



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ

*Παράλληλες τεχνικές κωδικοποίησης video με
χρήση slices*

Σπύρος Κρουστάλης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων
Λουκόπουλος Αθανάσιος

Λαμία, 2019



UNIVERSITY OF THESSALY

SCHOOL OF SCIENCE

INFORMATICS AND COMPUTATIONAL BIOMEDICINE

Video encoding parallelization using slices

Spyros Kroustalis

Master thesis

**Supervisor
Loukopoulos Athanasios**

Lamia, 2019



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ
ΚΑΤΕΥΘΥΝΣΗ:**

«ΥΠΟΛΟΓΙΣΤΙΚΗ ΙΑΤΡΙΚΗ ΚΑΙ ΒΙΟΛΟΓΙΑ»

Παράλληλες τεχνικές κωδικοποίησης video με χρήση slices

Σπύρος Κρουστάλης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Επιβλέπων
Λουκόπουλος Αθανάσιος**

Λαμία, 2019

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο «Παράλληλες τεχνικές κωδικοποίησης video με χρήση slices» αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο ΔΗΛΩΝ

20/6/2019

Παράλληλες τεχνικές κωδικοποίησης video με χρήση slices

Κρουστάλης Σπύρος

Τριμελής Επιτροπή:

Δρ. Λουκόπουλος Αθανάσιος (επιβλέπων)

Δρ. Αναγνωστόπουλος Ιωάννης

Δρ. Σταμούλης Γεώργιος

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη	1
Abstract	2
Πρόλογος	3
Κεφάλαιο 1: Επισκόπηση προτύπων κωδικοποίησης video	4
1.1 Γενική Επισκόπηση.....	4
1.1.1 Chroma subsampling	7
1.1.2 Τροποποίηση του ρυθμού των καρέ	7
1.2 H.264/MPEG-4 AVC.....	8
1.2.1 Ιστορία.....	10
1.2.2 Εφαρμογές.....	11
1.2.3 Χαρακτηριστικά	11
1.2.4 Προφίλ.....	15
1.3 High Efficiency Video Coding (HEVC).....	16
1.3.1 Ιστορία.....	18
1.3.2 Αποτελεσματικότητα της κωδικοποίησης.....	19
1.3.3 Χαρακτηριστικά.....	20
1.3.4 Εργαλεία κωδικοποίησης.....	22
1.3.5 Προφίλ.....	27
Κεφάλαιο 2: Αλγόριθμοι για Παραλληλοποίηση HEVC με χρήση slices	28
2.1 Στόχοι	28
2.2 Συναφής έρευνα στην παράλληλη κωδικοποίηση βίντεο	29
2.3 Συναφής έρευνα στην παραλληλοποίηση σε επίπεδο slices.....	30

2.4 Αλγόριθμοι βιβλιογραφίας.....	32
2.5 Προτεινόμενοι αλγόριθμοι που δε βασίζονται στο GOP	33
2.6 Προτεινόμενοι αλγόριθμοι βασισμένοι στο GOP	35
Κεφάλαιο 3: Πειράματα	38
3.1 Πλαίσιο διεξαγωγής πειραμάτων	38
3.2 Αξιολόγηση της απόκλισης εξισορρόπησης φόρτου	45
3.3 Αξιολόγηση υπολογιστικού χρόνου	46
Κεφάλαιο 4: Συμπεράσματα.....	53
Αναφορές.....	54

Περίληψη

Στην παρούσα διπλωματική εργασία γίνεται αναφορά στα πρότυπα κωδικοποίησης video HEVC και H.264. Αρχικά, γίνεται μία εισαγωγή στις έννοιες της κωδικοποίησης βίντεο, ενώ στη συνέχεια παρουσιάζονται τα κύρια τεχνικά χαρακτηριστικά των προαναφερθέντων προτύπων. Τέλος, παρουσιάζεται η αποτελεσματικότητα της παραλληλοποίησης σε επίπεδο slices για το πρότυπο κωδικοποίησης HEVC και πραγματοποιείται σύγκριση τριών εναλλακτικών αλγορίθμων. Η πειραματική αξιολόγηση περιλαμβάνει συνήθεις δοκιμαστικές ακολουθίες δύο σεναρίων (Low Delay και Random Access).

Λέξεις Κλειδιά: Συμπίεση βίντεο, κωδικοποίηση βίντεο, παραλληλοποίηση, slices, HEVC, H.264/AVC.

Abstract

The aim of the present thesis is to study parallel video coding techniques based on slices in HEVC standard. First, a review of the video coding process in H.264/AVC and HEVC standards is given. Then the thesis presents slice based parallel techniques for HEVC. Three algorithms are given and evaluated using common test sequences under two different scenarios (LowDelay and RandomAccess).

