

Ανοικτό Πανεπιστήμιο Κύπρου
Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
στα Πληροφοριακά Συστήματα



Τίτλος:

Μετάδοση Υπηρεσιών Video Streaming σε Ασύρματα Δίκτυα με Βάση το HEVC.

Μετάδοση Υπηρεσιών Video Streaming σε Ασύρματα Δίκτυα με Βάση το HEVC.

Μαρία Καλαθά

Επιβλέπων Καθηγητής
Αναστάσιος Νταγιούκλας

Ιανουάριος 2014

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

**Τίτλος: Μετάδοση Υπηρεσιών Video Streaming σε
Ασύρματα Δίκτυα με Βάση το Πρότυπο HEVC.**

Μαρία Καλαθά

Επιβλέπων Καθηγητής

Αναστάσιος Νταγιούκλας

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών στα Πληροφορικά Συστήματα από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου

Ιανουάριος 2014

Περίληψη

Σκοπός της πτυχιακής είναι η μελέτη και αξιολόγηση της κωδικοποίησης HEVC για υπηρεσίες video streaming σε ασύρματα δίκτυα με διαφορετικό ρυθμό απωλειών.

Στην παρούσα εργασία παρουσιάζεται το νέο πρότυπο H.265 για την κωδικοποίηση και μετάδοση video και οι διαφορές του με το μέχρι σήμερα επικρατών πρότυπο H.264. Αναπτύσσεται επίσης σε γλώσσα προγραμματισμού C, ένα μοντέλο απωλειών σε επίπεδο bit το οποίο εισάγει λάθη στο κωδικοποιημένο βίντεο προκειμένου να εκτιμηθεί και πειραματικά η αποδοτικότητα του για την μετάδοση του.

Summary

The purpose of the project was to study and evaluate the HEVC coding for video streaming services in wireless networks with different rate losses.

This paper presents a new model H. 265 for encoding and transmitting video and differences with the hitherto dominant standard H.264. Grows well in programming language C, a model losses in bit level which introduces errors in the encoded video to determine experimentally the efficiency of the transmission.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα Καθηγητή μου κ. Αναστάσιο Νταγιούκλα για την πολύτιμη βοήθειά που μου προσέφερε για την εκπόνηση αυτής της Μεταπτυχιακής Διατριβής. Επίσης θα ήθελα να ευχαριστήσω θερμά την οικογένεια μου και να αφιερώσω την παρούσα Μεταπτυχιακή Διατριβή στο δίδυμο αδερφό μου Χρίστο† , ο οποίος πίστευε σε μένα και ήταν δίπλα μου “φύλακας άγγελος”.

Περιεχόμενα

1. Εισαγωγή.....	6
2. Εισαγωγικές έννοιες	8
2.1 Ψηφιακό Βίντεο.....	8
2.2 Μέθοδοι μέτρησης ποιότητας βίντεο.....	8
2.3 Υποκειμενικές μέθοδοι.....	9
2.4 Αντικειμενικές μέθοδοι.....	9
2.5 α. Υπολογισμός PSNR ως μετρητή της ποιότητας βίντεο	10
β. PEVQ (Perceptual Evaluation of Video Quality).....	13
3. Το πρότυπο H.265.....	14
3.1 Σύγκριση του HEVC με το H.264.....	16
3.2 Κωδικοποίηση	17
3.3 Υψηλού επιπέδου αρχιτεκτονική σύνταξης.....	22
3.4 Σύνταξη παράλληλης επεξεργασίας και δομές επεξεργασίας slice	23
4. Προσομοίωση απωλειών πακέτων	25
4.1 Σύστημα Κωδικοποίησης HEVC.....	25
4.2 Διάγραμμα Ροής της εφαρμογής	31
4.3 Πειραματικά Σενάρια δυο ακολουθιών βίντεο.....	32
4.4 Συμπεράσματα -αποτελέσματα της προσομοίωσης.....	35
4.5 Βοηθητικές έννοιες.....	38
4.6 Το μέλλον.....	41
Παράρτημα	43
Άρθρα-Πηγές- Βιβλιογραφία.....	54

Κεφάλαιο 1

1. Εισαγωγή

Σκοπός της παρούσης διπλωματικής διατριβής είναι να παρουσιαστεί το πρότυπο του κωδικοποιητή HEVC/H.265 και να αξιολογηθεί για υπηρεσίες Video Streaming σε ασύρματα δίκτυα με διαφορετικό ρυθμό απωλειών. Σημαντικά χαρακτηριστικά που μας ενδιαφέρουν για την ασύρματη μετάδοση βίντεο είναι η αποδοτικότητα συμπίεσης (compression efficiency), κατανάλωση ισχύος (power dissipation) και ο έλεγχος σφαλμάτων (error control).

Ο υψηλής απόδοσης κωδικοποιητής βίντεο HEVC ή High Efficiency Video Coding ή H.265 είναι μορφή (format) συμπίεσης βίντεο, διάδοχος του H.264/MPEG-4 ή AVC (Advanced Video Coding). Το πρότυπο HEVC αποτελεί το πιο πρόσφατο προϊόν της συνεργασίας του ITU-T Video Coding Experts Group και του ISO/IEC MPEG (η συνεργασία αυτή χαρακτηρίζεται από την ονομασία Joint Collaborative Team on Video Coding - JCT-VC).

Το HEVC αναμένεται να διπλασιάσει τον όγκο δεδομένων υπό συμπίεση σε σύγκριση με H.264/MPEG-4 AVC διατηρώντας το ίδιο επίπεδο ποιότητας βίντεο.

Το HEVC προβλέπεται να επεκταθεί στην ικανοποίηση των απαιτήσεων διαφορετικού τύπου εφαρμογών, στην κωδικοποίηση του video και στις απαιτήσεις της 3D προβολής.

Στο πεδίο που αναφέρεται στη μετάδοση υψηλής ευκρίνειας βίντεο, γίνονται συνεχείς έρευνες για την βελτίωση της απόδοσης. Οι περισσότερες εφαρμογές βίντεο απαιτούν τη, σε πραγματικό χρόνο, μετάδοση μέσω ,τις περισσότερες φορές, ετερογενών περιβαλλόντων και δικτύων. Η απαίτηση αυτή εξυπηρετείται ικανοποιητικά μόνο με την ταχύτερη μετάδοση της πληροφορίας αλλά και την μείωση των σφαλμάτων μετάδοσης στο ελάχιστο δυνατό. Έτσι αναζητούνται και προτείνονται κωδικοποιήσεις οι οποίες καταφέρνουν να διατηρούν την ποιότητα της εικόνας σε υψηλό επίπεδο και παράλληλα να μειώνουν σε μεγάλο βαθμό τον απαιτούμενο χρόνο μετάδοσης. Σημαντική εξέλιξη αυτής προσπάθειας είναι η αντικατάσταση του προτύπου H.264/MPEG-4 AVC, από το πρότυπο High Efficiency Video Coding (HEVC/H.265), το οποίο έχει την υποστήριξη των κατασκευαστών τηλεοπτικών δεκτών, καθώς και του γνωστού Ινστιτούτου Fraunhofer.

Στην παρούσα εργασία παρουσιάζεται το νέο πρότυπο HEVC και οι διαφορές του με το μέχρι σήμερα επικρατών πρότυπο H.264. Επίσης, παρουσιάζεται σε γλώσσα C ένα μοντέλο απωλειών πακέτων σε κίνηση δικτύου καθώς και η παραγωγή και μετάδοση πακέτων με την κωδικοποίηση του νέου προτύπου HEVC, προκειμένου να εκτιμηθεί και πειραματικά η αποδοτικότητα του για την μετάδοση βίντεο μέσω δικτύου.

Η μετάδοση στο δίκτυο είναι μια πολύ απαιτητική εφαρμογή που δεσμεύει μεγάλη ποσότητα πόρων(εύρος ζώνης). Ωστόσο, το πρόβλημα που αντιμετωπίζουμε στα ασύρματα δίκτυα είναι ο περιορισμός των πόρων αυτών με αποτέλεσμα στις περισσότερες περιπτώσεις να μην είναι εφικτή η μετάδοση του βίντεο.

Τεχνικές που αναπτύχθηκαν για να αντιμετωπίσουν το πρόβλημα έλλειψης πόρων για τη μετάδοση βίντεο εφαρμόζουν κωδικοποίηση βίντεο. Η κωδικοποίηση βίντεο εκμεταλλεύεται το στατιστικό πλεονασμό που υπάρχει στο βίντεο, π.χ. δυο ίδιες διαδοχικές εικόνες, παράγοντας έτσι ένα κωδικοποιημένο αρχείο μικρότερου μεγέθους που ονομάζεται ροή από bit(bitstream). Όταν ολοκληρωθεί η μετάδοση, εφόσον δεν υπήρξε κάποια απώλεια πακέτου, ο παραλήπτης μπορεί να αποκωδικοποιήσει το βίντεο.

2. Εισαγωγικές έννοιες

2.1 Ψηφιακό βίντεο

Το ψηφιακό βίντεο μπορεί να προκύψει είτε με απευθείας "σύλληψη" του σε ψηφιακό μέσο όπως για παράδειγμα ψηφιακή βιντεοκάμερα είτε με μετατροπή ενός αναλογικού σήματος. Το ψηφιακό βίντεο είναι ένα σύνολο στατικών εικόνων που εναλλάσσονται γρήγορα πολλές φορές το δευτερόλεπτο. Συνεπώς διακρίνουμε ότι το video "κληρονομεί" τα χαρακτηριστικά των εικόνων που το αποτελούν (χρωματική και χωρική ανάλυση - color special resolution) και έχει επίσης το χαρακτηριστικό της συχνότητας εναλλαγής τους.

2.2. Μέθοδοι μέτρησης ποιότητας βίντεο

Τα συστήματα επεξεργασίας βίντεο μπορεί να επιφέρουν ορισμένα ποσά παραμόρφωσης στο σήμα του βίντεο, έτσι ώστε η αξιολόγηση της ποιότητας βίντεο είναι ένα σημαντικό πρόβλημα.

Από την στιγμή κατά την πρώτη καταγραφή ακολουθίας βίντεο στον κόσμο, πολλά συστήματα επεξεργασίας βίντεο έχουν σχεδιαστεί. Στις μέρες μας, τα ψηφιακά συστήματα βίντεο αντικαθιστούν τα αναλογικά, και οι μέθοδοι αξιολόγησης έχουν αλλάξει. Η απόδοση ενός συστήματος ψηφιακής επεξεργασίας βίντεο μπορεί να ποικίλει σημαντικά και εξαρτάται από τα δυναμικά χαρακτηριστικά του σήματος εισόδου βίντεο (π.χ. το ποσό της κίνησης). Γι ' αυτό η ποιότητα του ψηφιακού βίντεο θα πρέπει να αξιολογείται σε διαφορετικές ακολουθίες βίντεο, συχνά από τη βάση δεδομένων του χρήστη.

2.3 Υποκειμενικές Μέθοδοι (Subjective Methods)

Ο κύριος στόχος πολλών αντικειμενικών μετρικών εκτίμησης της ποιότητας του βίντεο είναι να εκτιμηθεί η διατύπωση γνώμης του μέσου χρήστη (θεατή) σχετικά με την ποιότητα του βίντεο που επεξεργάζεται το σύστημα.

Μερικές φορές, ωστόσο, η μέτρηση της υποκειμενικής ποιότητας βίντεο μπορεί επίσης να είναι προκλητική, διότι μπορεί να απαιτεί την κρίση ενός ειδικά εκπαιδευμένου.

Πολλές «υποκειμενικές μετρήσεις της ποιότητας του βίντεο» περιγράφονται στο ITU-T σύσταση BT.500. Κεντρική ιδέα τους είναι η μέση βαθμολογία (Mean Opinion Score) για ήχο: Τα βίντεο-ακολουθίες εμφανίζονται σε ομάδα τηλεθεατών και στη συνέχεια η γνώμη τους καταγράφεται και κατά μέσο όρο αξιολογείται η ποιότητα της κάθε ακολουθίας βίντεο. Ωστόσο, τα στοιχεία των δοκιμών μπορεί να ποικίλλουν σε μεγάλο βαθμό.

2.4 Αντικειμενικές Μέθοδοι (Objective Methods)

Οι Αντικειμενικές Μέθοδοι αξιολόγησης βίντεο είναι μαθηματικά μοντέλα τα οποία προσεγγίζουν τα αποτελέσματα της αξιολόγησης της υποκειμενικής ποιότητας, αλλά με βάση κριτήρια και μετρήσεις που μπορούν να μετρηθούν αντικειμενικά και αυτόματα και αξιολογούνται από ένα πρόγραμμα υπολογιστή.

Οι Αντικειμενικές μέθοδοι ταξινομούνται με βάση τη διαθεσιμότητα του αρχικού σήματος βίντεο, το οποίο θεωρείται ότι είναι υψηλής ποιότητας (γενικά όχι συμπιεσμένο).

Ως εκ τούτου, μπορούν να ταξινομηθούν ως μέθοδοι αναφοράς Full (Full Reference Methods-FR) , μειωμένης αναφοράς μέθοδοι (Reduced Reference Methods-RR) και No - Μέθοδοι Αναφοράς (No-Reference Methods-NR) .

Οι μετρήσεις FR υπολογίζουν τη διαφορά ποιότητας, συγκρίνοντας κάθε pixel σε κάθε εικόνα της παραμορφωμένης εικόνας στο αντίστοιχο εικονοστοιχείο της στο αρχικό βίντεο .

