du SVH à Luma et Chroma

Le macroblock

- **Macroblock** : un bloc de 16×16 échantillons de Luma et deux blocs 8×8 de Chroma
- Le MB est *l'unité de codage* :
 - Chaque MB est codé avec un *mode* (Intra, Inter, Skip)
 - Chaque mode produit : une prédiction, un résidu, des paramètres
 - Le 2 derniers sont envoyés au décodeur

Les slices

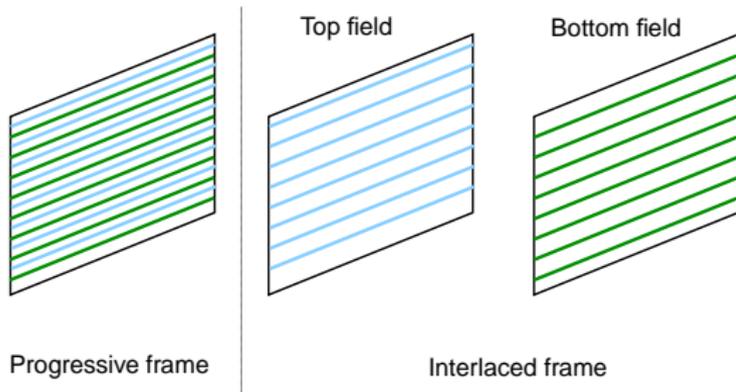
- Dans H.264/AVC les macroblocks sont groupés en *slices*
- Une slice est une ensemble de macroblocks en *raster scan order*
- Des autres structures de slice sont possibles



Les slices

- Normalement une slice correspond à une image
- Cinq types de slices existent :
 - Intra (I) : tous les MB en mode Intra
 - Predictive (P) : I + modes prédictives avec une image de référence
 - Bidirectional (B) : P + modes prédictives avec deux image de référence
 - Switch-Intra (SI) et Switch-Predictive (SP) : utilisées pour améliorer l'efficacité du passage d'un train binaire à l'autre

Codage par *frame* ou *field*



- Le field est lié aux images entrelacées (lignes paires et impaires)
- Le choix entre codage par frame ou par field est fait :
 - À niveau d'image (P-AFF)
 - À niveau de macroblock (MB-AFF)

Pictures et Sequences dans H.264

- Une vidéo codée en H.264 consiste dans une suite (*sequence*) d'images, appelées *coded pictures*
- Une *coded picture* peut être une frame ou un field
- L'image est divisée en MB
- Les MB sont groupés en slices
- Donc un'image est faite de une ou plusieurs slices

Rappel : schéma du codeur hybride

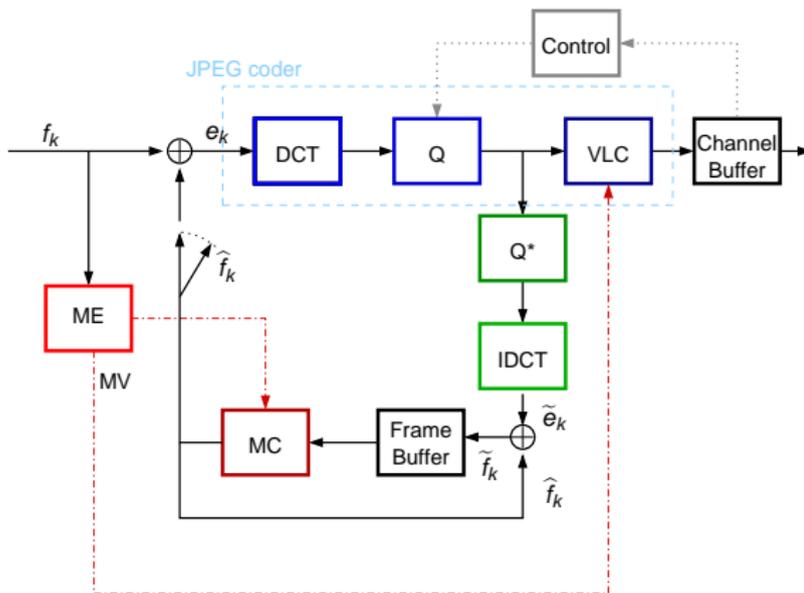
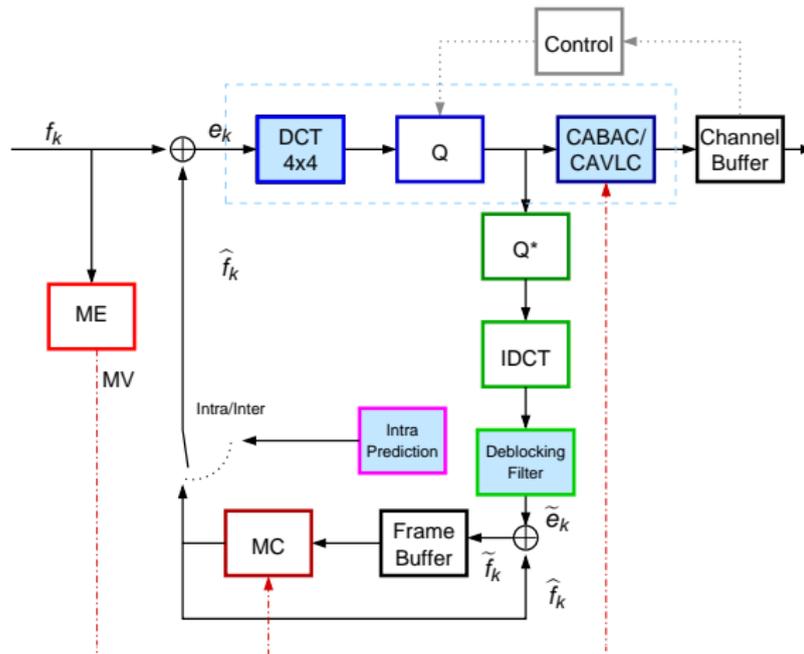


Schéma du codeur H.264

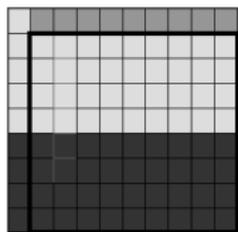


Nouveaux modes de codage

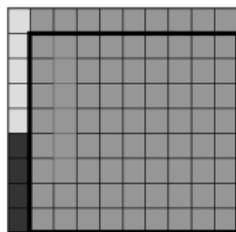
- Modes *Intra*
- Modes *Inter*
- Modes *Skip* et *Direct*

Modes Intra avec prédiction spatiale

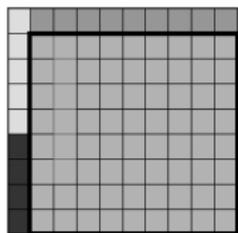
Prédiction sur des blocs 16x16 (régions uniformes), 4 types de prédiction:



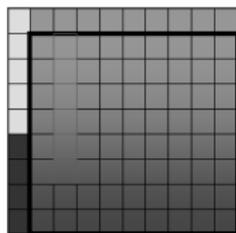
horizontale



verticale



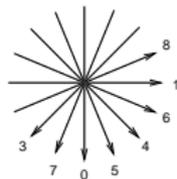
moyenne



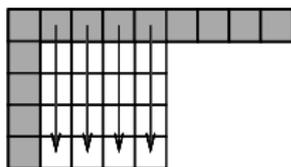
plane prediction

Modes Intra avec prédiction spatiale

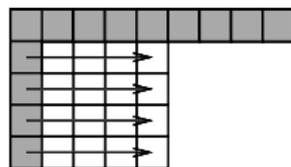
Prédiction sur des blocs 4x4 (détails), 9 types de prédiction: 8 directions et moyenne