Keywords: video compression, video coding, parallelism, slices, HEVC, H.264/AVC.

Πρόλογος

Η παρούσα διπλωματική εργασία με τίτλο: «Παράλληλες τεχνικές κωδικοποίησης video με χρήση slices» εκπονήθηκε στο πλαίσιο του διατμηματικού προγράμματος μεταπτυχιακών σπουδών: «Πληροφορική και Υπολογιστική Βιοϊατρική» του Πανεπιστημίου Θεσσαλίας.

Στην εργασία γίνεται μελέτη τεχνικών παραλληλοποίησης της διαδικασίας συμπίεσης βίντεο με έμφαση στο HEVC. Πιο συγκεκριμένα ερευνάται η δυνατότητα παραλληλοποίησης χρησιμοποιώντας ως «μονάδες» εργασίας τα slices στα οποία μπορεί να χωριστεί η εικόνα του βίντεο. Η εξισορρόπηση του φόρτου που προκύπτει στις μονάδες επεξεργασίας είναι σημαντικός παράγοντας στη μείωση του απαιτούμενου χρόνου συμπίεσης. Ως εκ τούτου μελετώνται τεχνικές πρόβλεψης του φόρτου και συνακόλουθης αλλαγής του μεγέθους των slices για να επιτευχθεί ο προαναφερόμενος στόχος. Σημειώνουμε ότι οι αλγόριθμοι που προτείνονται στην παρούσα διπλωματική εργασία αποτελούν επεκτάσεις των τεχνικών που παρουσιάστηκαν στην [15].

Η οργάνωση της εργασίας έχει ως εξής:

Στο Κεφάλαιο 1 γίνεται παρουσίαση των βασικών χαρακτηριστικών των προτύπων συμπίεσης H.264 και HEVC.

Στο Κεφάλαιο 2 παρουσιάζονται οι αλγοριθμικές τεχνικές για παραλληλία σε επίπεδο slices.

Το Κεφάλαιο 3 περιλαμβάνει τα πειραματικά αποτελέσματα.

Τέλος, το Κεφάλαιο 4 παρουσιάζει τα συμπεράσματα.

Κεφάλαιο 1: Επισκόπηση προτύπων κωδικοποίησης video

1.1 Γενική Επισκόπηση

Η κωδικοποίηση βίντεο είναι η διαδικασία συμπίεσης και ενδεχομένως αλλαγής της μορφής του περιεχομένου βίντεο, μερικές φορές ακόμη και η αλλαγή αναλογικής πηγής σε ψηφιακή. Όσον αφορά στη συμπίεση, ο στόχος είναι να καταναλώνει λιγότερο χώρο. Αυτό συμβαίνει επειδή είναι μια διαδικασία απώλειας που πετάει πληροφορίες που σχετίζονται με το βίντεο. Με την αποσυμπίεση για αναπαραγωγή, δημιουργείται μια προσέγγιση του πρωτοτύπου. Όσο περισσότερη συμπίεση εφαρμόζεται, τόσο περισσότερα δεδομένα απορρίπτονται και τόσο χειρότερη φαίνεται η προσέγγιση σε σχέση με την αρχική.

Υπάρχουν δύο λόγοι για τους οποίους η κωδικοποίηση βίντεο είναι σημαντική. Ο πρώτος λόγος αφορά στη ροή δεδομένων, διευκολύνοντας τη μετάδοση βίντεο μέσω του Διαδικτύου. Αυτό συμβαίνει επειδή η συμπίεση μειώνει το απαιτούμενο bandwidth, ενώ ταυτόχρονα δίνει μια ποιοτική εμπειρία. Χωρίς συμπίεση, το ακατέργαστο περιεχόμενο βίντεο θα απέκλειε πολλούς από το να είναι σε θέση μετάδοσης περιεχομένου μέσω του Διαδικτύου λόγω της ταχύτητας σύνδεσης που δεν θα επαρκούσε. Η πιο σημαντική μετρική όσον αφορά στη διαδικασία μετάδοσης βίντεο είναι ο ρυθμός μετάδοσης δεδομένων (bitrate) ή αλλιώς το ποσοστό δεδομένων ανά δευτερόλεπτο στο βίντεο. Σε εφαρμογές streaming, ο ρυθμός μετάδοσης καθορίζει εάν οι τελικοί χρήστες μπορούν εύκολα να παρακολουθήσουν το περιεχόμενο ή αν θα κολλήσουν στο επίπεδο φόρτωσης (buffering) του βίντεο.