Οι μετρήσεις RR βγάζουν ορισμένα χαρακτηριστικά και των δύο βίντεο και η σύγκρισή τους δίνει μια βαθμολογία ποιότητας . Χρησιμοποιούνται όταν όλα τα πρωτότυπα βίντεο δεν είναι διαθέσιμα, π.χ. σε μία μετάδοση με περιορισμένο εύρος ζώνης(bandwidth).

Οι NR μετρήσεις προσπαθούν να αξιολογήσουν την ποιότητα ενός παραμορφωμένου βίντεο, χωρίς καμία αναφορά στο αρχικό βίντεο . Οι μετρήσεις αυτές συνήθως χρησιμοποιούνται όταν είναι γνωστή η μέθοδος κωδικοποίησης βίντεο.

2.5 Υπολογισμός του PSNR (peak signal-to-noise ratio) ως μετρητή ποιότητας

Οι πιο παραδοσιακοί τρόποι αξιολόγησης της ποιότητας του ψηφιακού συστήματος επεξεργασίας βίντεο (π.χ. κωδικοποιητές βίντεο, όπως DivX, Xvid) είναι υπολογισμός του λόγου σήματος προς θόρυβο (signal-to-noise ratio - SNR) και της κορυφής σήματος προς θόρυβο (peak signal-to-noise ratio - PSNR) μεταξύ του αρχικού σήματος βίντεο και του σήματος που διέρχεται μέσω του συστήματος.

Ο υπολογισμός του PSNR είναι η πιο ευρέως χρησιμοποιούμενη αντικειμενικής ποιότητας βίντεο μετρική. Ωστόσο, οι τιμές PSNR δεν συσχετίζονται τέλεια με μια αντιληπτή οπτική ποιότητα λόγω της μη γραμμικής συμπεριφοράς του ανθρώπινου οπτικού συστήματος .

Κατά την εκτίμηση της ποιότητας ενός κωδικοποιητή βίντεο, όλες οι προαναφερθείσες αντικειμενικές μέθοδοι μπορεί να απαιτούν την επανάληψη των

post-encoding tests -δοκιμών μετά την κωδικοποίηση προκειμένου να καθορίσουν τις παραμέτρους κωδικοποίησης που πληρούν ένα απαιτούμενο επίπεδο οπτικής ποιότητας, που τις καθιστά χρονοβόρες, περίπλοκες και μη πρακτικές για εφαρμογή σε πραγματικές εμπορικές εφαρμογές .

Για το λόγο αυτό , μεγάλο μέρος της έρευνας έχει επικεντρωθεί στην ανάπτυξη νέων αντικειμενικών μεθόδων αξιολόγησης που επιτρέπουν την πρόβλεψη του επιπέδου της ποιότητας των κωδικοποιημένων βίντεο πριν από την πραγματοποίηση της πραγματικής(actual) κωδικοποίησης.

Το PSNR, είναι ο λόγος μεταξύ της μέγιστης δυνατής ισχύος του σήματος και της εξουσίας του θορύβου διαφθοράς (corrupting noise) που επηρεάζει την πιστότητα της αναπαράστασης της. Επειδή πολλά σήματα έχουν ένα πολύ μεγάλο δυναμικό εύρος, το PSNR εκφράζεται συνήθως σε σχέση με την λογαριθμική κλίμακα ντεσιμπέλ(dB).

Το PSNR πιο συχνά χρησιμοποιείται για τη μέτρηση της ποιότητας της ανοικοδόμησης lossy κωδικοποιητών συμπίεσης (lossy compression codecs), όπως για παράδειγμα στη συμπίεση εικόνας.

Το σήμα σε αυτή την περίπτωση είναι τα αρχικά δεδομένα, και ο θόρυβος είναι το σφάλμα που εισάγεται με τη συμπίεση. Κατά τη σύγκριση των κωδικοποιητών συμπίεσης, το PSNR είναι μια προσέγγιση για την ανθρώπινη αντίληψη της ποιότητας της ανασυγκρότησης.

Παρά το γεγονός ότι ένα υψηλότερο PSNR(πάνω από 35 dB) γενικά δείχνει ότι η ανοικοδόμηση είναι υψηλότερης ποιότητας, σε ορισμένες περιπτώσεις, μπορεί και να μην είναι.

Κάποιος πρέπει να είναι εξαιρετικά προσεκτικός με το φάσμα ισχύος αυτής της μετρικής. Είναι αδιαμφισβήτητα έγκυρο μόνο όταν χρησιμοποιείται για να συγκρίνει τα αποτελέσματα από τον ίδιο κωδικοποιητή (ή τύπο κωδικοποιητή) και το ίδιο περιεχόμενο.

Το PSNR ορίζεται μέσω του Μέσου Τετραγωνικού σφάλματος (Mean Squared Error-MSE). Δοθέντος μιας χωρίς θόρυβο $m \times n$ ασπρόμαυρης εικόνας I και της θορυβώδης προσέγγισης του K ,

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

Το PSNR ορίζεται ως:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10} (MAX_I) - 10 \cdot \log_{10} (MSE) \end{aligned}$$

όπου MAX_I : η μέγιστη πιθανή τιμή του εικονοστοιχείου (pixel) της εικόνας.

Όταν τα εικονοστοιχεία αντιπροσωπεύονται χρησιμοποιώντας 8 bits ανά δείγμα, αυτό είναι 255.

Γενικότερα, όταν τα δείγματα αντιπροσωπεύονται χρησιμοποιώντας LinearPCM, MAX_I is $2^B - 1$.

Για έγχρωμες εικόνες με τρεις τιμές RGB ανά pixel, ο ορισμός του PSNR είναι ο ίδιος, εκτός από το MSE που είναι το άθροισμα όλων των διαφορών υψωμένων στο τετράγωνο διαιρεμένο με το μέγεθος της εικόνας και από το τρία.

Εναλλακτικά, για έγχρωμες εικόνες η εικόνα μετατρέπεται σε ένα διαφορετικό χώρο χρώματος και PSNR έχει αναφερθεί έναντι κάθε καναλιού του εν λόγω χώρου χρώματος, π.χ., YCbCr ή HSL.

Τυπικές τιμές για το PSNR σε lossy συμπίεση εικόνας και βίντεο είναι μεταξύ 30 και 50 dB, εφόσον το βάθος bit είναι 8 bit, όπου υψηλότερη είναι η καλύτερη. Για τα δεδομένα 16 Bit είναι τυπικές τιμές για το PSNR μεταξύ 60 και 80 dB. Οι αποδεκτές τιμές για την απώλεια ποιότητα της ασύρματης μετάδοσης θεωρείται ότι είναι περίπου 20 dB έως 25 dB.

Εν απουσία θορύβου, οι δύο εικόνες I και K είναι ταυτόσημες, και έτσι η MSE είναι μηδέν. Στην περίπτωση αυτή το PSNR είναι απροσδιόριστο (βλέπε διαίρεση με το μηδέν).

PEVQ (Perceptual Evaluation of Video Quality)

Ο αλγόριθμος μέτρησης **PEVQ** (Perceptual Evaluation of Video Quality) αποτελεί έναν αλγόριθμο για την αξιολόγηση της ποιότητας της εικόνας ενός βίντεο.

PEVQ είναι αλγόριθμος πλήρους αναφοράς(FR) και αναλύει pixel-by-pixel την εικόνα μετά από μια προσωρινή ευθυγράμμιση (που αναφέρεται επίσης ως «προσωρινή καταχώρηση») των αντίστοιχων πλαισίων αναφοράς και του σήματος(test signal). Το PEVQ MOS(Perceptual Evaluation of Video Quality Mean Opinion Score) κυμαίνεται από 1 (κακή) έως 5 (άριστα).

Ο αλγόριθμος μέτρησης μπορεί να εφαρμοστεί για την ανάλυση visible artifacts που προκαλούνται από την κωδικοποίηση/αποκωδικοποίηση ενός ψηφιακού βίντεο, για τα δίκτυα RF-ή IP-based δικτύων.

Μπορεί να εφαρμοστεί σε δίκτυα επόμενης γενιάς και τις υπηρεσίες κινητών τηλεφώνων και στις IPTV τηλεοράσεις τυπικής και υψηλής ευκρίνειας, στα βίντεο συνεχούς ροής και στην τηλεφωνία μέσω βίντεο.

Κεφάλαιο 3

Το πρότυπο HEVC/H.265

Περιγραφή

Ο υψηλής απόδοσης κωδικοποιητής βίντεο HEVC ή High Efficiency Video Coding ή H.265 είναι μορφή(format) συμπίεσης βίντεο, διάδοχος του H.264/MPEG-4 ή AVC (Advanced Video Coding). Το πρότυπο HEVC αποτελεί το πιο πρόσφατο προϊόν της συνεργασίας του ITU-T Video Coding Experts Group και του ISO/IEC MPEG (η συνεργασία αυτή χαρακτηρίζεται από την ονομασία Joint Collaborative Team on Video Coding - JCT-VC).

Το HEVC αναμένεται να διπλασιάσει τον όγκο δεδομένων υπό συμπίεση σε σύγκριση με H.264/MPEG-4 AVC διατηρώντας το ίδιο επίπεδο ποιότητας βίντεο.

Η πρώτη έκδοση του προτύπου ολοκληρώθηκε και δημοσιεύθηκε στις αρχές του 2013. Το έργο προβλέπεται να επεκταθεί στην ικανοποίηση των απαιτήσεων διαφορετικού τύπου εφαρμογών, στην κωδικοποίηση του video και στις απαιτήσεις της 3D προβολής.

Η συνεργασία των δύο οργανισμών και η ανάπτυξη του νέου προτύπου βασίστηκε στις παρελθούσες επιτυχημένες συνεργασίες τους στα H.262/MPEG-2 και H.264/MPEG-4 πρότυπα, τα οποία εφαρμόστηκαν με μεγάλη επιτυχία και παρουσίασαν υψηλού βαθμού διεισδυτικότητα. Έτσι το νέο πρότυπο έχει σαν κύριο στόχο να αποτελέσει μία αναβαθμισμένη συνέχεια αυτών. Το πιο πρόσφατο από αυτά απετέλεσε το θεμέλιο της προσπάθειας αυτής, αφού αναπτύχθηκε την περίοδο 1999 - 2003 και αναβαθμίστηκε την περίοδο 2003 - 2009. Είναι μέχρι και σήμερα το βασικό πρότυπο για την κωδικοποίηση και την μετάδοση video. Χρησιμοποιείται σε μία μεγάλη γκάμα εφαρμογών όπως μετάδοση υψηλής

ευκρίνειας (High Definition - HD) τηλεοπτικού σήματος μέσω δορυφορικών και ενσύρματων ζεύξεων, συστήματα διαχείρισης και επεξεργασίας video, συσκευών λήψης, συστήματα ασφαλείας, διαδικτυακό και ασύρματο δικτυακό video, Blu-ray Discs και video επικοινωνίες πραγματικού χρόνου.

Η ομάδα ανάπτυξης, εκτός από τη γραπτή περιγραφή του νέου μοντέλου, παρέχει στις εταιρείες που δραστηριοποιούνται στην κατασκευή συσκευών ή την παροχή υπηρεσιών βίντεο, υποδειγματικό κώδικα χρησιμοποίησης του.

Η συνεχής ανάδειξη νέων υπηρεσιών, το αυξανόμενο πλήθος του κοινού που επιθυμεί πρόσβαση σε HD video και η εμφάνιση νέων HD μορφών, δημιουργούν την ανάγκη για αποδοτικότερη κωδικοποίηση σε σχέση με αυτήν του H.264/MPEG-4. Ειδικότερα όταν η κινούμενη εικόνα συνοδεύεται από ποιοτικό ήχο. Οι απαιτήσεις γίνονται μεγαλύτερες αν τεθούν υπόψη ότι οι εφαρμογές βίντεο προσανατολίζονται σε εφαρμογές για έξυπνα κινητά τηλέφωνα και συναφείς συσκευές (smartphones, tablets) ή τα συστήματα παροχής υπηρεσιών video on demand. Παράλληλα πάντα εξετάζεται η δυνατότητα επίτευξης βελτιωμένης ποιότητας εικόνας. Το HEVC σχεδιάστηκε για να εξυπηρετήσει αποδοτικότερα τις εφαρμογές που μέχρι σήμερα καλύπτονται από το H.264/MPEG-4 και επιπλέον να επιτύχει αύξηση της ποιότητας του βίντεο και τη δυνατότητα της παράλληλης επεξεργασίας.

Στο πρότυπο της κωδικοποίησης του HEVC η δομή του bitstream και οι αντιστοιχίες που χρησιμοποιούνται για την αποκωδικοποίηση των εικόνων παραμένουν ίδιες. Η αντιστοίχιση (mapping) προκύπτει από τη διαδικασία του καθορισμού του τρόπου που με τον οποίο κάθε αποκωδικοποιητής αντιλαμβάνεται το λαμβανόμενο bitstream και πως αυτός το μετατρέπει σε εικόνα. Στόχος είναι το σύνολο των συσκευών να παράγει την ίδια εικόνα από το ίδιο bitstream.