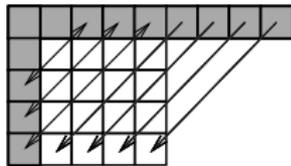
Directions



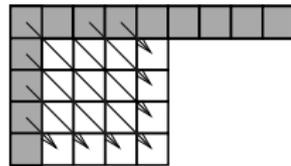
Mode 0



Mode 1



Mode 3



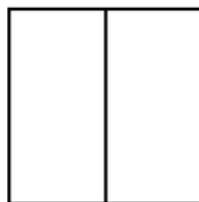
Mode 4

Modes Inter avec taille variable des blocs

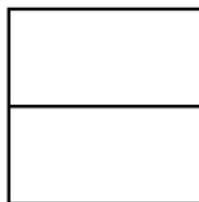
Chaque bloc 16x16 peut être décomposé pour vérifier si il est nécessaire une représentation plus fine du mouvement



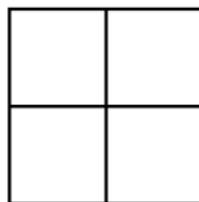
16x16



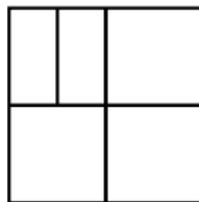
16x8



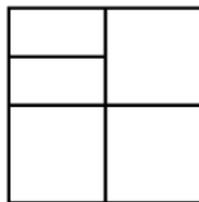
8x16



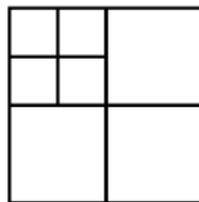
8x8



8x4



4x8



4x4

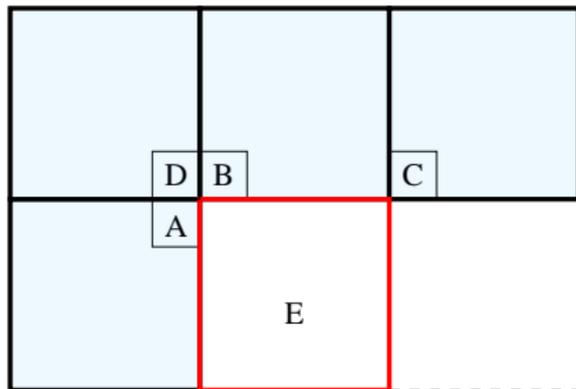
Estimation du mouvement

Exemple de partition avec bloc à taille variable



Représentation du mouvement

- Vecteur de mouvement à précision du quart de pixel
- Deux interpolation successive : avec un filtre de longueur 6, et bi-linéaire
- Codage des vecteurs par prédiction (médian des voisins)
- Les vecteurs peuvent sortir de l'image



Compensation du mouvement

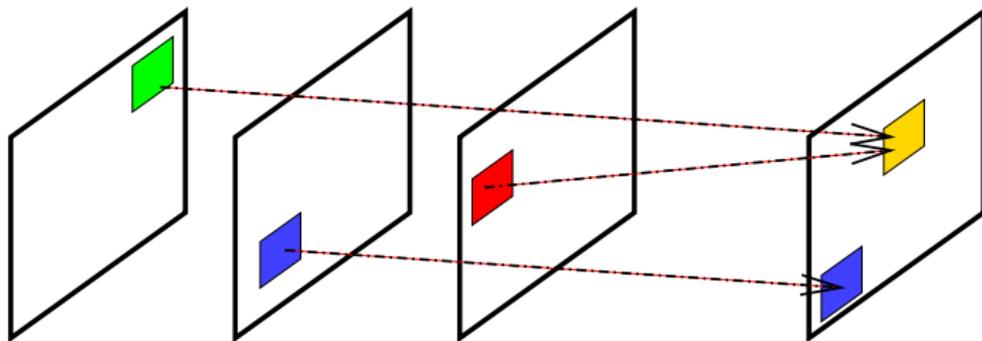
- Le choix de l'image de référence est très peu contraint
- Dans le cas P et dans le cas B il suffit donner une (ou deux) liste de image pour la prédiction
- Prédiction B pondérée avec coefficients arbitraires

Slices de type B

- Generalized B-slices
- Deux listes d'images de référence
- Multiple reference frame
- Flexibilité dans le choix de la référence
- Flexibilité dans l'ordre de codage
- Prédiction par moyenne pondérée

Compensation du mouvement

Références multiples



Références

Image à coder

Modes Skip et Direct

- Skipped MB : c'est comme un MB prédictif pour lequel on utilise
 - zéro bits pour coder le vecteur de mouvement
 - zéro bits pour coder le résidu
- On utilise le vecteur médian des MBs voisin pour trouver le MV
- On copie les pixels pointés par ce MV de l'image de référence
- Coût de codage : seulement la signalisation du mode
- Direct : mode Skip pour les B slices

Nouveaux outils de codage

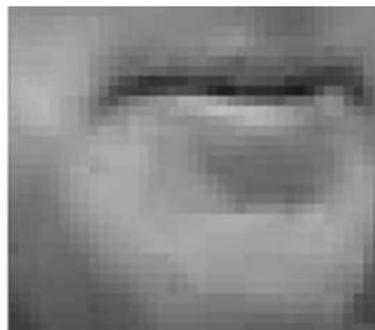
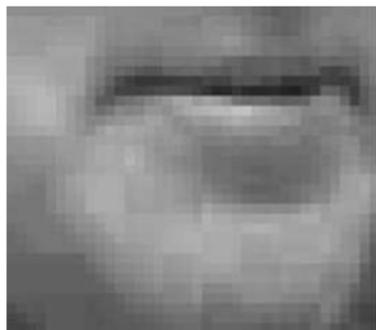
- Filtre de de-blocking
- Nouvelle transformée
- Codeurs sans perte : CAVLC et CABAC
- Outils pour le changement de flux (*switch*)
- Outils pour la robustesse

Filtre de de-blocking

- Problème : effet de bloc à bas débit
- Cause : codage indépendant des bloc
- Solution : post-filtering
 - N'affecte pas le codage (frame buffer)
 - Non normatif, maximum de liberté
 - Besoin de mémoire supplémentaire
- Solution : in-loop filtering
 - Les images filtrée sont utilisée comme référence
 - Le filtre doit être normalisé
 - Meilleures performances (objectives et subjectives)
 - Complexité élevée (même si il y pas besoin de multiplications ni de divisions) : branching, petits blocs

Filtre de de-blocking

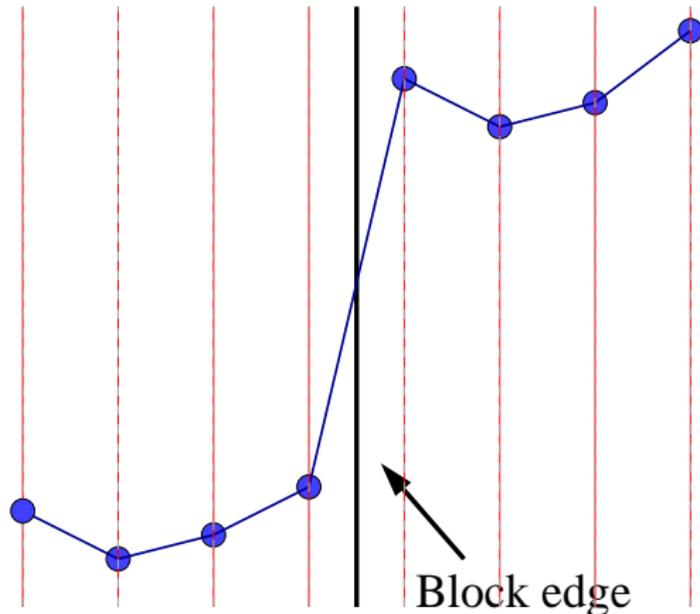
Exemple de filtre de de-blocking



Peter List, Anthony Joch, Jani Lainema, Gisle Bjøntegaard, and Marta Karczewicz: "Adaptive Deblocking Filter", in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, July 2003.