Ο δεύτερος λόγος για την κωδικοποίηση βίντεο είναι η συμβατότητα. Στην πραγματικότητα, μερικές φορές το περιεχόμενο είναι ήδη συμπιεσμένο σε ένα κατάλληλο μέγεθος, αλλά χρειάζεται και άλλη κωδικοποίηση για την συμβατότητα, αν και αυτό συχνά περιγράφεται με μεγαλύτερη ακρίβεια ως διακωδικοποίηση. Η συμβατότητα μπορεί να αφορά συγκεκριμένες υπηρεσίες ή προγράμματα, τα οποία απαιτούν συγκεκριμένες προδιαγραφές κωδικοποίησης.

Η διαδικασία της κωδικοποίησης βίντεο καθορίζεται από κωδικοποιητές βίντεο ή αλλιώς πρότυπα συμπίεσης βίντεο.

Οι κωδικοποιητές βίντεο είναι πρότυπα συμπίεσης βίντεο που πραγματοποιούνται μέσω εφαρμογών λογισμικού ή υλικού. Κάθε κωδικοποιητής αποτελείται από έναν κωδικοποιητή, για την συμπίεση του βίντεο και έναν αποκωδικοποιητή, για να δημιουργήσει το βίντεο αναπαραγωγής. Η έννοια του ονόματος codec (πρότυπο κωδικοποίησης) προέρχεται στην πραγματικότητα από τη συγχώνευση αυτών των δύο λέξεων σε μία μόνο λέξη: enCOder (κωδικοποιητής) και DECoder (αποκωδικοποιητής).

Παραδείγματα προτύπων κωδικοποιητών βίντεο περιλαμβάνουν H.264, VP8, RV40 και πολλά άλλα πρότυπα ή νεότερες εκδόσεις αυτών των κωδικοποιητών όπως VP9.

Αυτά τα πρότυπα κωδικοποιητών χρησιμοποιούν τεχνικές συμπίεσης για να μειώσουν το μέγεθος του περιεχομένου βίντεο χωρίς να επηρεαστεί σημαντικά η ποιότητά του. Το τελικό αποτέλεσμα ορισμένων τεχνικών είναι πιο αισθητό στον τελικό θεατή από άλλες.

Μια κοινή τεχνική συμπίεσης είναι η αλλαγή μεγέθους ή η μείωση της ανάλυσης. Αυτό συμβαίνει επειδή όσο μεγαλύτερη είναι η ανάλυση ενός βίντεο, τόσο περισσότερες πληροφορίες περιλαμβάνονται σε κάθε καρέ. Για παράδειγμα, ένα βίντεο 1280×720 έχει τη δυνατότητα για 921.600 pixels σε κάθε καρέ. Αντίθετα, ένα βίντεο 640×360 έχει τις δυνατότητες για 230.400 pixels ανά καρέ.

Επομένως, μια μέθοδος για τη μείωση της ποσότητας των δεδομένων είναι να «συρρικνωθεί» το μέγεθος της εικόνας. Αυτό θα δημιουργήσει λιγότερα pixels, μειώνοντας το επίπεδο λεπτομέρειας στην εικόνα προς όφελος της μείωσης της ποσότητας των απαραίτητων πληροφοριών.

Το αποτέλεσμα αυτής της αλλαγής μεγέθους μπορεί να είναι η εμφάνιση του «pixilation». Αυτό μερικές φορές αποκαλείται macroblocking, αν και συνήθως είναι πιο συχνά γνωστό ως pixilation. Γενικά, αυτό είναι ένα φαινόμενο όπου τμήματα μιας εικόνας φαίνονται μπλοκαρισμένα. Αυτό μπορεί να είναι ένας συνδυασμός μιας εικόνας χαμηλής ανάλυσης και διαπλαισιακής (interframe) επεξεργασίας, όπου οι λεπτομέρειες στο βίντεο στην πραγματικότητα αλλάζουν αλλά μέσω της διαπλαισιακής επεξεργασίας, περιοχές του βίντεο διατηρούνται. Ενώ αυτό μειώνει την ποσότητα δεδομένων που απαιτεί το βίντεο, παράλληλα μειώνει την ποιότητά του.

Σημειώστε ότι αυτή η διαδικασία αλλαγής μεγέθους αναφέρεται μερικές φορές και ως κλιμάκωση. Ωστόσο, η κλιμάκωση παίρνει μερικές φορές μια μη συμπίεσμένη έννοια. Για παράδειγμα, μερικές φορές χρησιμοποιείται για να περιγράψει τη λήψη της ίδιας εικόνας μεγέθους και την παρουσίασή της με μικρότερο τρόπο. Σε αυτά τα

σενάρια, ο αριθμός των εικονοστοιχείων δεν αλλάζει και έτσι δεν εφαρμόζεται συμπίεση.