3.1 Σύγκριση του HEVC με το H.264

Ο H.264/AVC ήταν ο κυρίαρχος κωδικοποιητής βίντεο αλλά δεν ήταν πολύ αποδοτικός όταν κωδικοποιούσε υψηλής ανάλυσης βίντεο. Λόγω της ανάγκης για κάλυψη σε νέες εφαρμογές που απαιτούσαν υψηλής ανάλυσης εικόνες αναπτύχθηκε ο κωδικοποιητής βίντεο ο HEVC/H.265.

Η νέα συμπίεση που χρησιμοποιείται από το High Efficiency Video Coding –HEVC– έχει διπλάσια αποδοτικότητα σε σχέση με αυτή του H.264. Αυτό σημαίνει ότι για την ίδια ποιότητα εικόνας ο HEVC χρησιμοποιεί σχεδόν το μισό bit-rate σε σχέση με τον H.264.

Ο νέος κωδικοποιητής, ο οποίος υποστηρίζει ανάλυση 8K, χρησιμοποιεί μεγαλύτερα μπλοκ εικόνας, μέχρι 64x64pixels (αντί 16x16 του H.264 ή AVC), καλύτερη υποκατηγοριοποίηση της εικόνας σε μεταβλητού μεγέθους περιοχές με quad-free partitioning (αντί των συμμετρικών του AVC) και 35 intra-modes.

Προτάθηκε από την Motion Pictures Expert Group (MPEG) ως μία προσπάθεια για αύξηση της συμπίεσης του μεταδιδόμενου βίντεο με ταυτόχρονη διατήρηση της ποιότητας σε υψηλά επίπεδα ώστε να χρησιμοποιείται λιγότερο εύρος ζώνης(bandwidth) και να είναι εφικτή η αναμετάδοση ή το κατέβασμα βίντεο με διπλάσια ταχύτητα.

Το βασικότερο πλεονέκτημα του νέου προτύπου είναι ότι απαιτεί το μισό εύρος ζώνης για τη μετάδοση περιεχομένου υψηλής ευκρίνειας σε σχέση με το H.264. Αντί για στοιχεία 16x16 pixels, το νέο πρότυπο χωρίζει τη συνολική εικόνα σε στοιχεία 64x64 pixels και αυτό οδηγεί σε κέρδος στην απεικόνιση της κίνησης της εικόνας.

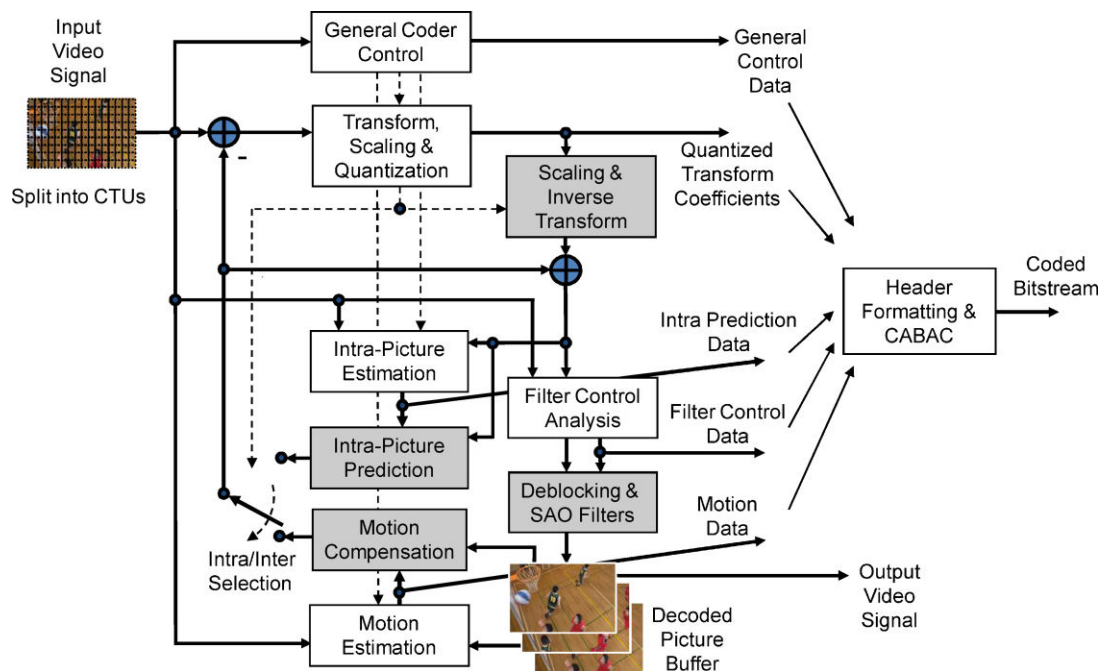
3.2 Κωδικοποίηση

Το πρότυπο H.265 σχεδιάστηκε για να επιτευχθεί:

- Αποδοτική κωδικοποίηση
- Εύκολη μετάδοση σε ετερογενή συστήματα
- Δυνατότητα παράλληλης επεξεργασίας

Το επίπεδο κωδικοποίησης του βίντεο στο HEVC χρησιμοποιεί την ίδια υβριδική προσέγγιση που χρησιμοποιείται σε όλα τα αντίστοιχα πρότυπα από το H.261 και μετά.

Στην εικόνα 2.1.1 παριστάνεται ένας υβριδικός κωδικοποιητής βίντεο που παράγει bitstream που συμμορφώνεται στο πρότυπο του HEVC. Κάθε εικόνα διαιρείται σε περιοχές (blocks) και κάθε ένα από αυτά οδηγείται στον αποκωδικοποιητή.



Εικόνα 1 Τυπικός HEVC video encoder(decoder modeling elements - γκρι χρώμα), Διαδικτυακή Πηγή:[01].

Η πρώτη εικόνα της ακολουθίας ενός video κωδικοποιείται με την χρήση μόνο της εσωτερικής πρόβλεψης (intrapicture prediction). Η τεχνική αυτή χρησιμοποιεί κάποια πρόβλεψη δεδομένων με βάση τις περιοχές της ίδιας εικόνας και δεν βασίζεται σε άλλες εικόνες. Όλες οι υπόλοιπες εικόνες της ακολουθίας εφαρμόζεται η τεχνική της μεταξύ των εικόνων πρόβλεψης (interpicture prediction). Η τεχνική αυτή βασίζεται στην επιλογή μίας βασικής εικόνας και ενός διανύσματος κίνησης (Motion Vector - MV) ώστε να γίνει η πρόβλεψη για την μορφή του κάθε block. Τόσο ο κωδικοποιητής όσο και ο αποκωδικοποιητής δημιουργούν κατάλληλες interpicture προβλέψεις για τα σήματα που στέλνουν ή λαμβάνουν με την εφαρμογή Motion Compensation (MC), την χρήση των MV και mode decision δεδομένα που παρέχονται σαν μετα-πληροφορία.

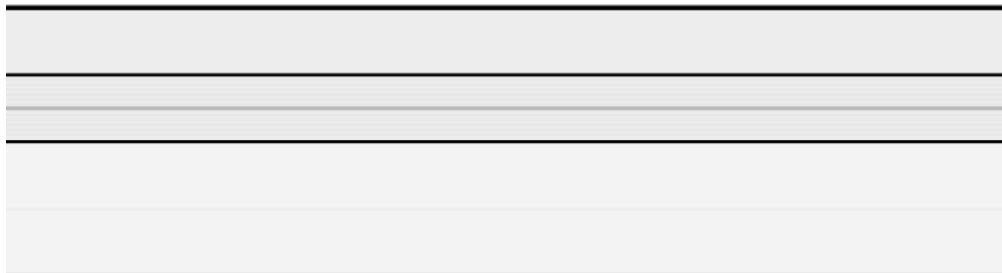
Ο κωδικοποιητής διπλασιάζει τον κύκλο επεξεργασίας του αποκωδικοποιητή έτσι ώστε και οι δύο να παράγουν κατάλληλες προβλέψεις για διαδοχικά δεδομένα. Έτσι τα αποτελέσματα της κβαντικής μετατροπής σχηματοποιούνται με inverse scaling του παρεμβαλλόμενου σήματος. Η εικόνα που προκύπτει αποθηκεύεται για να χρησιμοποιηθεί για την πρόβλεψη των επόμενων εικόνων.

Το πρότυπο προβλέπει ταυτόχρονα και την χρήση μετά-πληροφοριών ώστε να αντιμετωπιστεί και η περίπτωση της επεξεργασίας interlaced πλαισίων ώστε αυτά να αποστέλλονται σαν HEVC κωδικοποιημένες εικόνες. Έτσι επιτυγχάνεται η κωδικοποίηση τέτοιου είδους βίντεο χωρίς να είναι απαραίτητο να επιβαρύνεται ο αποκωδικοποιητής με την ανάγκη υποστήριξης επιπλέον διαδικασιών.

Τα βασικά στοιχεία της διαδικασίας είναι:

- **Coding Tree Units και Coding Tree Blocks:** Η καρδιά της κωδικοποίησης στα παρελθόντα πρότυπα ήταν η χρήση των macroblocks (16×16 block luma samples, 4:2:0 color sampling, δύο αντίστοιχα 8×8 blocks chroma samples). Το αντίστοιχο στοιχείο για το HEVC είναι το Coding Tree Unit το οποίο μπορεί να έχει μέγεθος τουλάχιστον όσο ένα παραδοσιακό macroblock το οποίο μπορεί να επεκταθεί ανάλογα με τον κωδικοποιητή. Αυτό αποτελείται από 16×16 block luma samples, 4:2:0 color sampling, δύο αντίστοιχα 8×8 blocks chroma samples. Υποστηρίζεται επίσης η διαίρεση του σε μικρότερα στοιχεία χρησιμοποιώντας την δενδροειδή μορφή.

- **Coding Units (CU) και Coding Blocks (CB):** Η σύνταξη των CTU καθορίζουν το μέγεθος και την θέση του luma και του chroma του CB. Η ρίζα του είναι σχετική με το CTU. Έτσι το μέγεθος του luma CTB είναι το μέγιστο που μπορεί να υποστηρίξει ένα luma CB. Η διάσπαση του CTU σε luma και chroma CB μετατρέπεται σε σήμα. Ένα luma CB και δύο chroma CB με την κατάλληλη σύνταξη σχηματίζουν ένα CU. Ένα CTB μπορεί να περιέχει μόνο ένα CU ή μπορεί να διασπάται σε πολλαπλά CU, με κάθε ένα από αυτά να έχει ένα αντίστοιχο μέρος σε Prediction Unit και σε δένδρο από Transform Units (TU). Στην επόμενη εικόνα φαίνεται η διαίρεση ενός CTB σε CB. Οι συνεχόμενες γραμμές προσδιορίζουν τα όρια των Tree Blocks ενώ φαίνεται και το αντίστοιχο quadtree.



Εικόνα 2 διαίρεση ενός CTB σε CB Διαδικτυακή Πηγή:[01].

- **Prediction Units (PU) και Prediction Blocks (PB):** Η απόφαση για το αν η κωδικοποίηση μίας εικόνας θα γίνει με intrapicture ή interpicture prediction γίνεται σε επίπεδο CU. Μία PU διάσπαση έχει την ρίζα της στο επίπεδο του CU. Ανάλογα με τον βασικό τύπο πρόβλεψης το Luma και το chroma CB μπορεί επιπλέον να διασπαστεί σε μικρότερο μέγεθος και να προβλεφτεί από τα luma και chroma prediction blocks. Το HEVC υποστηρίζει μεγέθη από 64X64 μέχρι 4X4.
- **Transform Units (TU) και Transform Blocks (PB):** Το prediction residual κωδικοποιείται με την χρήση block μετατροπών. Ένα TU δένδρο έχει την ρίζα του στο CU επίπεδο. Το υπόλοιπο luma CB μπορεί να είναι ίδιο με το luma Transform Block (TB) ή μπορεί αν διασπαστεί σε μικρότερα luma TB. Το ίδιο εφαρμόζεται και για το chroma TB. Χρησιμοποιούνται κατάλληλες συναρτήσεις 4X4, 8X8, 16X16 και 32X32.
- **Motion vector signaling:** Χρησιμοποιείται Advanced motion vector prediction (AMVP). Αυτή περιλαμβάνει την παραγωγή διάφορων πιο πιθανών υποψηφίων Motion Vector(με βάση τα δεδομένα του σχετικού PB και της εικόνας αναφοράς). Επίσης μπορεί να χρησιμοποιηθεί μία ενσωμάτωση για την κωδικοποίηση του MV που επιτρέπει την ιεράρχηση των MV τυχαία ή βασισμένη σε γειτονικά PB.
- **Motion Compensation(αντιστάθμιση κίνησης):** αλγοριθμική τεχνική για την κωδικοποίηση δεδομένων βίντεο για τη συμπίεση του βίντεο (video compression). Η Motion compensation περιγράφει μια εικόνα όσον αφορά το μετασχηματισμό μιας εικόνας αναφοράς (reference picture) προς την τρέχουσα (current picture). Η εικόνα αναφοράς μπορεί να είναι χρονικά προηγούμενη ή επόμενη. Στο MPEG(Motion Picture Experts Group), οι εικόνες προβλέπονται από προηγούμενα πλαίσια (P frames) ή bidirectionally-προηγούμενα και επόμενα πλαίσια (B frames). Μετά την πρόβλεψη των πλαισίων χρησιμοποιώντας Motion Compensation, ο κωδικοποιητής βρίσκει το λάθος (residual) το οποίο μετά συμπιέζεται και μεταδίδεται.