Filtre de de-blocking

Analyse de la discontinuité



Filtre de de-blocking

- Réduit les discontinuités entre blocs 4×4 adjacents
- Le filtre est adapté à :
 - caractéristiques de la vidéo ;
 - le mode de codage du bloc et de ses voisins (filtre plus fort si blocs Intra ou si compensation par images différentes)
 - l'amplitude de la discontinuité et le pas de quantification
- 4 types de filtrage sont possibles (plus le mode “non filtering”)
- Filtres non-linéaires (passe-bas plus seuillage)

Nouvelle transformée

- Transformée sur des blocs 4x4
 - La meilleure prédiction réduit la corrélation spatiale
- Approximation de la TCD, à coefficients entiers
 - Parfaite synchronisation codeur/décodeur
 - Coefficients : $0, \pm 1, \pm 2$ → implémentation par bit-shift et somme/ soustraction
- Deuxième niveau de transformée
 - Transformée 4x4 sur les 16 coefficients DC d'un bloc en Intra16
 - Transformée 2x2 sur les 4 coefficients DC des blocs de Chroma
- Une transformée 8x8 est possible (profil High) pour le blocs avec beaucoup de corrélation

Codage sans perte

- Deux techniques sont supportées
 - Une technique à faible complexité, basée sur des codes à longueur variable choisis en fonction du contexte (CAVLC)
 - Une technique à haute complexité, basée sur le codage arithmétique adaptative et basé contexte (CABAC)
- Les deux techniques améliorent sensiblement le codage sans perte des standards précédents
- On prend en compte que les statistiques peuvent varier et dépendent du contexte

Codage a longueur variable adapté au contexte (CAVLC)

- C'est la technique de codage sans perte du profile *baseline*
- Un type de dictionnaire pour les résidus
- Un second type pour les autres éléments de la syntaxe (vecteurs, modes, signalisation)
- L'adaptivité du dictionnaire permet de gagner 2–7% de débit

Codage arithmétique binaire adapté au contexte (CABAC)

- Chaque élément de la syntaxe est transformé en binaire
- Pour chacun on utilise un codeur arithmétique
 - Adaptatif : apprend les statistiques du signal
 - Contextuel : reconnaît que le même signal peut avoir différents comportements
- Complexité
- Gain en débit de 5–15% par rapport à CAVLC

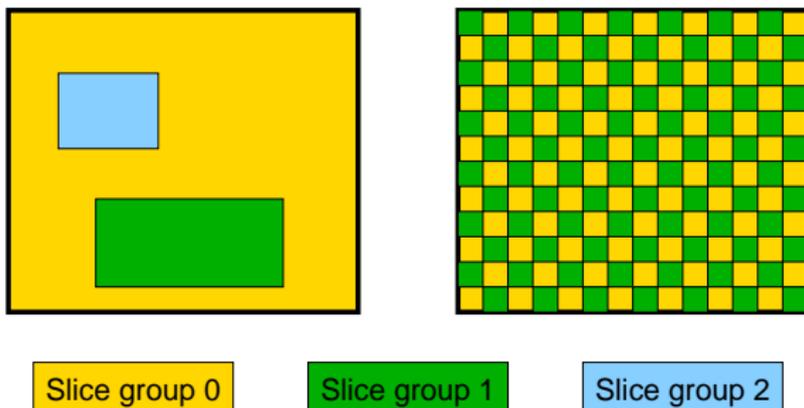
Robustesse aux erreurs : concepts généraux

- Mitiger la propagation des erreurs
 - Protection inégal (entêtes mieux protégés)
 - Limitation aux modes prédictifs
- Rendre facile la récupération de synchronisation
 - Utilisation de marqueurs de synchronisation
 - *Marker emulation prevention*
- Limiter l'impact visuel des erreurs
 - Reconnaître les images erronées
 - Afficher un image précédente plutôt que un'image erronée (*concealing*)

Robustesse dans H.264

- Flexible Macroblock Ordering
- Redundant Picture
- Data Partitioning
- Images de synchronisation et switching (SI et SP)

Flexible macroblock ordering



- Slices codées indépendamment : une perte affecte seulement une slice
- *Unequal protection* pour les régions d'intérêt
- Meilleur *concealing* dans le cas de perte

Redundant picture

- Dans une *redundant picture* il peut y avoir des *redundant slices*
- Une *redundant slice* est une nouvelle représentation d'une slice déjà codée, cette dernière étant appelée *primary*
- Si la *primary slice* est reçue, on n'utilise pas la *redundant*
- La *redundant* est utilisée si la *primary* est perdue
- Pour ne pas trop augmenter le débit de codage, la *redundant slice* a typiquement une qualité dégradée par rapport à la *primary*

Data partitioning

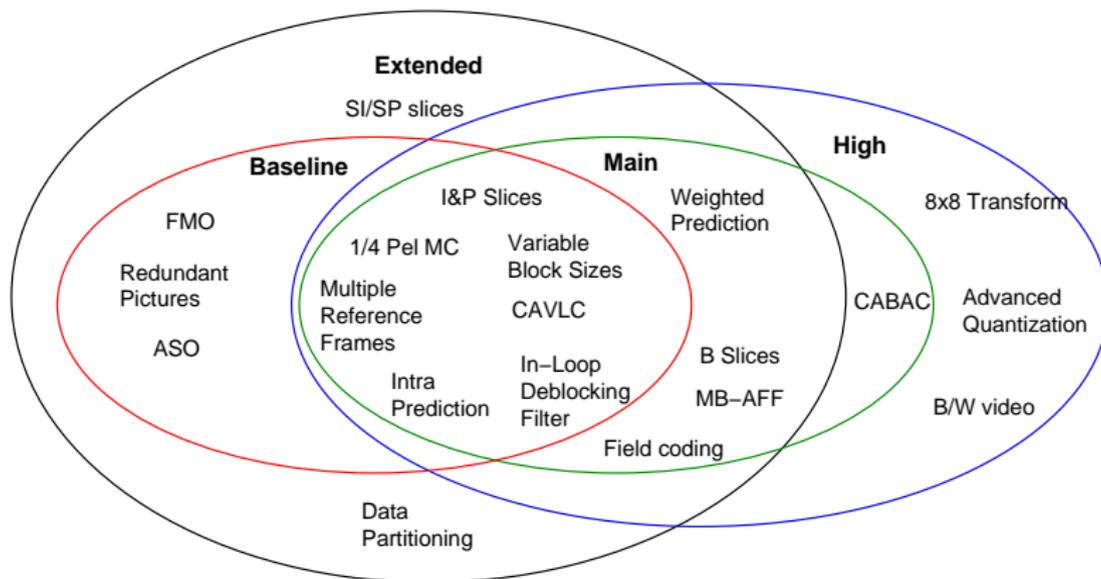
- Les données encodées sont réparties en 3 groupes
 - 1 Entêtes et vecteurs de mouvement
 - 2 Résidus pour les slices Intra
 - 3 Résidus pour les autres slices
- Cela permet d'utiliser UEP ou de donner des priorités aux données
- Standard précédents : 2 et 3 n'étaient pas séparés

Profils et niveaux

Il est possible de définir profils et niveaux avec une énorme variété

- Sept profils (Baseline, Extended, Main, High, High10, High4:2:2, High 4:4:4)
- 16 niveaux qui peuvent être combinés avec les profils.