Μια βασική τεχνική συμπίεσης βίντεο είναι η διαπλαισιακή (Interframe) η οποία είναι μια διαδικασία που μειώνει τις «περιττές» πληροφορίες από πλαίσιο σε πλαίσιο. Για παράδειγμα, ένα βίντεο με 30 FPS (καρέ ανά δευτερόλεπτο) σημαίνει ότι το ένα δευτερόλεπτο του βίντεο ισούται με 30 καρέ ή ακίνητες εικόνες. Όταν παίζονται μαζί, προσομοιώνουν την κίνηση. Ωστόσο, οι πιθανότητες είναι στοιχεία από καρέ σε καρέ εντός αυτών των 30 καρέ να παραμείνουν ουσιαστικά τα ίδια. Συνειδητοποιώντας αυτό, αναπτύχθηκε στη βιβλιογραφία η τεχνική interframe πρόβλεψης, για την απομάκρυνση των πλεοναζόντων δεδομένων, ουσιαστικά μειώνοντας τα δεδομένα που θα χρησιμοποιηθούν για να μεταδώσουν ότι ένα στοιχείο δεν έχει αλλάξει ανάμεσα στα καρέ. Για την εκτέλεση της τεχνικής, χρησιμοποιούνται τρεις διαφορετικοί τύποι καρέ για να επιτευχθεί αυτό: I-frames, P-frames και B-frames.

I-frame

Επίσης γνωστό ως keyframe, είναι ένα καρέ του βίντεο με την πλήρη εικόνα. Το πόσο συχνά εμφανίζεται το I-frame εξαρτάται από τον τρόπο κωδικοποίησής του. Όσο πιο συχνά δημιουργείται ένα I-frame, τόσο περισσότερο χώρο χρειάζεται. Ωστόσο, υπάρχουν οφέλη για την δημιουργία περιοδικών I-frames κάθε 2 δευτερόλεπτα (καλύτερη ποιότητα).

P-frame

Γνωστό ως καρέ πρόβλεψης (Predictive frame), είναι ένα καρέ δέλτα (delta frame) που περιέχει μέρος της εικόνας. Θα κάνει έλεγχο προς τα πίσω σε ένα I-frame ή άλλο P-frame για να δει αν ένα μέρος της εικόνας είναι το ίδιο. Αν ναι, αυτό το τμήμα θα αποκλειστεί για εξοικονόμηση χώρου.

B-frame

Ονομάζεται αμφίδρομο καρέ πρόβλεψης (Bi-directional frame) και είναι και αυτό ένα delta frame που περιέχει επίσης μέρος της εικόνας. Η διαφορά μεταξύ αυτού και ενός P-frame, είναι ότι μπορεί να κοιτάζει προς τα πίσω ή προς τα εμπρός για άλλα delta frames ή I-frame όταν επιλέγει ποιες λεπτομέρειες πρέπει να αφαιρεθούν καθώς υπάρχουν σε άλλο καρέ. Ως αποτέλεσμα, τα B-frames προσφέρουν βελτιωμένη συμπίεση χωρίς να μειώνουν την εμπειρία προβολής. Ωστόσο, απαιτούν υψηλότερο προφίλ κωδικοποίησης.

1.1.1 Chroma subsampling

Όπως μπορεί να είναι αναμενόμενο, όσο περισσότερες πληροφορίες χρωμάτων απαιτούνται για την αναπαραγωγή μιας εικόνας, τόσο περισσότερο χώρο καταλαμβάνει. Ως αποτέλεσμα, ένας τρόπος συμπίεσης των δεδομένων σε ένα βίντεο είναι να απορριφθούν ορισμένες από τις πληροφορίες χρώματος. Αυτή είναι μια διαδικασία που ονομάζεται chroma subsampling. Η κύρια ιδέα πίσω από αυτό είναι ότι το ανθρώπινο μάτι ανιχνεύει πολύ πιο εύκολα διαφορές στην φωτεινότητα από ότι πληροφορίες χρώματος. Ως εκ τούτου, γίνεται προσπάθεια να διατηρηθεί η φωτεινότητα, θυσιάζοντας την ποιότητα των χρωμάτων.

Αυτό επιτυγχάνεται με τη μετατροπή ενός φάσματος RGB (κόκκινο, πράσινο, μπλε) σε YCbCr. Με τον τρόπο αυτό, η φωτεινότητα Y χωρίζεται και μπορεί να παραδοθεί ασυμπίεστα. Οι πληροφορίες χρώματος περιέχονται στο CbCr, και μειώνονται για εξοικονόμηση χώρου. Ουσιαστικά το Y μπορεί να θεωρηθεί ως μια εκδοχή της εικόνας σε κλίμακα του γκρι. Αυτό επικαλύπτεται από την εκδοχή Cb, η οποία περιέχει κίτρινες και μπλε αποχρώσεις και την εκδοχή Cr, που περιέχει πράσινες και κόκκινες αποχρώσεις

Προσθέτοντας όλα τα παραπάνω στοιχεία μαζί έχουμε μια πλήρη, έγχρωμη εικόνα.