- **Intrapicture prediction:** Τα αποκωδικοποιημένα οριακά samples των επικολλούμενων blocks χρησιμοποιούνται σαν αναφορές για χωρική πρόβλεψη σε περιοχές όπου δεν έχει επιτευχθεί interpicture prediction. Η intrapicture prediction υποστηρίζει 33 κατευθυντικές μορφές (εν συγκρίσει με τις 8 αντίστοιχες του H.264/MPEG-4), planar και DC μορφές πρόβλεψης. Οι επιλεγμένες intrapicture prediction μορφές κωδικοποιούνται στις πιο πιθανές καταστάσεις βασισμένες σε αυτές των πιο πρόσφατα αποκωδικοποιημένων γειτονικών PB.

- **Quantization control:** Όπως και στο H.264/MPEG-4, η ομοιόμορφη κβαντική αναδόμηση (uniform reconstruction quantization - URQ) χρησιμοποιείται στο HEVC, με κατάλληλες κβαντικές μετρικές κλίμακες που υποστηρίζονται για τα διάφορα μεγέθη μετατροπών.

- **Entropy encoding:** Για την κωδικοποίηση της εντροπίας χρησιμοποιείται η Context adaptive binary arithmetic coding - CABAC). Είναι σχεδόν ίδιο με το σχήμα που χρησιμοποιείται στο H.264/MPEG-4.

- **In-loop deblocking filtering:** Χρησιμοποιείται ένα μη γραμμικό amplitude mapping σε κάθε interpicture prediction κύκλο μετά από το φιλτράρισμα του deblocking. Σκοπός του είναι η βέλτιστη αναδόμηση του αρχικού σήματος με την χρήση ενός πίνακα αναζήτησης που περιγράφεται από μερικές επιπλέον καταγραφές που μπορεί να επεξεργαστούν με ιστογραμμική ανάλυση στην πλευρά του κωδικοποιητή.

3.3 Υψηλού επιπέδου αρχιτεκτονική σύνταξης

Ένας αριθμός από νέες τεχνικές σύνταξης στο HEVC αυξάνουν την ευελιξία του για λειτουργία του σε μία πλειάδα συσκευών, εφαρμογών και δικτυακών περιβαλλόντων καθώς και βελτιώνουν την αντοχή σε απώλειες δεδομένων. Από την άλλη πλευρά έχει εγκαταλειφθεί η αντίστοιχη αρχιτεκτονική που χρησιμοποιεί το H.264/MPEG-4. Έτσι το νέο πρότυπο χρησιμοποιεί τα ακόλουθα χαρακτηριστικά:

- **Δομή του συνόλου παραμέτρων:** Το σύνολο των παραμέτρων περιλαμβάνει πληροφορίες που μπορεί να διαμοιραστούν για την αποκωδικοποίηση διαφόρων περιοχών του αποκωδικοποιημένου video. Η δομή του συνόλου παραμέτρων προσφέρει έναν μηχανισμό για τη μεταφορά των δεδομένων τα οποία είναι ζωτικά για την διαδικασία της αποκωδικοποίησης. Οι ακολουθίες και τα σύνολα παραμέτρων εικόνας του H.264/MPEG-4 αυξήθηκαν από την δομή του συνόλου παραμέτρων video (video parameter set - VPS).
- **Σύνταξη δομής μονάδας NAL:** Κάθε δομή σύνταξης τοποθετείται σε ένα πακέτο που καλείται network abstraction layer (NAL) unit. Με την χρήση μίας επικεφαλίδας μεγέθους δύο byte μπορεί να προσδιοριστεί ο σκοπός των υπόλοιπων δεδομένων του πακέτου.
- **Slices:** Ένα slice είναι μία μονάδα δεδομένων που μπορεί να αποκωδικοποιηθεί ανεξάρτητα από άλλες αντίστοιχες μονάδες της ίδιας εικόνας, υπό την έννοια των entropy coding, signal prediction και residual signal reconstruction. Ένα slice μπορεί να είναι μία εικόνα ή μία περιοχή εικόνας. Ένας από του βασικότερους σκοπούς των slices είναι ο επανασυγχρονισμός σε περιπτώσεις απώλειας δεδομένων. Σε περίπτωση της μετάδοσης πακέτου ο μέγιστος αριθμός payload bits σε ένα slice τυπικά περιορίζεται και ο αριθμός των CTU στο slice κυμαίνεται ανάλογα με το ελάχιστο μέγεθος του overhead του πακέτου ώστε το μέγεθος του πακέτου να μένει μέσα στα επιτρεπτά όρια.

- **Supplemental enhancement information (SEI) and video usability information (VUI) metadata:** Η σύνταξη περιλαμβάνει υποστήριξη για διάφορους τύπους μεταδεδομένων γνωστά σαν SEI και VUI. Τέτοια δεδομένα προσφέρουν πληροφορίες σχετικά με την χρονοθέτηση των εικόνων του video και την κατάλληλη μετάφραση του πλαισίου χρωμάτων που χρησιμοποιείται στο σήμα του video.

3.4 Σύνταξη παράλληλης επεξεργασίας και δομές επεξεργασίας slice

Τρεις καινοτομίες περιλαμβάνονται στο HEVC για την βελτίωση της παράλληλης επεξεργασίας ή για την αλλαγή της δομής των slice δεδομένων για την ενθυσιάκωση των δεδομένων σε πακέτα. Κάθε μία από αυτές εξυπηρετεί μία πλειάδα εφαρμογών. Οι καινοτομίες αυτές είναι:

Tiles: Στο πρότυπο υπάρχει η δυνατότητα της διαίρεσης της εικόνας σε τετραγωνικές περιοχές που καλούνται tiles. Ο κύριος σκοπός τους είναι η αύξηση της ικανότητας για παράλληλη επεξεργασία παρά για την αντιμετώπιση σφαλμάτων. Αποτελούν ανεξάρτητες αποκωδικοποιήσιμες περιοχές μίας εικόνας που κωδικοποιούνται με επιπλέον πληροφορία header. Τα tiles μπορεί επιπλέον να χρησιμοποιηθούν για σκοπούς χωρικής τυχαίας πρόσβασης σε τοπικές περιοχές των εικόνων του video. Μία τυπική ρύθμιση tiles περιλαμβάνει την τμηματοποίηση της εικόνας σε ορθογώνιες περιοχές με περίπου ίδιο πλήθος CTU σε κάθε ένα από αυτά.

Wavefront parallel processing: Όταν ενεργοποιείται ο μηχανισμός του Wavefront parallel processing (WPP), ένα slice διαιρείται σε γραμμές από CTU. Η πρώτη γραμμή υφίσταται επεξεργασία σε μία κανονική μορφή, η δεύτερη ξεκινάει να υφίσταται επεξεργασία μόνο όταν έχει ολοκληρωθεί η επεξεργασία σε δύο CTU της πρώτης γραμμής, η τρίτη γραμμή ξεκινάει να υφίσταται επεξεργασία μόνο όταν έχει ολοκληρωθεί η επεξεργασία σε δύο CTU στην δεύτερη γραμμή κ.ο.κ. Το περιεχόμενο του μοντέλου της κωδικοποίησης της εντροπίας σε κάθε γραμμή αναφέρεται σε εκείνο της προηγούμενης γραμμής με δύο CTU . Το WPP προσφέρει μία μορφή παράλληλης επεξεργασίας σε αρκετά υψηλό επίπεδο granularity(κοκκοποίησης). Το WPP μπορεί συχνά να προσφέρει καλύτερη απόδοση συμπίεσης από τα tiles.

Dependent slice segments: Τα Dependent slice segments αποτελούν δομές που επιτρέπουν στα δεδομένα που σχετίζονται με ένα δεδομένο wavefront entry point ή tile να μεταφερθούν σε ένα ξεχωριστό NAL unit και έτσι να κάνουν τα δεδομένα διαθέσιμα για ένα σύστημα για να ενταχθούν σε πακέτα με μικρή καθυστέρηση σε σχέση μάλιστα με την επιλογή να γινόταν η κωδικοποίηση για όλα μαζί σε ένα slice. Ένα ανεξάρτητο τμήμα slice για ένα wavefront entry point μπορεί να αποκωδικοποιηθεί μόνο αν τουλάχιστον ένα τμήμα της επεξεργασίας ενός άλλου τμήματος slice έχει ολοκληρωθεί. Τα Dependent slice segments είναι κυρίως χρήσιμα σε μικρής καθυστέρησης κωδικοποίηση όπου η χρησιμοποίηση άλλων παρόμοιων τεχνικών δεν είναι αρκούντως αποδοτικές.

Κεφάλαιο 4

4.1 Σύστημα Κωδικοποίησης HEVC

Ο στόχος είναι να μελετήσουμε την απόδοση του HEVC σε ένα δίκτυο με απώλειες. Θεωρούμε την επίδραση των απωλειών του φυσικού καναλιού. Η εφαρμογή Test έχει αναπτυχθεί σε γλώσσα C. Η λειτουργία της μπορεί να διαιρεθεί σε τρία στάδια

- Στο πρώτο στάδιο γίνεται η κωδικοποίηση ενός ασυμπίεστου αρχείου video με τον αλγόριθμο H.265. Η κωδικοποίηση αυτή γίνεται με την βοήθεια του εργαλείου κωδικοποίησης που είναι διαθέσιμο στην ιστοσελίδα:http://x264.fushizen.eu/builds/hevc-hm/hm_12.0_r3541_release.7z, καλείται δηλαδή η εφαρμογή TAppEncoder για το input αρχείο video (στην προσομοίωση χρησιμοποιείται το αρχείο foreman_cif.yuv. Οδηγίες για την χρήση της εφαρμογής TAppEncoder υπάρχουν στο αρχείο readme.txt.(στο παράρτημα).
- Στο δεύτερο στάδιο, το bitstream που δημιουργείται και αποθηκεύεται στο αρχείο hevc.bin περνάει από μία διαδικασία απώλειας πακέτων και δημιουργείται το αρχείο hevc_loss.bin. Στο στάδιο αυτό προσομοιάζεται η μετάδοση του video μέσω καναλιού με σφάλματα και απώλειες πακέτων.

- Στο τελευταίο στάδιο γίνεται η αποκωδικοποίηση του αρχείου `hevc_loss.bin` ώστε να παραχθεί το αρχείο `foreman_cif_loss.yuv`.

Η αποκωδικοποίηση αυτή γίνεται με την βοήθεια του εργαλείου κωδικοποίησης που είναι διαθέσιμο στην ιστοσελίδα: http://x264.fushizen.eu/builds/hevc-hm/hm_12.0_r3541_release.7z, καλείται δηλαδή η εφαρμογή TAppDecoder για το αρχείο video (στην προσομοίωση χρησιμοποιείται το αρχείο `foreman_cif.yuv`). Οδηγίες για την χρήση της εφαρμογής TAppDecoder υπάρχουν στο αρχείο `readme.txt`.

ΜΕΘΟΔΟΛΟΓΙΑ ΕΙΣΑΓΩΓΗΣ ΛΑΘΩΝ ΣΤΟ BITSTREAM

Η συνάρτηση looseit της εφαρμογής Test, με βάση το BER(Bit Error Rate) πηγαίνει σε οποιοδήποτε σημείο του bitstream και το αλλάζει από 1 σε 0 ή αντίστροφα.

Η εφαρμογή Test

```
#include <stdio.h>
#include <stdlib.h>
#define MAXPACKETSIZE 50000 //ΜΕΓΕΘΟΣ ΠΑΚΕΤΩΝ
//ΔΗΜΙΟΥΡΓΙΑ PACKET LOSS ΠΟΥ ΚΑΘΟΡΙΖΕΤΑΙ ΑΠΟ ΤΟ ΑΡΧΕΙΟ ERROR
int looseit(FILE *err){
    int c;
    do {
        c = getc (err);
        if (c == EOF) {
            fseek(err, 0, SEEK_SET);
        }
        else if (c == '0') return 1;
        else return 0;
    } while (1);
}

int main(){
    FILE *in, *out, *err0=NULL;
    unsigned char buf[MAXPACKETSIZE];
    unsigned char nal_type;
    int i;
    int packet_number=0;
    char command_line[1000];
    printf("Starting video encoding!\n");
    printf("\n\n Please Wait!!! \n\n");
    //ΚΩΔΙΚΟΠΟΙΗΣΗ ΤΟΥ video foreman_cif.yuv ΚΑΙ ΠΑΡΑΓΩΓΗ ΤΟΥ
    BINARY FILE hevc.bin
    sprintf(command_line,"%s","TAppEncoder.exe -c
    cfg\\encoder_randomaccess_main.cfg -q 26 -fr 30 -wdt 352 -hgt 288 -f 300 -i
    foreman_cif.yuv -b hevc.bin");
    system(command_line);

    printf("End of video encoding!\n");
```

```

        printf("Starting video transmission \n\n");
printf("A number of packets are going to be lost!!! \n\n");
printf("\n\n Please Wait!!! \n\n");
if (NULL == (in = fopen("rec.yuv", "rb"))) {
    perror("Cannot read input binary file that generated by the video
file!");
    exit(1);
}

if (NULL == (out = fopen("hevc_loss.yuv", "wb"))) {
    perror("Cannot open destination binary file with errors");
    exit(2);
}

if (NULL == (err0 = fopen ("error", "rb"))) {
    perror ("Cannot open error pattern file");
    exit (3);
}
//ΔΗΜΙΟΥΡΓΙΑ ΣΦΑΛΜΑΤΩΝ ΣΤΟ BINARY FILE
buf[0]=fgetc(in);
buf[1]=fgetc(in);
buf[2]=fgetc(in);
buf[3]=fgetc(in);
buf[4] = fgetc(in);
nal_type = buf[4] & 0x0F;

for(i=5; ; i++)
{
    buf[i] = fgetc(in);
    if(feof(in)==0)
    {
        if(buf[i-3]==0 && buf[i-2]==0 && buf[i-1]==0 && buf[i]==1)
        {
            fwrite(buf, sizeof(unsigned char), i-3, out);
            packet_number++;
            fflush(out);
            buf[0]=buf[i-3];
            buf[1]=buf[i-2];
            buf[2]=buf[i-1];
            buf[3]=buf[i];
            break;
        }
    }
}

while(feof(in)==0)