Profils



Profils

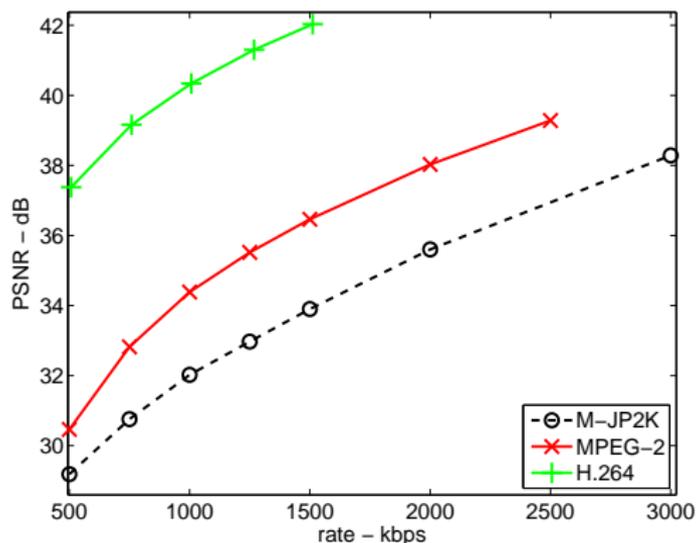
- Baseline Profile : applications à bas-coût (mobiles, visio-conférence)
- Main Profile : applications grand public de diffusion et de stockage ; a perdu de l'importance quand le profil High a été ajouté avec le même objectif.
- Extended Profile : Diffusion en flux (streaming) ; a des capacités de robustesse et de switching.
- High Profile : Le profil principal pour la diffusion et le stockage (TNT-HD, HD-DVD, BD)
- Les autres profils sont destinés à la production et aux applications professionnelles

Niveaux

- Les niveaux sont des limitations sur des paramètres
- Cela permet aux décodeurs de limiter les ressources employées pour le décodage
- Débit de 64 kbps à 240 Mbps pour les profils Baseline et Extended
- Débit de 80 kbps à 300 Mbps pour le profile High
- Résolutions : de QCIF, 15 fps à HD, 120 fps (ou 4K, 30 fps)

Performances débit-distorsion

Comparaison des performances de codage



H.264 – Exemples d'images décodées



MPEG-2



H.264

H.264 – Exemples d'images décodées



Motion JPEG2000



H.264

H.264 – Exemples d'images décodées



MPEG-2



H.264

H.264 – Exemples d'images décodées



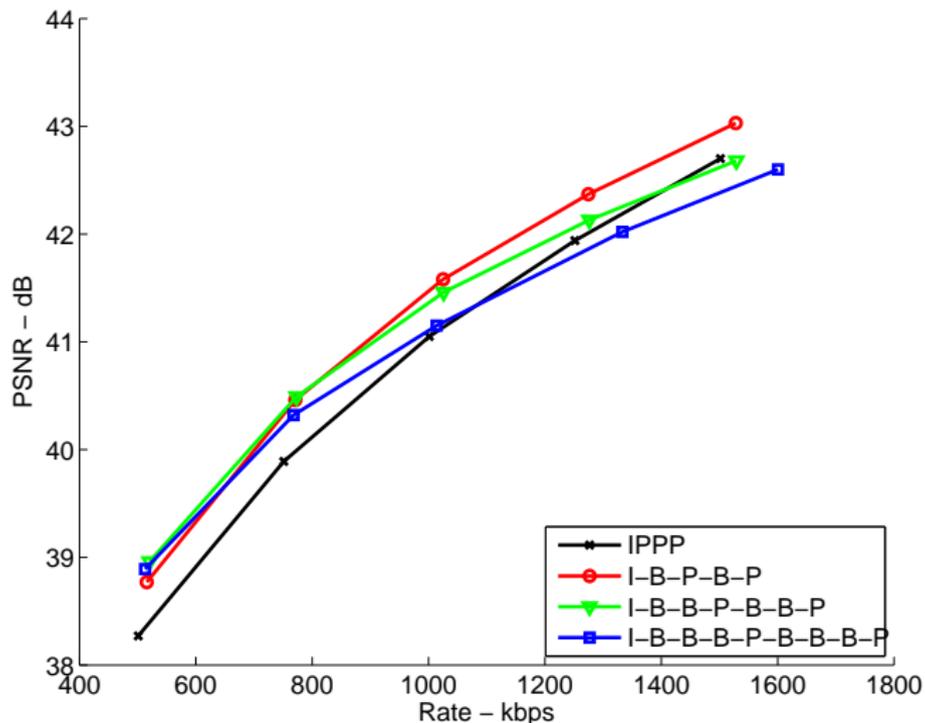
Motion JPEG2000



H.264

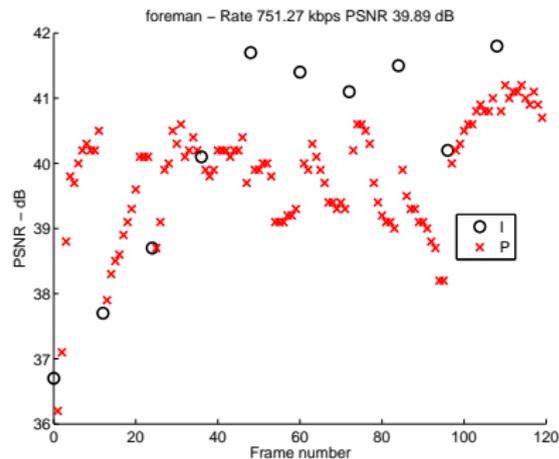
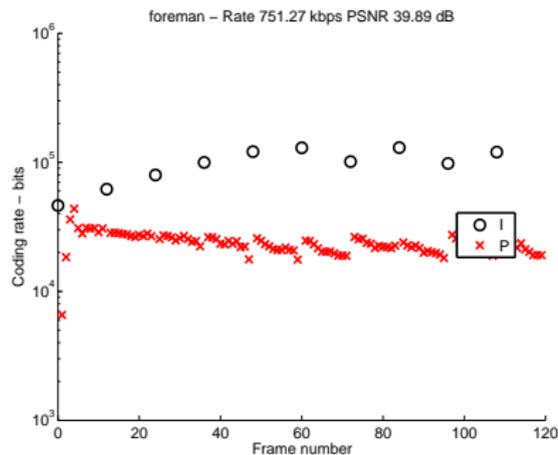
Performance H.264