Η συμπίεση πραγματοποιείται χρησιμοποιώντας αναλογία χώρου 4:2:2 έναντι της μη συμπίεστης έκδοσης 4:4:4. Κάθε ένας από τους αριθμούς αντιστοιχεί με μια περιοχή του YCbCr. Επομένως, το 4:2:2 σημαίνει ότι η γκρι κλίμακα (Y) δεν συμπιέζεται, καθώς εξακολουθεί να είναι 4, ενώ το Cb και το Cr είναι το καθένα μειωμένο στο 2 ώστε να έχουν το μισό σε σχέση με την φωτεινότητα. Ως αποτέλεσμα, το απαιτούμενο bandwidth μειώνεται κατά 33% ($(4+2+2)/(4+4+4) = 66\%$ δεδομένα έναντι του αρχικού). Υπάρχουν και άλλες παραλλαγές, για παράδειγμα 4:1:1, οι οποίες μειώνουν ακόμα περισσότερο το bandwidth και 4:2:0 (Εναλλαγές μεταξύ Cb και Cr).

1.1.2 Τροποποίηση του ρυθμού των καρτέ

Μια άλλη μέθοδος για τη συμπίεση βίντεο είναι η μείωση της ποσότητας των καρτέ ανά δευτερόλεπτο. Αυτό μειώνει την ποσότητα δεδομένων που απαιτούνται σε ένα βίντεο, καθώς χρειάζονται λιγότερες πληροφορίες για τη μετάδοση κάθε

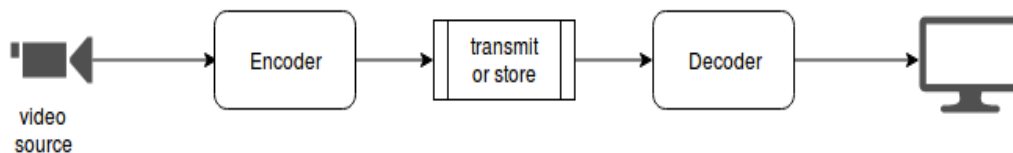
δευτερόλεπτο. Αυτό μπορεί να είναι λίγο πιο καταστροφικό από άλλες μεθόδους, καθώς η μείωση του ρυθμού των καρέ οδηγεί σε βίντεο χωρίς σωστή κίνηση.

Παρακάτω θα αναφερθούμε σε δύο, ίσως τα πιο γνωστά, πρότυπα κωδικοποίησης βίντεο το H.264/MPEG-4 AVC και το HEVC όπου θα γίνει ανάλυση και σύγκριση αυτών.