```

```

        {
            for(i=4; ;i++)
            {
                buf[i] = fgetc(in);
                nal_type = buf[4] & 0x1F;
                if(!feof(in))
                {
                    if(buf[i-3]==0 && buf[i-2]==0 && buf[i-1]==0 &&
buf[i]==1 )
                        {
                            if(nal_type==1 || nal_type==20)
                                {
                                    if(err0 != NULL && looseit(err0)){

                                        else
                                        {
                                            fwrite(buf, sizeof(unsigned
char), i-3, out);

                                            fflush(out);

                                        }

                                        else
                                        {
                                            fwrite(buf, sizeof(unsigned char), i-3,
out);

                                            fflush(out);

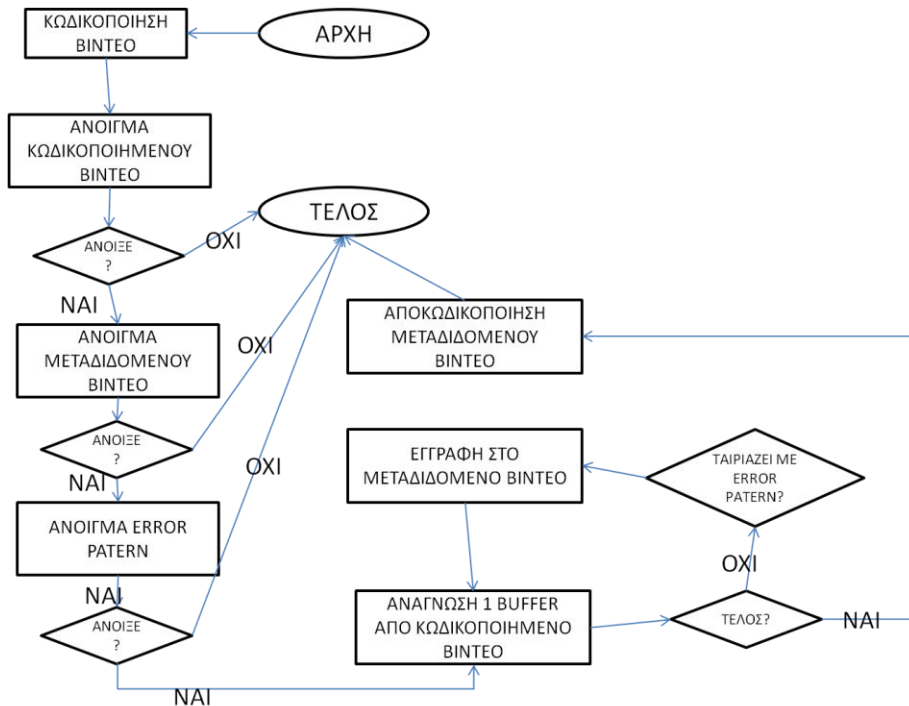
                                        }

                                        packet_number++;
                                        buf[0]=buf[i-3];
                                        buf[1]=buf[i-2];
                                        buf[2]=buf[i-1];
                                        buf[3]=buf[i];
                                        break;
                                    }
                                }
                            else
                            {
                                fwrite(buf, sizeof(unsigned char), i, out);
                                fflush(out);
                                break;
                            }
                        }
                }
            }
        }

```

```
        fclose(in);
        fclose(out);
        if ( err0)
        fclose(err0);
        printf("End of transmtion!!!\n\n");
        printf("Starting video decoding\n\n");
        printf("\n\n Please Wait!!! \n\n");
        sprintf(command_line,"%s","TAppDecoder.exe    -b    hevc_loss.bin    -o
foreman_cif_loss.yuv");
        system(command_line);
        printf("End of video decoding!!!\n\n");
        return 0;
}
```

4.2 Διάγραμμα Ροής εφαρμογής Test



Σχήμα 3









Περιγραφή του διαγράμματος ροής

1. Γίνεται η κωδικοποίηση του video και το αποτέλεσμα σώζεται σε αρχείο.
2. Το νέο συμπιεσμένο αρχείο ανοίγει. Σε περίπτωση που αποτύχει για οποιονδήποτε λόγο το άνοιγμα του αρχείου το πρόγραμμα τερματίζεται.
3. Αν ανοίξει επιτυχώς το αρχείο τότε ανοίγει το αρχείο που θα συγκεντρώσει την προς μετάδοση - συμπιεσμένη - πληροφορία. Αν δεν ανοίξει το αρχείο αυτό για κάποιο λόγο τότε το πρόγραμμα τερματίζεται.
4. Αν ανοίξει και το δεύτερο αρχείο τότε ανοίγει το αρχείο που περιγράφει το ERROR PATTERN. Αν το αρχείο αυτό δεν ανοίξει τότε το πρόγραμμα τερματίζεται.
5. Από το σημείο αυτό και μετά διαβάζεται μία ποσότητα πληροφορίας από το προς μετάδοση αρχείο (video) και μέχρι να ολοκληρωθεί η αναγνωσή του, η πληροφορία αυτή ελέγχεται αν ταιριάζει με το ERROR PATTERN. Αν ταιριάζει τότε η ποσότητα αυτή αντιγράφεται στο δεύτερο αρχείο που άνοιξε.

6. Αφού ολοκληρωθεί η ανάγνωση του πρώτου αρχείου γίνεται η αποκωδικοποίηση του δεύτερου.

4.3 Πειραματικά σενάρια δυο ακολουθιών βίντεο

To link για τα bitstreams είναι <https://www.dropbox.com/s/ij0vbmiwgk9cbog/BER.zip>

	CREW_4CIF_LD_...	3,139,795	3,140,275	VLC media file (.bin)	12/17/2013 8:2...	D7878CFB
	CREW_4CIF_LD_...	3,139,795	3,140,275	VLC media file (.bin)	12/17/2013 8:2...	72AB256B
	CREW_4CIF_RA_...	2,961,756	2,962,211	VLC media file (.bin)	12/17/2013 8:2...	C4015792
	CREW_4CIF_RA_...	2,961,756	2,962,211	VLC media file (.bin)	12/17/2013 8:2...	17EA2214
	SOCCKER_4CIF_L...	2,968,829	2,969,284	VLC media file (.bin)	12/17/2013 8:2...	C63E10FD
	SOCCKER_4CIF_L...	2,968,829	2,969,284	VLC media file (.bin)	12/17/2013 8:2...	D37EEF94
	SOCCKER_4CIF_R...	2,976,568	2,977,023	VLC media file (.bin)	12/17/2013 8:2...	C588FD07
	SOCCKER_4CIF_R...	2,976,568	2,977,023	VLC media file (.bin)	12/17/2013 8:2...	9B912E79

Στα πειραματικά σενάρια κωδικοποιούνται δυο ακολουθίες οι Soccer4CIF, Crew 4CIF και μετριέται η ποιότητα(συμπεριφορά) για δυο διαφορετικά BER(Bit Error Rates). Συγκεκριμένα εξετάζεται η συμπεριφορά του HEVC στις ακολουθίες για τις τιμές του Bit Error Rate(BER): 10^{-6} και 10^{-7} . Οι επιλογές όσον αφορά το HEVC έγιναν με βάση το Low Complexity(Low Delay) και το High Efficiency(Random Access) και παίρνουμε αποτελέσματα. Οι δυο ακολουθίες βίντεο SOCCER, CROWD έχουν τα ακόλουθα χαρακτηριστικά: α) ίδιο αριθμό frames 600(60fps), β) το SOCCER έχει αρκετή κίνηση μεταξύ των καρέ, γ) το CROWD έχει μεγαλύτερη πολυπλοκότητα των frames.

Τα χαρακτηριστικά είναι τα ακόλουθα:

- 4CIF (Common Intermediate Format) ανάλυση για τις δυο ακολουθίες
- GOP size :8, (GOP-Group Of Pictures)
- QP :24, (QP- Quantization Parameter)
- YUV 4:2:0 (color format)

Απο την ιστοσελίδα <http://media.xiph.org/video/derf/> βρίσκουμε τις ακολουθίες Crew και Soccer

- Crew (4:3 | 600 frames) ,Πηγή: <ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/>

Download: 60 **fps** 4CIF (349 MB)

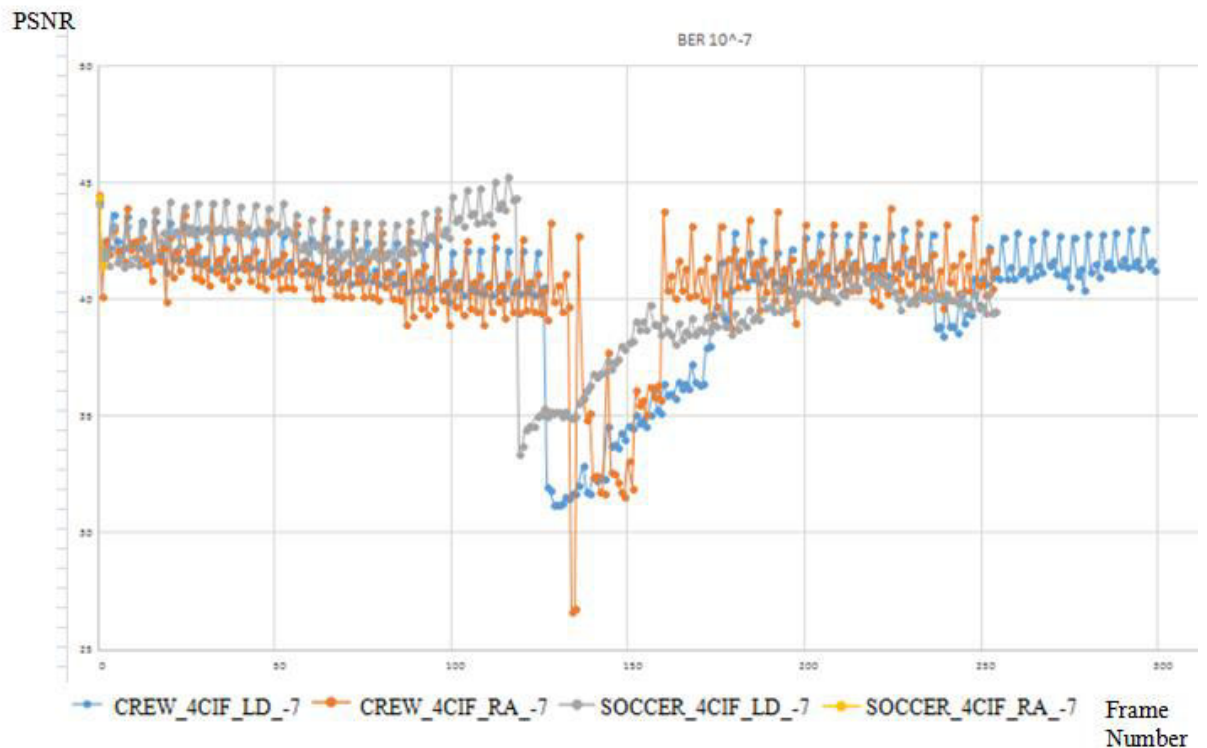
- Soccer (4:3 | 600 frames), Πηγή: <ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/>

Download: 60 fps 4CIF (349 MB)

Στο EXCEL με τη χρήση του λογισμικού YUV ANALYSER (για μέτρηση PSNR)έχουμε για τα CREW_4CIF_LD_-7, CREW_4CIF_RA_-7 και SOCCER_4CIF_LD_-7, SOCCER_4CIF_RA_-7 όπου LD (Low Delay), RA (Random Access) τα ακόλουθα σχήματα.