Effet de la structure du GOP



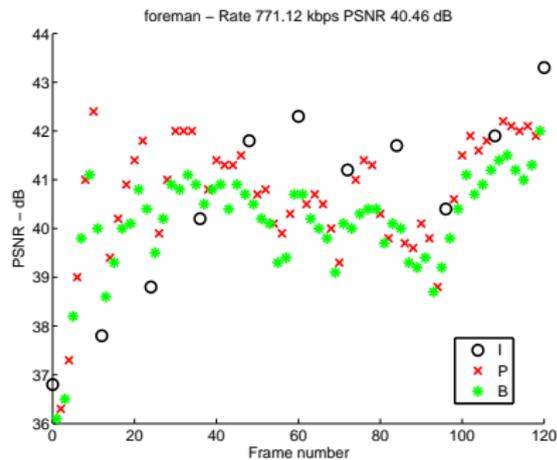
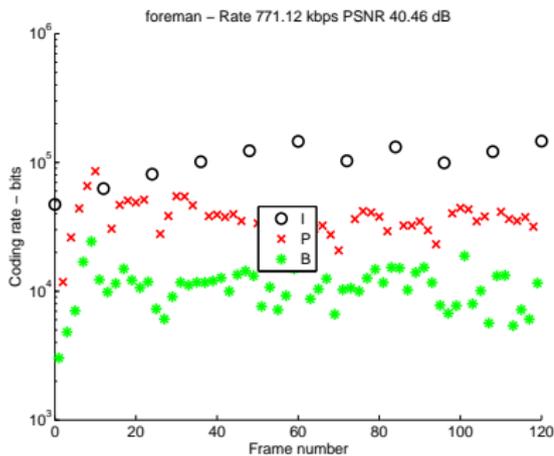
Performance H.264

GOP IPPP : Débit et PSNR par image



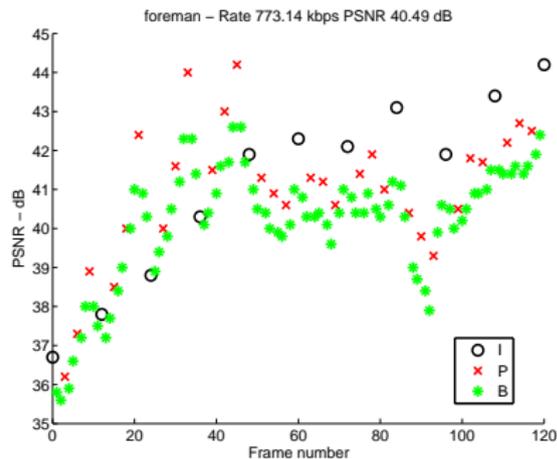
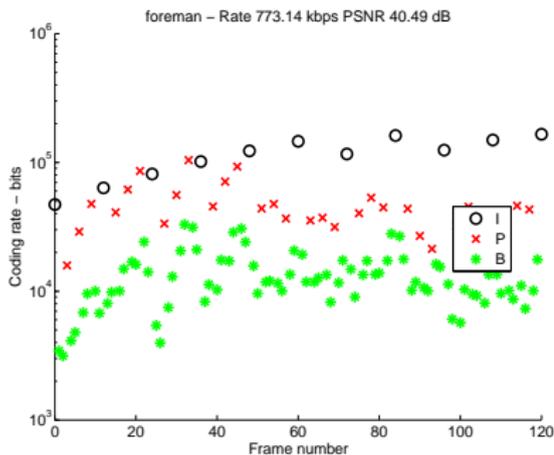
Performance H.264

GOP IBPBP : Débit et PSNR par image



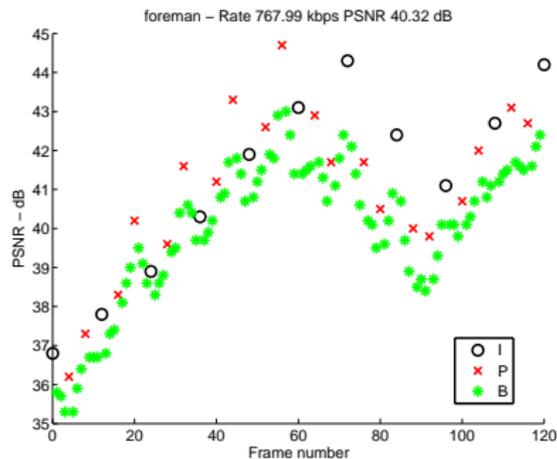
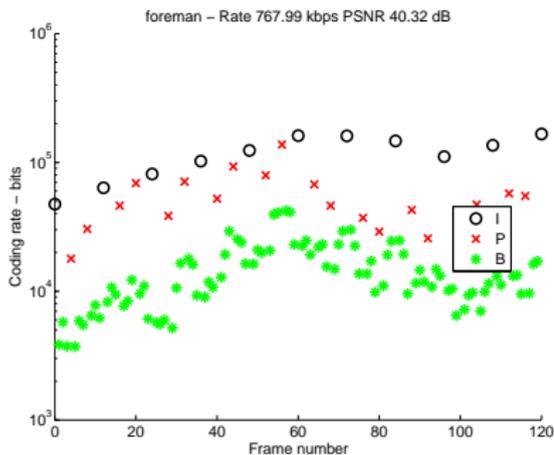
Performance H.264

GOP IBBPBBP : Débit et PSNR par image



Performance H.264

GOP IBBPBBBP : Débit et PSNR par image



Network Abstraction Layer

Le NAL est réalisé en sorte de fournir “network friendliness”

- Simple interface entre le VCL (video coding layer) et les systèmes externes
- “Mapping” des données sur les transport layers :
 - RTP/IP pour les services en temps réel
 - Format de fichiers (ISO MP4) pour le stockage
 - MPEG-2 system pour les service de diffusion
 - H.32X pour les services de visioconférence

NAL units

- La vidéo codée est organisée en NAL units (NALU)
- Une NALU est un paquet avec 1 octet d'entête et $N - 1$ octet de payload
- Le header indique le type de payload
- Dans le payload on introduit, si nécessaire, des *emulation prevention bytes*
- *Byte-Stream* et *Packet-Transport* formats

Byte-Stream format NALU

- Cas où le train de NALU doit être délivré comme une suite ordonnée d'octets
- Exemples : MPEG-2 systems
- Les limites des NALUs doivent être identifiables
- Solution : Start code prefix et emulation prevention bytes

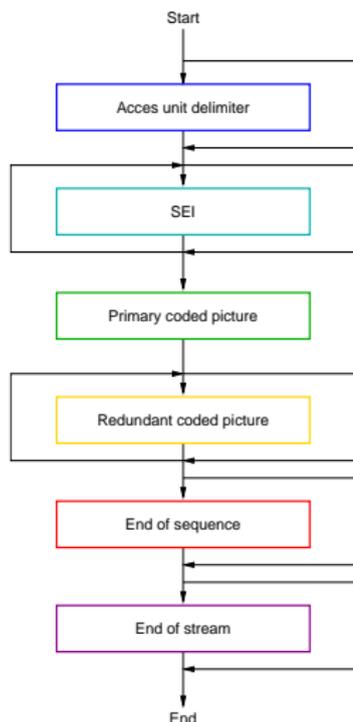
Packet-Transport format NALU

- Cas où les données sont partitionnées par le niveaux de transport
- Exemples : RTP/IP, UDP/IP
- Dans ce cas les start codes seraient un gaspillage de bande