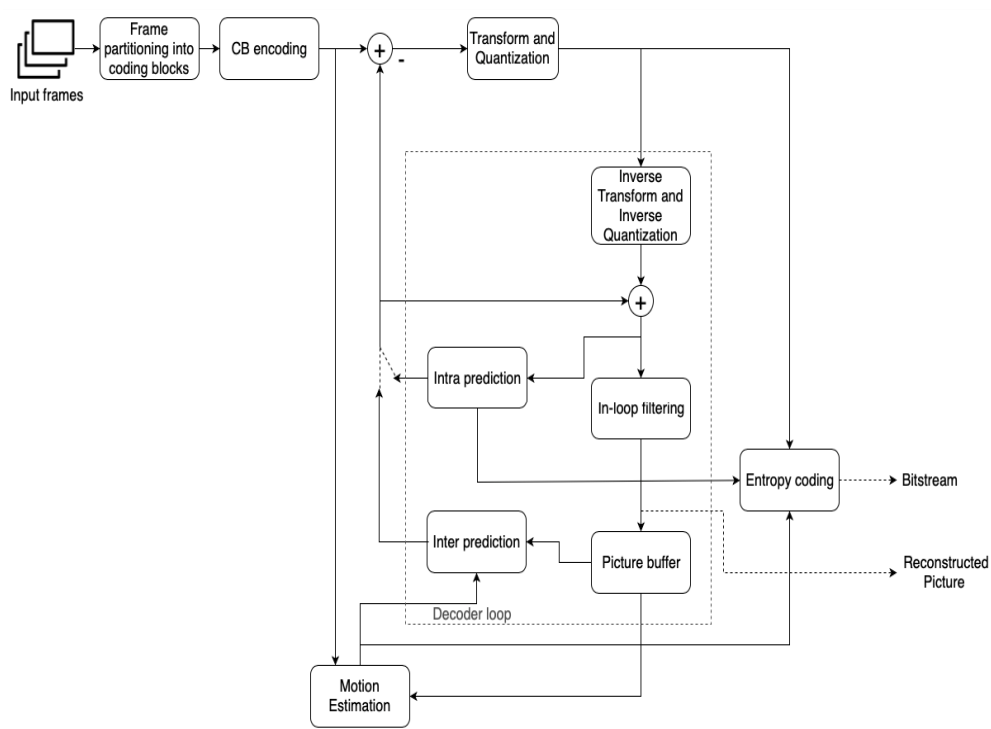
1.2 H.264/MPEG-4 AVC

Το H.264 [39] ή αλλιώς MPEG-4 Part 10, Advanced Video Coding (MPEG-4 AVC) είναι ένα πρότυπο συμπίεσης βίντεο το οποίο βασίζεται πάνω σε μια αλγοριθμική τεχνική η οποία έχει σκοπό να «προβλέψει» τα καρέ ενός βίντεο συνυπολογίζοντας την κίνηση της κάμερας ή/και των στοιχείων και αντικειμένων του βίντεο (block-oriented motion-compensation). Οι Εικόνες 1 και 2 παρουσιάζουν σχηματικά τη διαδικασία κωδικοποίησης και αποκωδικοποίησης και τη βασική ροή ενός κωδικοποιητή H.264.

Η πρόθεση για την δημιουργία του H.264 ήταν για ένα πρότυπο ικανό να παρέχει καλή ποιότητα βίντεο με πολύ μικρότερο ρυθμό δεδομένων (bit rate) σε σχέση με τα προηγούμενα πρότυπα (δηλαδή, στο μισό ή λιγότερο bit rate σε σχέση με MPEG-2, H.263), χωρίς να αυξάνεται πολύ η πολυπλοκότητα του σχεδιασμού ούτως ώστε να το καθιστά είτε μη πρακτικό είτε υπερβολικά δαπανηρό για την υλοποίησή του.



Εικόνα 1: Η διαδικασία κωδικοποίησης και αποκωδικοποίησης.



Εικόνα 2: Γενικό διάγραμμα ροής H.264/AVC encoder.

Ένας επιπλέον λόγος για την δημιουργία του, ήταν για ένα πρότυπο συμπίεσης αρκετά «ευέλικτο» ώστε να μπορεί να εφαρμοστεί σε μια μεγάλη ποικιλία εφαρμογών λογισμικού, σε ένα ευρύ φάσμα δικτύων και συστημάτων, συμπεριλαμβανομένων των χαμηλών και υψηλών bit rate, βίντεο υψηλής ευκρίνειας, αναμετάδοση, αποθήκευση DVD, πρωτόκολλα δικτύων RTP/IP (Real-time Transport Protocol, μεταφορά ήχου και βίντεο μέσω IP δικτύου) και τηλεφωνικά συστήματα πολυμέσων ITU-T (International Telecommunication Union-Telecommunication Standardization Sector). Το πρότυπο H.264 μπορεί να θεωρηθεί ως μια «οικογένεια προτύπων» η οποία αποτελείται από πολλά διαφορετικά προφίλ. Ένας συγκεκριμένος αποκωδικοποιητής (decoder) μπορεί να αποκωδικοποιήσει τουλάχιστον ένα, αλλά όχι απαραίτητα όλα τα προφίλ. Οι προδιαγραφές ενός αποκωδικοποιητή είναι αυτές που περιγράφουν ποια από τα προφίλ μπορεί να αποκωδικοποιήσει. Το H.264 τυπικά χρησιμοποιείται για κωδικοποίηση με συμπίεση απώλειας (lossy compression: μέθοδος κωδικοποίησης δεδομένων που χρησιμοποιεί ανακριβείς προσεγγίσεις και ενέχει απώλεια πληροφορίας), παρόλο που είναι επίσης δυνατό να δημιουργηθούν πραγματικά κωδικοποιημένα κομμάτια χωρίς

απώλειες μέσα σε μια κωδικοποιημένη εικόνα με συμπίεση απώλειας ή σπάνιες περιπτώσεις στις οποίες ολόκληρη η κωδικοποίηση είναι χωρίς απώλειες.