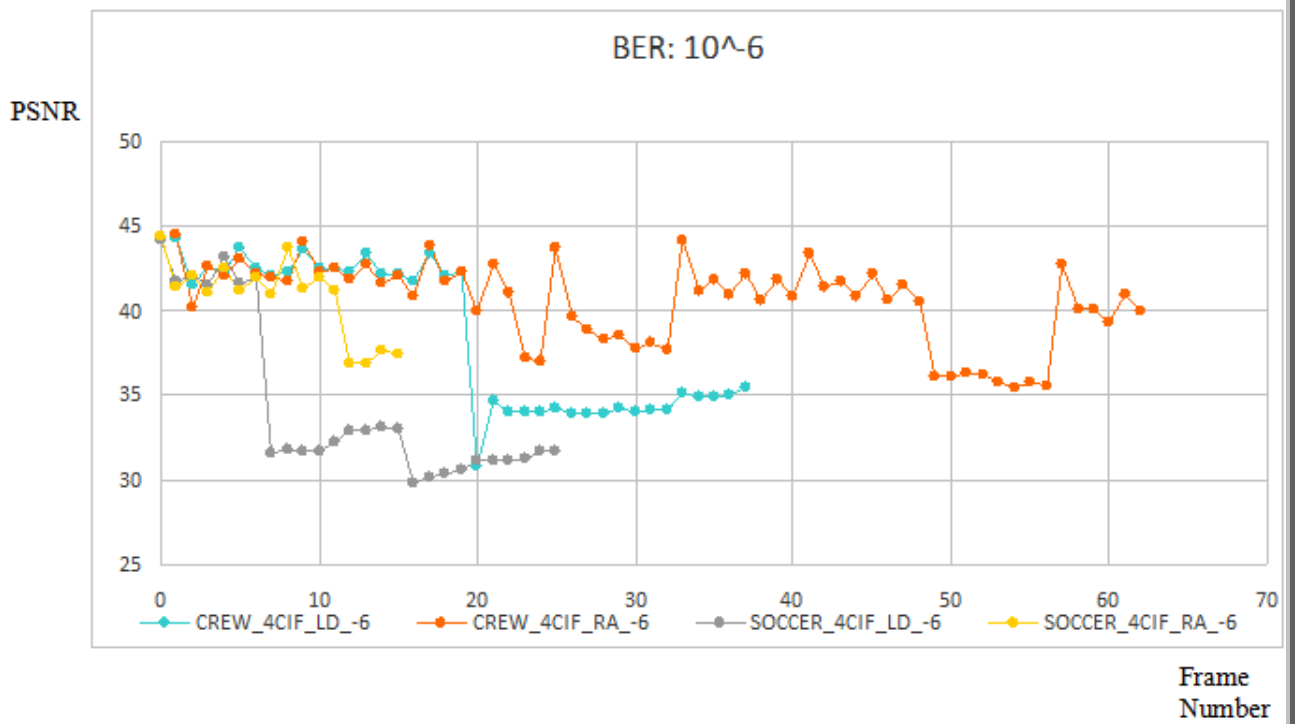
Ο κάθετος άξονας είναι ο Quality Y-PSNR[dB] και ο οριζόντιος άξονας είναι το Frame Number (αφορά τον αριθμό των σωστά ληφθέντων frames).

✚ Για BER 10^{-7} έχουμε



Σχήμα 4

✚ Για BER 10^{-6} έχουμε



Σχήμα 5

Μεθοδολογία

Οι κοινές test conditions κατηγοριοποιούνται ως random access(τυχαία προσπέλαση) και low delay(καθυστέρηση) schemes. Το random access setting επιτρέπει τη δομική καθυστέρηση των processing units(coded frames order). Σε αντίθεση το low delay scheme δεν επιτρέπει κανενός είδους reordering εικόνας ή intra updating.

Ορίζονται δυο configuration schemes για λόγους αξιολόγησης του HEVC.

Συγκεκριμένα το random access mode χρησιμοποιεί B-pictures coding και το low delay mode χρησιμοποιεί P-pictures. Και τα δυο settings υιοθετούν το high profile, για περισσότερο ρεαλιστική αξιολόγηση του νέου standard HEVC.

4.4 Συμπεράσματα- Παρατηρήσεις

Το μετρήσιμο PSNR αφορά τα σωστά ληφθέντα frames.

Με βάση τα στοιχεία του παρακάτω πίνακα

	PSNR	Frame Number	BER
CREW_4CIF_LD_-6	35,449	37	10^{-6}
CREW_4CIF_RA_-6	39,948	62	10^{-6}
SOCCER_4CIF_LD_-6	31,73	26	10^{-6}
SOCCER_4CIF_RA_-6	37,474	16	10^{-6}
CREW_4CIF_LD_-7	41,323	300	10^{-7}
CREW_4CIF_RA_-7	41,325	255	10^{-7}
SOCCER_4CIF_LD_-7	39,592	255	10^{-7}
SOCCER_4CIF_RA_-7	43,202	128	10^{-7}

Πίνακας 6

Βγάζουμε τα ακόλουθα συμπεράσματα:

- Συνοπτικά από τα παραπάνω σχήματα παρατηρούμε ότι κάθε βίντεο δεν αποκωδικοποιείται πλήρως και αυτό φαίνεται από το ότι χάνονται πολλά frames. Ο αριθμός frames που μεταδόθηκε ήταν μεταξύ 16 και 300 σε σύνολο 600 frames!. Όπως προαναφέρθηκε το μετρήσιμο PSNR αφορά τα σωστά ληφθέντα frames και οι τεράστιες απώλειες σημαίνουν μη ικανοποιητική θέαση του βίντεο.
- Παρατηρείται αισθητή απώλεια frames ειδικότερα στην περίπτωση όπου BER μεγαλύτερο του 10^{-7} . Λογικό είναι με βάση τον ορισμό του Bit Error Rate όσο μεγαλώνει το BER(ρυθμός σφαλμάτων ψηφίων) τόσο μεγαλύτερη να είναι και η απώλεια των frames.

Κάτι τέτοιο είναι προφανές βλέποντας τις τιμές του frame number των CREW και SOCCER για LD και RA και BER $10e-6$ ΚΑΙ BER $10e-7$.

Στις περιπτώσεις του Soccer LD και Soccer RA μεταδίδονται 26 και 16 frames για $10e-6$ BER και 255 και 128 frames αντίστοιχα, πολλά περισσότερα, για BER $10e-7$.

Στις περιπτώσεις του CREW LD και CREW RA μεταδίδονται 37 και 62 frames για $10e-6$ BER και 300 και 128 frames αντίστοιχα, πολλά περισσότερα, για BER $10e-7$.

- Η ποιότητα είναι χειρότερη για την περίπτωση BER $10e-6$ σε σχέση με BER $10e-7$ και για τις δυο ακολουθίες και για LD και για RA.

Για το **CREW LD** παρατηρούμε ότι για BER $10e-7$ μεταδόθηκαν 300 frames σε πολύ καλή ποιότητα **41,323** PSNR, ενώ για BER $10e-6$ μεταδόθηκαν μόλις 37 frames σε χειρότερη ποιότητα **35,449** PSNR.

Για το **CREW RA** παρατηρούμε ότι για BER 10e-7 μεταδόθηκαν 255 frames σε πολύ καλή ποιότητα **41,325** PSNR, ενώ για BER 10e-6 μεταδόθηκαν μόλις 62 frames σε χειρότερη ποιότητα **39,948** PSNR.

Για το **SOCCER LD** παρατηρούμε ότι για BER 10e-7 μεταδόθηκαν 255 frames σε πολύ καλή ποιότητα **39,592** PSNR, ενώ για BER 10e-6 μεταδόθηκαν μόλις 26 frames σε χειρότερη ποιότητα **31,73** PSNR.

Για το **SOCCER RA** παρατηρούμε ότι για BER 10e-7 μεταδόθηκαν 128 frames σε πολύ καλή ποιότητα **43,202** PSNR, ενώ για BER 10e-6 μεταδόθηκαν μόλις 16 frames σε χειρότερη ποιότητα **37,474** PSNR.

- Η ποιότητα είναι χειρότερη για την περίπτωση LD απ' ότι για RA και για BER 10e-6 και για BER10e-7 και για τις δυο ακολουθίες.

Άρα στην ποιότητα μεγαλύτερο ρόλο έχει το High Efficiency (random access) παρά το Low complexity (Low Delay).

Για παράδειγμα στην περίπτωση CREW_4CIF_LD_-6 έχουμε PSNR 35,449 ενώ όταν CREW_4CIF_RA_-6 έχουμε μεγαλύτερο PSNR 39,948.

Ομοίως για SOCCER_4CIF_LD_-7 έχουμε PSNR 39,592 ενώ όταν SOCCER_4CIF_RA_-7 έχουμε μεγαλύτερο PSNR 43,202.

- Παρατηρούμε ότι η διαφορά ποιότητας μεταξύ LD(Low Delay), RA(Random Access) είναι πολύ μικρότερη για BER10-7 απ' ότι για BER10e-6 και για το βίντεο SOCCER και για το CREW.

Η διαφορά ποιότητας μεταξύ LD,RA για το SOCCER όταν BER 10e-6 είναι **6**, ενώ για BER 10e-7 είναι πολύ μικρότερη **3, 61**.

Ομοίως η διαφορά ποιότητας μεταξύ LD,RA για το CREW όταν BER 10e-6 είναι **4, 499** ενώ για BER 10e-7 είναι πολύ μικρότερη **0,002**.

Πράγματι,

✚ Για SOCCER

Για BER 10e-6

	PSNR
SOC CER_ 4CIF_ LD_-6	31, 73
SOC CER_ 4CIF_ RA_-6	37, 474

Διαφορά 37,474-31,73=6

Ενώ για BER 10e-7

	PSNR
SOC CER_ 4CIF_ LD_-7	39,592
SOC CER_ 4CIF_ RA_-7	43,202

Διαφορά 43,202-39,592=3,61

✚ Για CREW

Για BER 10e-6

CREW_ 4CIF_ LD_-6	35, 449
CREW_ 4CIF_ RA_-6	39,948

Διαφορά 39,948-35,449=4,499

Ενώ για BER 10e-7

CREW_ 4CIF_ LD_-7	41,323
CREW_ 4CIF_ RA_-7	41,325

Διαφορά 41,325-41,323=0,002

4.5 Βοηθητικές Έννοιες

fps ή frames per second μια μετρική του πόση πληροφορία χρησιμοποιείται για αποθήκευση και προβολή κίνησης βίντεο.

Κάθε καρέ-πλαίσιο(frame) είναι μια εικόνα. Η εμφάνιση frames σε γρήγορη διαδοχή, δημιουργεί την ψευδαίσθηση της κίνησης. Όσο περισσότερα είναι τα καρέ ανά

δευτερόλεπτο (fps), τόσο πιο ομαλή κίνηση εμφανίζεται. Σε γενικές γραμμές, ο ελάχιστος αριθμός fps που απαιτούνται για την αποφυγή απότομης κίνησης είναι περίπου 30. Ορισμένες μορφές βίντεο υπολογιστή, όπως AVI, παρέχουν μόνο 15 καρέ ανά δευτερόλεπτο.

BER-Bit Error Rate

Στη ψηφιακή μετάδοση, ο αριθμός των σφαλμάτων bit είναι ο αριθμός των δυαδικών ψηφίων που λαμβάνονται από μία ροή δεδομένων σε ένα κανάλι επικοινωνίας που έχουν μεταβληθεί λόγω θορύβου, παρεμβολών. Ο ρυθμός σφάλματος ψηφίων - BER είναι ο λόγος του αριθμού των σφαλμάτων δυαδικών ψηφίων προς το συνολικό αριθμό των δυαδικών ψηφίων που μεταδίδονται κατά τη διάρκεια ενός χρονικού διαστήματος που μελετήθηκαν. Ο BER είναι μετρητής της απόδοσης, που συχνά εκφράζεται ως ποσοστό επί τοις εκατό.

Ως παράδειγμα, ας υποθέσουμε η μεταδιδόμενη ακολουθία bit:

0 1 0 1 0 0 1 0 1 1, και η ακόλουθη λαμβανόμενη ακολουθία δυαδικών ψηφίων:

0 0 1 0 1 0 1 0 0 1.

Ο αριθμός των σφαλμάτων bit (τα υπογραμμισμένα bits) είναι στην περίπτωση αυτή 3. Η BER είναι 3 λάθος bits διαιρείται με 10 bits που μεταδίδονται, με αποτέλεσμα το BER είναι 0,3 ή 30%.

Το **Common Intermediate Format-CIF** ορίζει την ανάλυση σε όρους των width και height του βίντεο frame και του frame rate. Το CIF είναι 352 X 240pixels στο National Television System Committee-NTSC format. Άλλη ανάλυση είναι και το **4CIF** (704 X 480pixels).

Bitrate

Το Bitrate είναι ο αριθμός των bits που μεταδίδονται στη μονάδα του χρόνου (bits per second-bps ή kbps ή Mbps ή Gbps

Δομή GOP -Group Of Pictures

Στην κωδικοποίηση βίντεο, μια ομάδα εικόνων, ή η δομή **GOP**, καθορίζει τη σειρά με την οποία τα intra και inter-frames είναι τοποθετημένα. Η GOP είναι μια ομάδα διαδοχικών εικόνων μέσα σε μια κωδικοποιημένη ροή βίντεο. Κάθε ρεύμα κωδικοποιημένων βίντεο αποτελείται από διαδοχικές GOPs.

Ένα **GOP** περιέχει τους ακόλουθους τύπους εικόνας:

- I picture ή I frame (intra coded picture) είναι μια εικόνα που κωδικοποιείται ανεξάρτητα από όλες τις άλλες εικόνες. Κάθε GOP ξεκινά (προκειμένου για αποκωδικοποίηση) με αυτόν τον τύπο της εικόνας .
- P picture ή P frame (predictive coded picture) περιέχει motion-compensated difference πληροφορίες συσχετισμένες με προηγούμενες αποκωδικοποιημένες εικόνες, και κάθε predictively coded region μέσα στη P picture αναφέρεται σε μια μόνο προηγούμενη αποκωδικοποιημένη εικόνα ως reference picture.
- B picture ή B frame (bipredictive coded picture) περιέχει motion-compensated difference πληροφορίες συσχετισμένες με προηγούμενες αποκωδικοποιημένες εικόνες, και κάθε predictively coded region μέσα στη B picture αναφέρεται σε μια ή δυο προηγούμενες αποκωδικοποιημένες εικόνες ως references.