Type de NALU par contenu

- VCL NALU (modes, vecteurs, résidus)
- Non-VCL NALU (header, parameter sets)
 - Parameter set pour image et pour sequence
- On peut appliquer l'UEP

Access unit



- Une access unit (AU) produit une image
- Une *video sequence* est une suite de AUs
- Au début d'une séquence vidéo il y a une image IDR (instantaneous decoding refresh)
- Un NALU stream peut contenir une ou plusieurs *video sequences*

Plan

1 Introduction

2 H.264/AVC

- Définitions et schéma
- Nouveaux modes et outils
- Profils, niveaux, performances
- Network abstraction layer

3 HEVC

- Définitions
- Coding tools
- Stream

High Efficiency Video Coding

- Joint ISO/ITU-T standard
 - MPEG-H Part 2 (ISO/IEC 23008-2)
 - ITU-T H.265
- Objectifs : les mêmes que H.264, plus :
 - Très haute résolution
 - Codage parallèle
 - Performances de compression améliorées
- Encore un codeur hybride avec prédiction temporelle et spatiale

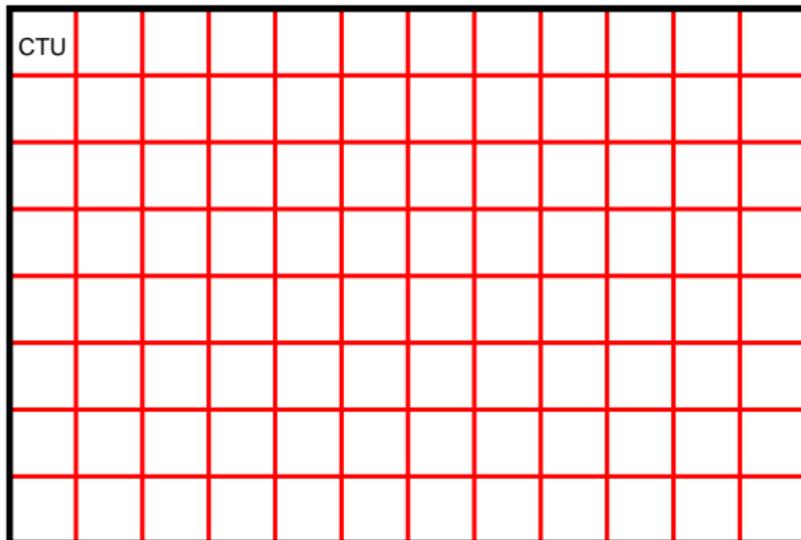
High Efficiency Video Coding

- *Call for Evidence* : Avril 2009
- *Joint Collaborative Team on Video Coding (JVT-VC)* : Jan. 2010
- Call for Proposal : Jan. 2010
- Tests subjectifs jusqu'à Avril 2010
- Test Model under Consideration : Avril 2010
- HEVC Test Model v1 (HM 1.) : Juillet 2010
- Final Draft International Standard : Jan. 2013
- HM 10.0 Avril 2013
- En parallèle : norme de codage 3D en évolution (3D-VC)

Structures de données en HEVC

- Séparation entre unité pour le codage, la prédiction et la transformée
- Units : ensemble de donnée affecté à une même région rectangulaire de la vidéo.
- Blocks : une “unit” correspond à un bloc de luminance et 2 blocs de chrominance
- L'élément fondamental de codage est la *Coding Tree Unit* (CTU)
- Chaque CTU est divisée en coding units (CU)
- Chaque CU est divisé en prediction units (PU) et en transform units (TU)

Coding tree units



Coding tree units

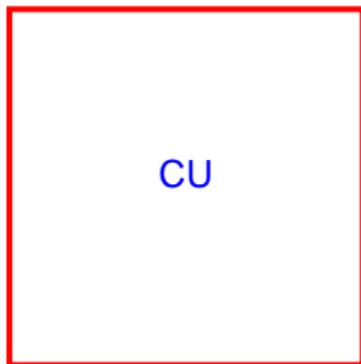
- L'image courante (picture) est divisée en CTU
- Une CTU est formée par un Coding Tree Block (CTB) de luminance et 2 CTB de chroma
- Les CTB de chroma ont une résolution inférieure, car en HEVC la vidéo d'entrée est toujours en 4 : 2 : 0
- La taille des CTU est choisie par l'encodeur
- Trois tailles possibles : 16×16 , 32×32 et 64×64
- Normalement, une plus grande taille donne une meilleure compression (mais cela a un coût en complexité)

Coding units

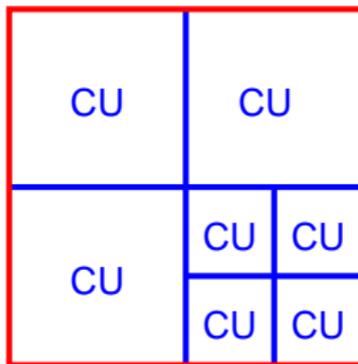
- Les CTU sont divisées en coding units (CU)
- Une CU est formée par un Coding Block (CB) de luma et 2 de chroma
- Les CU sont toujours de forme carrée
- La taille maximale d'une CU est celle de la CTU
- La taille minimale est 8×8
- La structure est celle d'un quad-tree
- La décision entre codage Intra et codage Inter est faite à niveau de CU