Το H.264 αναπτύχθηκε από την ITU-T Video Coding Experts Group (VCEG) μαζί με την ISO/IEC JTC1 Moving Pictures Experts Group (MPEG). Η προσπάθεια συνεργασίας αυτού του έργου είναι γνωστή και ως Joint Video Team (JVT). Τα πρότυπα ITU-T H.264 και ISO/IEC MPEG-4 AVC έχουν διατηρηθεί από κοινού έτσι ώστε να έχουν το ίδιο τεχνικό περιεχόμενο. Η πρώτη έκδοση αυτού του προτύπου ολοκληρώθηκε τον Μάιο του 2003 και από τότε έχουν προστεθεί πολλές επεκτάσεις αυξάνοντας τις δυνατότητες του σε νεότερες εκδόσεις. Το HEVC (High Efficiency Video Coding) ή αλλιώς H.265 και MPEG-H Part 2 είναι διάδοχος του H.264/MPEG-4 AVC και αναπτύχθηκε από τους ίδιους οργανισμούς ενώ παράλληλα εξακολουθούν και χρησιμοποιούνται παλιότερα πρότυπα συμπίεσης.

Το H.264 είναι ίσως το πιο γνωστό ως ένα από τα πρότυπα κωδικοποίησης βίντεο για δίσκους Blu-ray. Χρησιμοποιείται επίσης από ιστοσελίδες όπως το YouTube για την αναπαραγωγή βίντεο, από το iTunes Store, από εταιρείες ανάπτυξης διαδικτυακού λογισμικού όπως η Adobe (Adobe Flash Player) και η Microsoft (Microsoft Silverlight), καθώς για διάφορες HDTV εκπομπές μέσω επίγειων (ATSC, ISDB-T, DVB-T ή DVB-T2), καλωδιακά (DVB-C) και δορυφορικά (DVB-S και DVB-S2).

1.2.1 Ιστορία

Στις αρχές του 1998 η ομάδα VCEG (Video Coding Experts Group) δημοσίευσε μια πρόσκληση για υποβολή προτάσεων για ένα project με τίτλο H.26L, με σκοπό να διπλασιαστεί η αποτελεσματικότητα της κωδικοποίησης βίντεο (το οποίο σημαίνει τη μείωση των απαραίτητων bit rate, δεδομένου ενός επιπέδου πιστότητας) σε σύγκριση με οποιοδήποτε άλλο υπάρχων πρότυπο κωδικοποίησης βίντεο για μια ευρεία ποικιλία εφαρμογών. Πρόεδρος της ομάδας VCEG εκείνη την εποχή ήταν ο Gary Sullivan (Microsoft, πρώην PictureTel, Η.Π.Α.). Το πρώτο σχέδιο για το νέο αυτό πρότυπο κωδικοποίησης εγκρίθηκε τον Αύγουστο του 1999. Το 2000 ο Thomas Wiegand (Heinrich Hertz Institute, Γερμανία) έγινε συμπρόεδρος της VCEG.

Το Δεκέμβριο του 2001, η VCEG και η MPEG (Moving Pictures Experts Group) σχημάτισαν την JVT (Joint Video Team) με σκοπό την ολοκλήρωση του προτύπου H.264. Η τελική έγκριση έγινε τον Μάρτιο του 2003, ενώ εκείνη την περίοδο η JVT προήδρευε (έως και σήμερα) από τους Gary Sullivan, Thomas Wiegand και Ajay

Luthra (Motorola, Η.Π.Α.). Από τον Ιανουάριο του 2005 έως τον Νοέμβριο του 2007, η JVT εργάστηκε πάνω σε μια νέα επέκταση του H.264/AVC, περίοδος κατά την οποία η διαχείριση της JVT επεκτάθηκε με τον Jens-Rainer Ohm (Aachen University, Γερμανία).

1.2.2 Εφαρμογές

Το πρότυπο κωδικοποίησης βίντεο H.264 έχει ένα ευρύ φάσμα εφαρμογών (χρήσεων) και μπορεί και καλύπτει όλες τις μορφές του ψηφιακά συμπιεσμένου βίντεο από low bit-rate (χαμηλής ροής δεδομένων) Internet streaming εφαρμογές έως HDTV εκπομπές και Digital Cinema εφαρμογές με σχεδόν χωρίς απώλειες κωδικοποίηση. Με την χρήση του H.264 ως πρότυπο κωδικοποίησης έχει παρατηρηθεί έως 50% ή και παραπάνω εξοικονόμηση του bit-rate σε σχέση με το MPEG-2 Part 2. Για παράδειγμα, το H.264 έχει αναφερθεί ότι παρέχει την ίδια ποιότητα Δορυφορικής Ψηφιακής Τηλεόρασης (Digital Satellite TV) όπως οι τρέχουσες εφαρμογές MPEG-2 με λιγότερο από το μισό bit-rate (με τις τρέχουσες εφαρμογές MPEG-2 να δουλεύουν με περίπου 3.5 Mbit/s ενώ το H.264 1.5 με Mbit/s). Η Sony υποστηρίζει πως η λειτουργία εγγραφής AVC στα 9 Mbit/s έχει ισοδύναμη ποιότητα εικόνας με το HDV format (High Definition Video) το οποίο χρησιμοποιεί περίπου 18-25 Mbit/s.