Η GOP structure-δομή αναφέρεται συχνά από δύο αριθμούς, για παράδειγμα, $M = 3$, $N = 12$. Ο πρώτος αριθμός λέει την απόσταση μεταξύ δύο πλαισίων (I ή P). Η δεύτερη λέει την απόσταση ανάμεσα σε δύο πλήρεις εικόνες (I-frames): είναι το GOPsize.)

Για παράδειγμα $M = 3$, $N = 12$, η δομή GOP είναι IBBPBBPBBPBBBI.. Αντί της παραμέτρου M μπορεί να χρησιμοποιηθεί η μέγιστη μέτρηση (maximal count) των B-frames μεταξύ δύο διαδοχικών anchor frames.

Για παράδειγμα, σε ένα μοτίβο αλληλουχίας με IBBBBPBBBBPBBBBIBBBBPB, το μέγεθος GOP είναι ίσο με 16 (μήκος μεταξύ δύο πλαισίων I) και την απόσταση μεταξύ των δύο πλαισίων άγκυρας (τιμή M) είναι 4 (μήκος μεταξύ των πλαισίων I και P ή το μήκος μεταξύ δύο διαδοχικών πλαισίων P).

4.6 Το μέλλον

Η υιοθέτηση του H.265 είναι αργή και παρά το γεγονός πως φαίνεται πολλά υποσχόμενο πρότυπο, δεν υπάρχει καμία εγγύηση ότι θα αναλάβει πλήρως τον κόσμο των βίντεο με τον ίδιο τρόπο που το H.264 έχει.

VP9

Ο VP9 είναι ένα ανοικτό, χωρίς πνευματικά δικαιώματα πρότυπο που αναπτύσσεται από την Google και είναι διάδοχος του VP8. Ο VP9 είχε αρχικά ονομαστεί Next Gen Open Video(NGOV) καθώς και VP-Next. Η ανάπτυξη του VP9 ξεκίνησε το τρίτο τρίμηνο του 2011. Ένας από τους στόχους του VP9 είναι να μειώσει το bit rate κατά 50% σε σχέση με το VP8 για την ίδια ποιότητα βίντεο.

Έχει αρκετές σχεδιαστικές βελτιώσεις σε σχέση με τον VP8 μερικές από τις οποίες είναι η χρήση superblocks μεγέθους 32 X 32 pixels (με πιθανή και την υποστήριξη 64 X 64) και για τα οποία θα χρησιμοποιείται δομή κωδικοποίησης τετραδικού δέντρου.

Άλλος στόχος είναι να βελτιωθεί σε τέτοιο σημείο όπου θα έχει καλύτερη αποδοτικότητα από τον HEVC.

3D HEVC Extension(επέκταση) για κωδικοποίηση 3D video

Οι πρόσφατες βελτιώσεις στην 3D τεχνολογία βίντεο έχουν οδηγήσει σε αυξανόμενο ενδιαφέρον για τα τρισδιάστατα βίντεο. Στα σύγχρονα σινεμά έχει αυξηθεί η προβολή 3D ταινιών. Επίσης, λόγω της ύπαρξης 3D TV τηλεοράσεων και Blu-ray συσκευών, 3D broadcast καναλιών και Blu-ray δίσκων έχει εισαχθεί το τρισδιάστατο video και στα σπίτια των καταναλωτών.

Οι Autostereoscopic οθόνες, δίνουν τρισδιάστατη εμπειρία χωρίς γυαλιά, συνεχώς βελτιώνονται και θεωρούνται πολλά υποσχόμενη τεχνολογία για τη μελλοντική τρισδιάστατη διασκέδαση στο σπίτι. Σε αντίθεση με τις κοινές οθόνες, οι Autostereoscopic απαιτούν όχι μόνο δυο, αλλά πλήθος από διαφορετικές όψεις για να παρέχουν την 3D εμπειρία.

Πολλά υποσχόμενη είναι η μετάδοση 3D βίντεο στο Multiview Video Depth(MVD) format. Σε αυτό το format, τυπικά μόνο μερικές όψεις κωδικοποιούνται, αλλά η κάθε μια χαρακτηρίζεται από κωδικοποιημένα δεδομένα βάθους, τα οποία αναπαριστούν τη βασική γεωμετρία του στιγμιότυπου του βίντεο.

Οι ερευνητικές ομάδες Image and Video Coding Group και 3D Coding Group ανέπτυξαν το 3D HEVC Extension(επέκταση) για κωδικοποίηση 3D video σε MDV format. Το 3D HEVC Extension(επέκταση) είναι επέκταση του σχεδιασμού του HEVC, το οποίο βελτιώνει τις δυνατότητες συμπίεσης για στιγμιότυπα βίντεο και δεδομένα βάθους, τα οποία με διάφορες τεχνικές μετατρέπονται σε 3D video bitstreams.

Συμπερασματικά, η ανάπτυξη του προτύπου βρίσκεται ακόμα σε εξέλιξη. Το γεγονός αυτό του προσδίδει μεγάλη δυναμική για το μέλλον. Η χρήση του HEVC αναμένεται από τα στελέχη της MPEG να επηρεάσει σε μεγάλο βαθμό την σχετική βιομηχανία, καθώς οι προβλέψεις θέλουν τα βίντεο να καταλαμβάνουν το 90% της συνολικής διαδικτυακής κίνησης το 2015.

ΠΑΡΑΡΤΗΜΑ

1. Παράδειγμα σύγκρισης H.265 και H.264 από το You Tube

a)



Cisco H.265 Demo: High quality video with great bandwidth efficiency

(Πηγή : www.youtube.com/watch?v=PZP75PEQo6E)

2. README TEXT USAGE-Software usage example

Contact: Woo-Jin Han, wjhan.han@samsung.com

1. Installation and compilation

1.1 Source tree

- Root
 - |--- bin
 - |--- build
 - |--- linux
 - |--- cfg
 - |--- cfp
 - |--- doc
 - |--- source
 - |--- App
 - |--- TAppDecoder
 - |--- TAppEncoder
 - |--- Lib
 - |--- TLibCommon
 - |--- TLibDecoder
 - |--- TLibEncoder
 - |--- TLibVideoIO

1.2 Windows using MS Visual Studio

- Workspaces of VC6 and VC2008 are included in Root/build directory

1.3 Linux

- Makefile is included in Root/build/linux directory

2. Encoder option

2.1 Parameters

- TAppEncoder.exe -c config.cfg [options]
- Options
 - help this help text
 - c configuration file name
 - i, --InputFile original YUV input file name
 - b, --BitstreamFile bitstream output file name
 - o, --ReconFile reconstructed YUV output file name
 - wdt, --SourceWidth Source picture width
 - hgt, --SourceHeight Source picture height
 - BitDepth
 - BitIncrement bit-depth increasement
 - pdx, --HorizontalPadding horizontal source padding size
 - pdy, --VerticalPadding vertical source padding size

--PAD automatic source padding of multiple of 16
 -fr, --FrameRate Frame rate
 -fs, --FrameSkip Number of frames to skip at start of input YUV
 -f, --FramesToBeEncoded number of frames to be encoded (default=all)
 --FrameToBeEncoded deprecated alias of FramesToBeEncoded
 --MaxCUWidth
 --MaxCUHeight
 -s, --MaxCUSize max CU size
 -h, --MaxPartitionDepth CU depth
 -t, --MaxTrSize max transform size
 -utd, --MaxTrDepth max transform depth
 -ltd, --MinTrDepth min transform depth
 --QuadtreeTUFlag
 --QuadtreeTULog2MaxSize
 --QuadtreeTULog2MinSize
 -ip, --IntraPeriod intra period in frames, (-1: only first frame)
 -g, --GOPSize GOP size of temporal structure
 -rg, --RateGOPSize GOP size of hierarchical QP assignment
 (-1: implies inherit GOPSize value)
 -r, --NumOfReference Number of reference (P)
 -rb0, --NumOfReferenceB_L0 Number of reference (B_L0)
 -rb1, --NumOfReferenceB_L1 Number of reference (B_L1)
 --HierarchicalCoding
 --LowDelayCoding low-delay mode
 --GPB generalized B instead of P in low-delay mode
 --QBO skip refers highest quality picture
 --NRF non-reference frame marking in last layer
 --BQP hier-P style QP assignment in low-delay mode
 -ldm recommended low-delay setting (with LDC),
 (0=slow sequence, 1=fast sequence)
 -int, --InterpFilterType Interpolation Filter:
 0: DCT-IF
 1: 4-tap MOMS
 2: 6-tap MOMS
 3: DIF
 4: SIFO
 --DIFTap number of interpolation filter taps (luma)
 --SPF
 --FastSearch 0:Full search 1:Diamond 2:PMVFAST
 -sr, --SearchRange motion search range
 --HadamardME hadamard ME for fractional-pel
 --ASR adaptive motion search range
 --AMVRES Adaptive motion resolution
 -v, --GRefMode additional reference for weighted prediction
 (w: scale+offset, o: offset)
 -q, --QP Qp value, if value is float, QP is switched

once during encoding

- dqr, --DeltaQpRD max dQp offset for slice
- d, --MaxDeltaQP max dQp offset for block
- m, --dQPFile dQP file name
- RDOQ
- tq0, --TemporalLayerQPOffset_L0
 QP offset of temporal layer 0
- tq1, --TemporalLayerQPOffset_L1
 QP offset of temporal layer 1
- tq2, --TemporalLayerQPOffset_L2
 QP offset of temporal layer 2
- tq3, --TemporalLayerQPOffset_L3
 QP offset of temporal layer 3
- sym, --SymbolMode symbol mode (0=VLC, 1=SBAC)
- SBACRD SBAC based RD estimation
- MultiCodewordThreshold
- MaxPIPEBufferDelay
- BalancedCPUs
- LoopFilterDisable
- LoopFilterAlphaCOffset
- LoopFilterBetaOffset
- CIP combined intra prediction
- AIS adaptive intra smoothing
- MRG merging of motion partitions
- IMP interleaved motion vector predictor
- ROT
- ALF Adaptive Loop Filter
- AMP Asymmetric motion partition
- EdgePredictionEnable Enable edge based prediction for intra
- EdgeDetectionThreshold Threshold for edge detection of edge based
 prediction
- FEN fast encoder setting
- 1 turn option <name> on
- 0 turn option <name> off
- <name> = ALF - adaptive loop filter
- IBD - bit-depth increasement
- GPB - generalized B instead of P in low-delay mode
- HAD - hadamard ME for fractional-pel
- SRD - SBAC based RD estimation
- RDQ - RDOQ
- LDC - low-delay mode
- NRF - non-reference frame marking in last layer
- BQP - hier-P style QP assignment in low-delay mode
- PAD - automatic source padding of multiple of 16
- QBO - skip refers highest quality picture
- ASR - adaptive motion search range

FEN - fast encoder setting
AIS - adaptive intra smoothing
MRG - merging of motion partitions
IMP - interleaved motion vector predictor
AMVRES - Adaptive motion resolution

2.2 Config file (example)

```
#===== File I/O =====
InputFile      : d:\test\origcfp\RaceHorses_416x240_30.yuv
BitstreamFile  : RaceHorses.bin
ReconFile      : RaceHorses_enc.yuv
FrameRate    : 30      # Frame Rate per second
FrameSkip      : 0      # Number of frames to be skipped in input
SourceWidth    : 416    # Input frame width
SourceHeight   : 240    # Input frame height
FrameToBeEncoded : 9      # Number of frames to be coded

#===== Coding Structure =====
IntraPeriod    : 32     # Period of I-Frame ( -1 = only first)
GOPSize     : 8       # GOP Size (number of B slice = GOPSize-1)
NumOfReference : 1     # Number of reference frames
NumOfReferenceB_L0 : 1   # Number of reference frames for L0 for B-
slices
NumOfReferenceB_L1 : 1   # Number of reference frames for L1 for B-
slices
QP         : 32     # Quantization parameter(0-51)

#===== New Structure =====
MaxCUWidth     : 128    # Maximum Coding Unit size in width
MaxCUHeight    : 128    # Maximum Coding Unit size in Height
MaxPartitionDepth : 5    # Maximum partition depth. ( minimum width
= MaxWidth >> (MaxPartitionDepth-1) )

#===== B Slice =====
HierarchicalCoding : 1     # B hierarchical coding ON/OFF (if OFF, no
reference B is used)

#===== Entropy Coding =====
SymbolMode      : 1     # CAVLC: 0, SBAC: 1, only 1 supported, CAVLC
implementation is not completed

#===== Loop/Deblock Filter =====
LoopFilterDisable : 0     # Disable loop filter in slice header (0=Filter,
1=No Filter)
LoopFilterAlphaC0Offset : 0 # Range: -26 ~ 26
```


LoopFilterBetaOffset : 0 # Range: -26 ~ 26

#===== Motion search =====

FastSearch : 1 # 0:Full search 1:Diamond 2:PMVFAST(not supported)

SearchRange : 128 # (0: Search range is a Full frame)

MaxDeltaQP : 0 # Absolute delta QP (1:default)

2.3 Typical example

Example 1) TAppEncoder.exe -c test.cfg -q 32 -g 8 -f 9 -s 64 -h 4
(Hier-B) -> QP 32, hierarchical-B GOP 8, 9 frames, 64x64-8x8 CU (~4x4 PU)

Example 2) TAppEncoder.exe -c test.cfg -q 32 -g 4 -f 9 -s 64 -h 4 --LDC=1
(Hier-P) -> QP 32, hierarchical-P GOP 4, 9 frames, 64x64-8x8 CU (~4x4 PU)

Example 3) TAppEncoder.exe -c test.cfg -q 32 -g 1 -f 9 -s 64 -h 4 --LDC=1
(IPPP) -> QP 32, hierarchical-P GOP 4, 9 frames, 64x64-8x8 CU (~4x4 PU)

3. Decoder option

3.1 Parameters

- TAppDecoder.exe -b test.bin -o test.yuv
. Decode test.bin and make test.yuv as the reconstructed YUV
- TAppDecoder.exe -b test.bin
. Decode test.bin but YUV writing is skipped

4. Contact point

=====
Woo-Jin Han, Principal Engineer
(wjhan.han@samsung.com)

M/M Platform Lab

Digital Media & Communications R&D Center
Digital Media & Communications Business
Samsung Electronics Co. Ltd.