Coding units

CTU



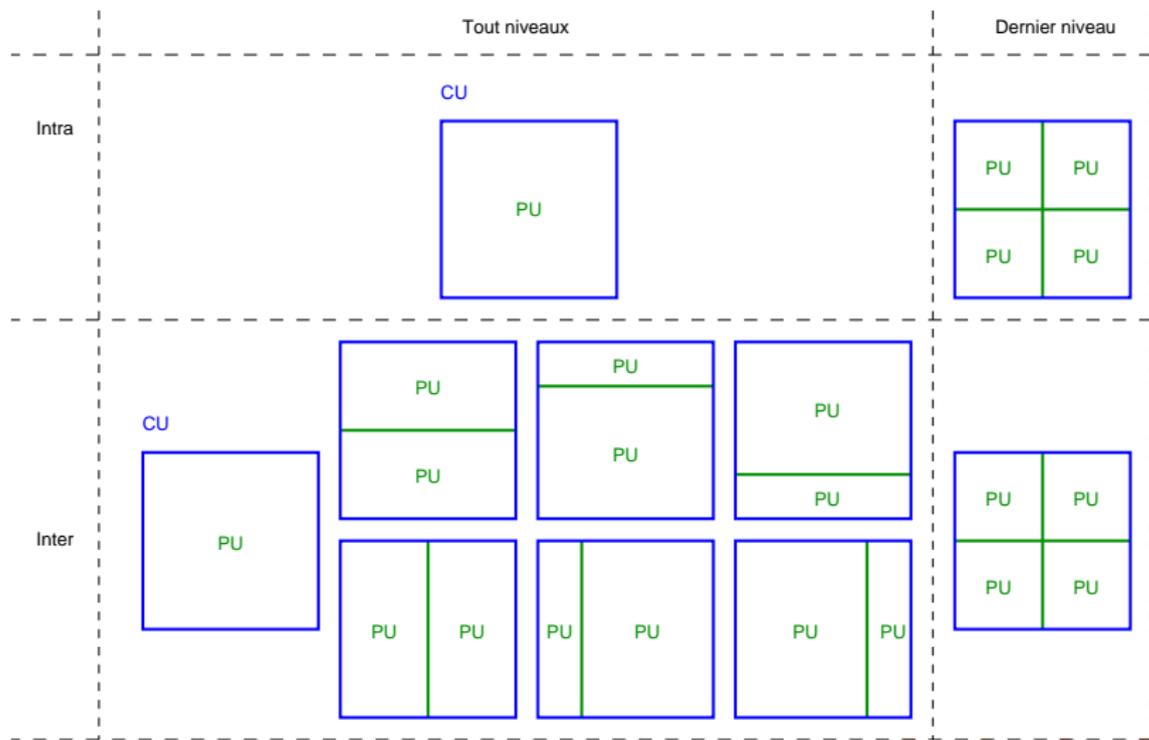
CTU



Prediction units

- Pour une CU Intra, la PU correspond à la CU
 - Sauf pour de CU de dernier niveau (les plus petites possibles) qui peuvent encore être décomposées en 4 PU carrées
- Pour le CU Inter, la CU peut correspondre à la PU...
- ... ou peut être coupée en 2 (six choix possibles) PU
- ou, uniquement pour le dernier niveau, divisée en 4 PU
- Une PU est formée par un PB de luma et 2 de chroma
- Pour chaque PU on code l'information de prédiction (direction de prédiction en Intra, information de mouvement en Inter)
- Le résidu est transformé dans la TU, qui peut avoir une forme différente de la PU

Prediction units



Transform units

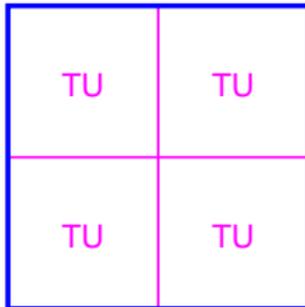
- Les CU peuvent être divisées en TU avec un schéma en quadtree
- Cela est indépendant de la partition en PU
- Une TU peut donc s'étendre sur plusieurs PU ...
- ... ou au contraire ne correspondre qu'à une partie d'une PU
- Le découpage est implicite si la taille de la CU est supérieure à la taille maximale des TUs
- Le non-découpage est implicite si la taille de la TU est égale à la taille minimale

Transform units

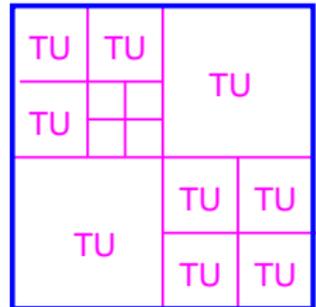
CU



CU



CU



Codage Intra

- Le mode de prédiction Intra est choisi à niveau de PB
- Le PB correspond au CB à l'exception du dernier niveau
- 35 mode de codage Intra sont possible pour la luminance
- 33 directions + IntraDC + IntraPlanar
- Cela est indépendant de la taille du bloc (entre 4×4 et 32×32)
- Les directions de prédiction sont plus denses en proximité de l'horizontale et de la verticale
- La prédiction est calculée avec une interpolation bi-linéaire
- Les valeurs de référence et sur les contours peuvent être adoucis pour réduire les artefacts

Codage Intra

- Le codage du mode est fait en créant une liste de 3 modes probables (MPM, most probable modes)
- La création de la MPM est possible du côté du décodeur à l'identique
- On utilise les voisins de haut et gauche si disponibles et si codé en Intra, et on remplace les valeurs non disponible avec des modes de défaut (planar, DC, vertical)
- Si le mode luma est un des 3 MPM, on code l'index 0, 1 ou 2
- Sinon, on code le mode entre les 32 restants sur 5 bits
- Pour le PB chroma on peut choisir seulement entre Planar, V, H, DC et "Direct", c'est à dire le même que luma
- Le mode chroma est codé directement (sans MPM)

Codage Inter

- Le CB peut correspondre au PB ...
- ... ou peut être coupé en 2 PB : horizontal haut, milieu, bas ; vertical gauche, centre, droite
- Le CB ne peut pas être coupé en 4 PB (dans ce cas, il serait plus efficace de couper à niveau de CB, est utiliser en suite des PB de la même taille que les nouveaux CB)
- Seule exception : si le CB ne peut pas être ultérieurement découpé

Codage Inter

- Comme dans H.264, la prédiction Inter peut être faite en utilisant des références multiples dans une liste
- On a deux listes pour les slices de type B
- L'estimation de mouvement est faite au quart de pixel
- Deux filtres différents sont utilisés pour interpoler les positions au demi-pixel et au quart de pixel

Codage Inter

Merge

- Le mode “merge” permet de partager l’information de mouvement entre PB voisins
- Ce mode est similaire au mode SKIP de H.264 avec deux différences
 - Plusieurs candidats existent pour l’information de mouvement, et on transmet explicitement lequel est choisi
 - On transmet aussi explicitement quel est l’image de référence utilisé (en H.264 cela était déduit)

Codage Inter

Merge

- L'information de mouvement partagée consiste en
 - Un ou deux vecteurs de mouvement
 - Un ou deux index de référence
- On crée une liste de C candidats possibles pour le mode merge
- Le nombre C est spécifié dans les entêtes de la slice
- Si on peut pas coder l'information de mouvement avec le merge, on code la différence par rapport à une prédiction
- Cette prédiction est choisie entre 2 des candidats du merge

Codage Inter

Merge

- Les candidats du merge sont, selon leur disponibilité :
 - Cinq candidats spatiaux (vecteurs des PB voisins)
 - Un candidat temporel (vecteur du bloc en bas à droite dans l'image de référence, ou sinon du bloc dans la même position)
 - Candidats "générés", jusqu'à avoir C candidats différents (les doublons sont enlevés)

Transformée

- Quatre tailles de transformée sont possibles : 32×32 , 16×16 , 8×8 et 4×4
- Transformées séparables, à coefficients entiers, qui approximent une DCT
- Les matrices plus petites sont un sous-échantillonnage des plus grandes
- Sur les blocs 4×4 on peut utiliser une transformée alternative, qui approxime la DST et qui est mieux adapté à représenter les résidus sur les petits blocs
- La DST réduit le débit de 1 % pour les blocs 4×4 (en Intra), et de façon négligeable pour les grand blocs

Codage entropique

- HEVC utilise une version améliorée et simplifiée de CABAC
- Les contextes sont mieux choisis, les coefficients sont toujours lus par blocs 4×4 , mais avec 3 ordres possibles (raster H, raster V, diagonal vers droite-haut)

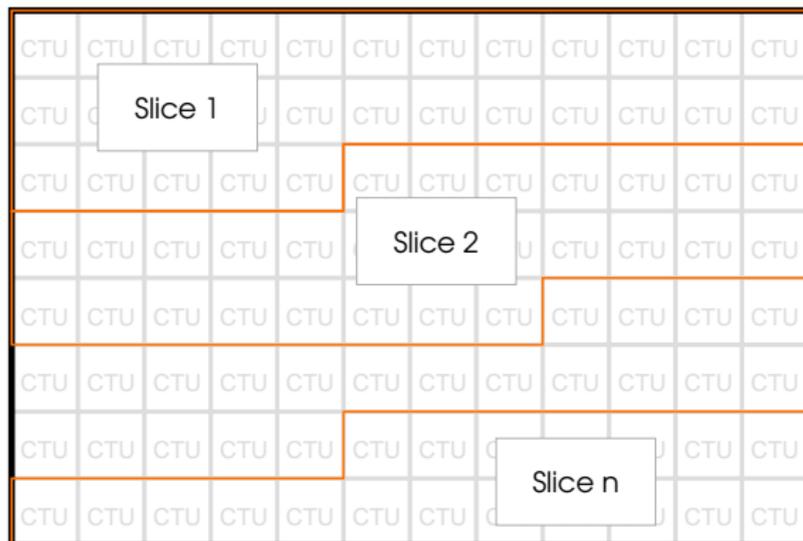
Deblocking filter

- Un nouveau filtre de deblocking est implémenté, avec l'objectif de réduire la complexité
- Il est appliqué seulement sur une grille 8×8
- Cela réduit la complexité sans trop dégrader les performances
- Seulement 3 niveaux de filtrage sont possibles
- Un filtre SAO (sample adaptive offset) est appliqué après le deblocking
- Le SAO ajoute un offset en fonction de la valeur du pixel et des caractéristiques de la région (plate, minimum, contour, maximum)

High level syntax: slices

- Les CTU peuvent être regroupées en slices ou en tiles
- Les slices sont formées par une suite de CTU en *raster scan order*
- Comme en H.264 les slices sont *self-contained* : la prédiction Intra ne se fait pas entre une slice et un autre
- Les slices sont de type I, P et B
- Le but principale des slices est de permettre la re-synchronisation après une perte de données

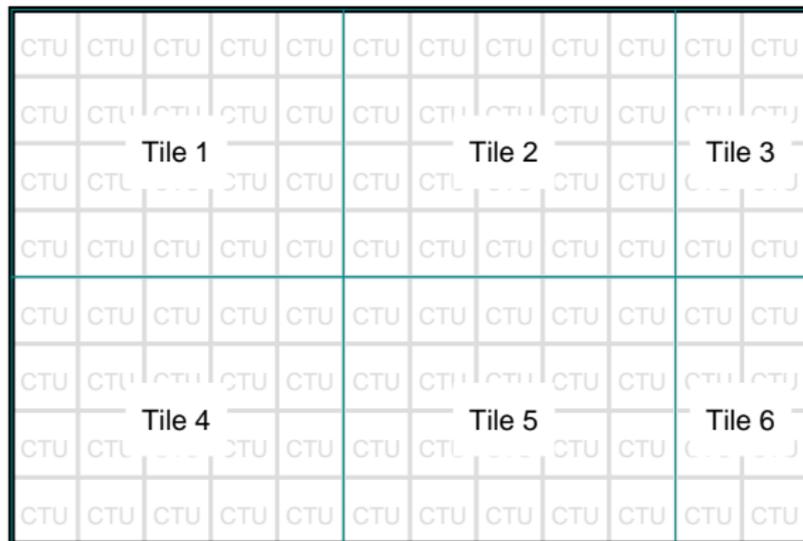
Partition d'une picture en slices



High level syntax: tiles

- Les CTU peuvent aussi être regroupées en régions rectangulaires, les “tiles”
- Les tiles sont aussi “self-contained”
- Les tiles sont indépendantes des slices
- Les tiles ont comme objectif celui de permettre un niveau grossier de parallélisme (les tiles peuvent être décodées en parallèle)

Partition d'une picture en tiles

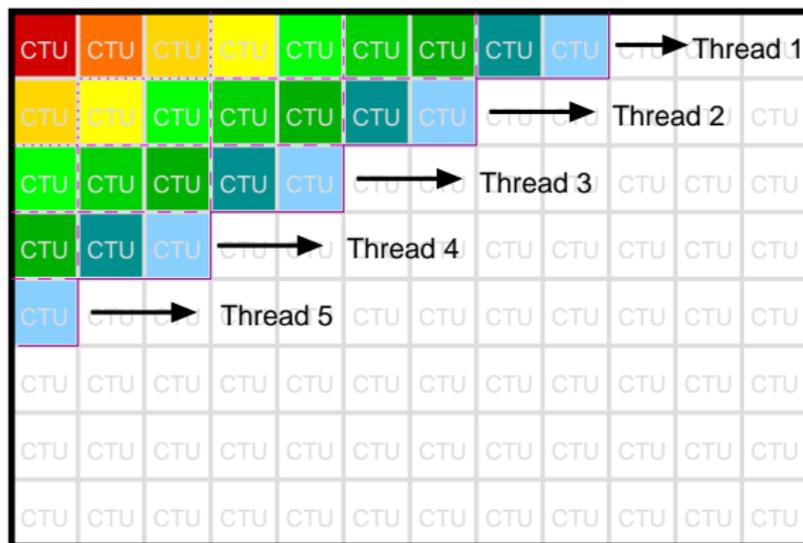


Traitement par wavefront

- Un degré plus fin de parallélisme peut être obtenu avec le wavefront parallel processing (WPP)
- Les slices sont divisées en lignes
- Le traitement d'une ligne peut commencer dès que les deux premières CTU de la ligne immédiatement supérieure est terminé
- Il y a toujours un retard minimum de 2 CTU entre deux lignes consécutives

Traitement par wavefront

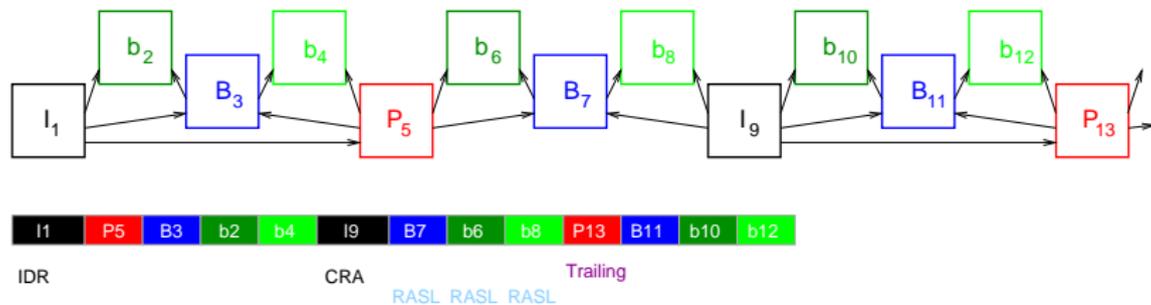
Wavefront



Random access et bitstream splicing

- Le décodage d'une séquence HEVC peut commencer aux random access points (RAP)
- Un RAP peut être
 - Une image IDR : tout ce qui la suit dans le bitstream peut être décodé indépendamment du passé
 - Une image CRA (clean random access) : elle est suivie dans le bit-stream par des images temporellement précédentes, éventuellement non décodables
 - Une image BLA (broken link access) : on a coupé le bitstream à niveau d'un CRA et on la "renoué" (splicing) avec un autre bitstream

Structure du stream



Acronymes

- BLA** Broken Link Acces
- CRA** Clean Random Access
- IDR** Instantaneous Decoding Refresh
- RADL** Random Access Decodable Leading
- RAP** Random Access Point
- RASL** Random Access Skipped Leading

Conclusions

Pour approfondir

- 1 A. Bovik. *Handbook of image and video processing*
- 2 I. Richardson. *H.264 and MPEG-4 Video Compression*
- 3 Special issues de *IEEE Trans. Circ. Video Tech.*