Korea phone)
+82-31-279-8831 (office), +82-10-3329-6393 (cellular)

3. PACKETLOSS SIMULATOR HEVC

```
// PacketLossSimulator HEVC
#include <stdio.h>
#include <stdlib.h>

#define MAXPACKETSIZE 50000

int looseit(FILE *err)
{
    int c;
    do {
        c = getc (err);
        if (c == EOF) {
            fseek(err, 0, SEEK_SET);
        }
        else if (c == '0') return 1;
        else return 0;
    } while (1);
}

int main(int ac, char *av[])
{
    FILE *in, *out, *err0=NULL;
    unsigned char buf[MAXPACKETSIZE];
    unsigned char nal_type;
    int i;
    int packet_number=0;

    if (ac < 4) {
        printf ("Usage: %s <inpackets> <outpackets> <pattern error-file> \n", av[0]);
        exit (-1);
    }

    if (NULL == (in = fopen(av[1],"rb"))) {
        perror("read input open");
        exit(1);
    }

    if (NULL == (out = fopen(av[2],"wb"))) {
        perror("write output open");
```

```

exit(2);
}

if (NULL == (err0 = fopen (av[3], "rb"))) {
perror ("read err0 open");
exit (3);
}

buf[0]=fgetc(in);
buf[1]=fgetc(in);
buf[2]=fgetc(in);
buf[3]=fgetc(in);
buf[4] = fgetc(in);
nal_type = buf[4] & 0x0F;

for(i=5; ; i++)
{
buf[i] = fgetc(in);
if(feof(in)==0)
{
if(buf[i-3]==0 && buf[i-2]==0 && buf[i-1]==0 && buf[i]==1)
{
fwrite(buf, sizeof(unsigned char), i-3, out);
printf("Packet Number = %2d Packet Size = %d (%d)\n", packet_number, i-3, nal_type);
packet_number++;
fflush(out);
buf[0]=buf[i-3];
buf[1]=buf[i-2];
buf[2]=buf[i-1];
buf[3]=buf[i];
break;
}
}
}
while(feof(in)==0)
{
for(i=4; ;i++)
{
buf[i] = fgetc(in);
nal_type = buf[4] & 0x1F;
if(feof(in)==0)
{
if(buf[i-3]==0 && buf[i-2]==0 && buf[i-1]==0 && buf[i]==1 )
{
if(nal_type==1 || nal_type==20)
{

```

```

if(err0 != NULL && looseit(err0))
printf("Packet Lost! (%d)\n", nal_type);
else
{
fwrite(buf, sizeof(unsigned char), i-3, out);
printf("Packet Number = %2d Packet Size = %d (%d)\n", packet_number, i-3, nal_type);
fflush(out);
}
}
else
{
fwrite(buf, sizeof(unsigned char), i-3, out);
printf("Packet Number = %2d Packet Size = %d (%d)\n", packet_number, i-3, nal_type);
fflush(out);
}


packet_number++;
buf[0]=buf[i-3];
buf[1]=buf[i-2];
buf[2]=buf[i-1];
buf[3]=buf[i];
break;
}
}
else
{
fwrite(buf, sizeof(unsigned char), i, out);
printf("Packet Number = %2d Packet Size = %d (%d)\n", packet_number, i, nal_type);
fflush(out);
break;
}
}
}

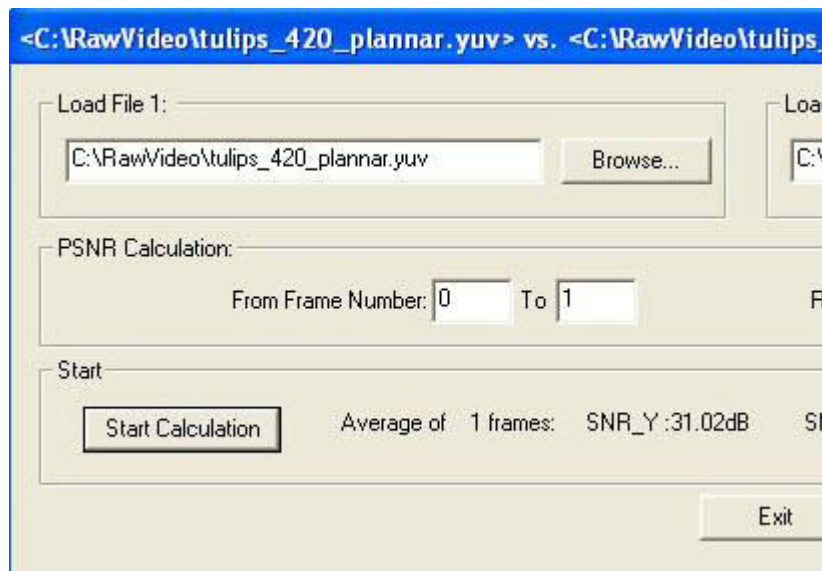
fclose(in);
fclose(out);
if ( err0)
fclose(err0);
return 0;
}

```

4. Yuv Analyzer

Στο ακόλουθο λίνκ http://www.sunrayimage.com/yuvanalyzer_psnr.html έχουμε τη συνάρτηση (function) για υπολογισμό του PSNR μεταξύ δυο YUV αρχείων, το αρχικό (original) video data και το επεξεργασμένο (processed) YUV video data. Τα δυο YUV αρχεία πρέπει να έχουν το ίδιο YUV format, και την ίδια ανάλυση (resolution).

Στο main menu, διαλέγουμε "YUV Analyzer", μετά επιλέγουμε "PSNR", ή κλικάρουμε στο  και θα προκύψει το παρακάτω παράθυρο.



5.

3D HEVC Extension

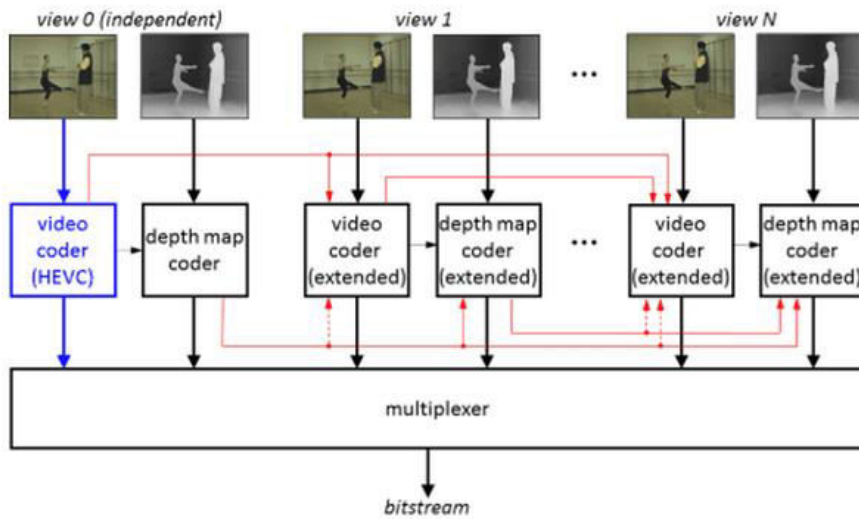


Figure 1: Simplified block diagram of the HEVC extension for 3d video coding.

Πηγή: <http://www.hhi.fraunhofer.de/en/fields-of-competence/image-processing/research-groups/3d-coding/3d-high-efficiency-video-coding.html>

ΒΙΒΛΙΟΓΡΑΦΙΑ

[01] David Gómez-Barquero, "An Overview of the Next Generation High Efficiency Video Coding (HEVC)", CRC Press, March 7, 2013.

[02] Andrew S. Tanenbaum, David J. Wetherall, "Computer Networks", 5th Edition, Pearson, 2011.

[03] P. H. Bardell, W. H. McAnney, and J. Savir, "Built-In Test for VLSI: Pseudorandom Techniques", John Wiley & Sons, New York, 1987.

[04] K.Cabeen, P.Gent, "Image Compression and the Discrete Cosine Transform", College of Redwoods, U.S.A, 2012.

[05] Γ. Σύρκος, "Ψηφιακή Επεξεργασία Σήματος", Εκδόσεις Παπασωτηρίου, Αθήνα, 2007.

[06] David Salomon, "Data Compression: The Complete Reference (4ed.)", p. 281. ISBN 978-1846286025, Springer, 2007.

[07] Stefan Winkler, "Digital Video Quality", ISBN 0-470-02404-6, Wiley, March 2005.

APOPA

[01] Gerhard Hablinger, Oliver Hohlfeld, "The Gilbert-Elliott Model For Packet Loss In Real Time Services On The Internet", T-Systems, Deutsche-Telecom-Allee 7, Germany.

[02] Philippe Hanhart, Martin Rerabek, Francesca De Simone, and Touradj Ebrahimi, "Subjective Quality Evaluation of the Upcoming HEVC Video Compression Standard", Ecole Polytechnique Federale De Lausanne (EPFL), Lausanne, Switzerland.

[03] J.Nightgale, Q.Wang and C.Grecos, "HEVCStream: A Framework for Streaming and Evaluating of High Efficiency Video Coding Content In Loss-prone Networks", IEEE Trans. On Consumer Electronics, May 2012.

[04] T.Weigand et al. , "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. On Circuits and Systems on Video Technology, July 2003.

[05] Kenneth Anderson, Arild Fuldseth, "Low Complexity Video Coding And The Emerging HEVC", Picture Coding Symposium(PCS), 28th Picture Coding Symposium, IEEE, Nagoya, Japan ,2010.

[06] G.J. Sullivan; J.-R. Ohm; W.-J. Han; T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Transactions on Circuits and Systems for Video Technology, 2012.

[07] R. Zhang, S.L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience", IEEE J. Select. Areas Commun. Vol. 18, NO. 6, pp. 966 -976, 2000.

[08] Huynh-Thu, Q.; Ghanbari, M., "Scope of validity of PSNR in image/video quality assessment", Electronics Letters 44 (13): 800. doi:10.1049/el:20080522. edit, 2008.

[09] Minoru Etoh, "Advances in Wireless Video Delivery", Proceedings of the IEEE, Vol. 93, NO. 1, January 2005.

[10] F. Bossen, "Common test conditions and software reference configurations", 9th Meeting of the JCT-VC, Geneva (Switzerland), April 2012.

ΔΙΑΔΙΚΤΥΑΚΕΣ ΠΗΓΕΣ

[01] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard".

Διαθέσιμο στο http://www.img.lx.it.pt/~fp/cav/Additional_material/HEVC-Overview.pdf.

[02] Jan Ozer, "The Future of HEVC: It's Coming, but with Plenty of Questions".

Διαθέσιμο στο <http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/The-Future-of-HEVC-Its-Coming-but-with-Plenty-of-Questions-89010.aspx>.

[03] Jens-Rainer Ohm, Gary Sullivan, "MPEG High Efficiency Video Coding (HEVC)".

Διαθέσιμο στο <http://mpeg.chiariglione.org/standards/mpeg-h/high-efficiency-video-coding>.

[04] Στάθης Ασπιώτη, "4K Μύθοι και Πραγματικότητα - HEVC".

Διαθέσιμο στο <http://www.tvwars.gr/4k-hevc/>.

[05] Wikipedia - jpeg & digital videos (2013), "An Introduction to MPEG Video Compression"; 2010. Διαθέσιμο στο <http://en.wikipedia.org/wiki/JPEG>

John Wiseman.

[06] Πηγή: http://www.john-wiseman.com/technical/MPEG_tutorial.htm.

[07] Daniel Frederick Finchelstein, "Low-Power Techniques for Video

Decoding", MIT, June 2009. Διαθέσιμο στο <http://dSPACE.mit.edu/bitstream/handle/1721.1/52794/547232263.pdf?sequence=1>.

[08] Πηγή : <http://www.webopedia.com/TERM/F/fps.html>

[09] Πηγή : http://www.nikon-sys.co.jp/products/index_2_8a_en.htm#04

[10] Πηγή : http://en.wikipedia.org/wiki/Motion_compensation

[11] Πηγή: <ftp://ftp.cs.wpi.edu/pub/techreports/pdf/06-02.pdf>

[12] Πηγή:

www.academia.edu/2676387/The_Use_of_H_264_AVC_and_the_Emerging_High_Efficiency_Video_Coding_HEVC_Standard_for_Developing_Wireless_Ultrasound_Video_Telemedicine_Systems