Για να εξασφαλιστεί η συμβατότητα και η χωρίς προβλήματα εφαρμογή του H.264/AVC, πολλοί οργανισμοί έχουν τροποποιήσει ή το έχουν προσθέσει σε όλα τα πρότυπα που σχετίζονται με βίντεο, έτσι ώστε οι χρήστες αυτών να μπορούν να το χρησιμοποιούν. Τόσο ο δίσκος Blu-ray όσο και το πλέον όχι διαδεδομένο HD DVD format περιλαμβάνουν το πρότυπο H.264/AVC High Profile ως ένα από τα τρία υποχρεωτικά πρότυπα συμπίεσης βίντεο. Επίσης το DVB project (Digital Video Broadcast) ενέκρινε τη χρήση του H.264/AVC για τηλεοπτική εκπομπή στα τέλη του 2004.

1.2.3 Χαρακτηριστικά

Το H.264/AVC/MPEG-4 Part 10 περιέχει μια σειρά από νέες λειτουργίες που του επιτρέπουν να συμπιέζει πιο αποτελεσματικά από τα παλαιότερα πρότυπα και επίσης να παρέχει μεγαλύτερη ευελιξία για χρήση σε ένα μεγάλο φάσμα δικτυακών

περιβαλλόντων. Συγκεκριμένα, μερικά από αυτά τα βασικά χαρακτηριστικά περιλαμβάνουν:

- Διαπλασιασική πρόβλεψη με πολλαπλές εικόνες (Multi picture inter-picture prediction: αυτό το είδος πρόβλεψης προσπαθεί να επωφεληθεί από τον προσωρινό πλεονασμό μεταξύ γειτονικών εικόνων επιτρέποντας υψηλότερα ποσοστά συμπίεσης) με τα παρακάτω χαρακτηριστικά:

- Χρησιμοποιώντας εικόνες που έχουν ήδη κωδικοποιηθεί, ως σημείο αναφοράς, με πολύ πιο ευέλικτο τρόπο από ό, τι σε προηγούμενα πρότυπα, επιτρέποντας σε μερικές περιπτώσεις να χρησιμοποιηθούν έως και 16 πλαίσια αναφοράς.
- Αντιστάθμιση κίνησης με μεταβλητού μεγέθους block (VBSMC) με μέγεθος block από 16x16 έως και 4x4, επιτρέποντας τον ακριβή χωρισμό τμημάτων κινούμενων περιοχών. Τα υποστηριζόμενα μεγέθη μπλοκ πρόβλεψης φωτεινότητας περιλαμβάνουν τα 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 και 4x4, πολλά από τα οποία μπορούν να χρησιμοποιηθούν μαζί σε ένα μόνο macroblock. Τα μεγέθη μπλοκ πρόβλεψης χρώματος είναι σχετικά μικρότερα ανάλογα με την διαδικασία chroma subsampling που χρησιμοποιείται.
- Η δυνατότητα χρήσης πολλαπλών διανυσμάτων κίνησης ανά macroblock (ένα ή δύο ανά τμήμα) με μέγιστο αριθμό 32 στην περίπτωση ενός B macroblock που κατασκευάζεται από 16 4x4 τμήματα.
- Η δυνατότητα χρήσης οποιουδήποτε τύπου macroblock με B-frames, συμπεριλαμβανομένων των I-macroblocks, με αποτέλεσμα την πολύ πιο αποτελεσματική κωδικοποίηση όταν χρησιμοποιείτε B-frame.
- Ακρίβεια Quarter-pixel για αντιστάθμιση κίνησης, που επιτρέπει ακριβή περιγραφή των μετατοπίσεων των κινούμενων περιοχών. Για το χρώμα, η ανάλυση τυπικά μειώνεται κατά το ήμισυ τόσο κατακόρυφα όσο και οριζόντια και ως εκ τούτου η αντιστάθμιση κίνησης του χρώματος χρησιμοποιεί το 1/8 σε μονάδες πλέγματος pixel χρώματος.
- Σταθμισμένη πρόβλεψη, επιτρέποντας σε έναν κωδικοποιητή να καθορίζει τη χρήση της κλιμάκωσης και μετατόπισης κατά την

εκτέλεση της αντιστάθμισης της κίνησης, παρέχοντας σε ειδικές περιπτώσεις ένα σημαντικό πλεονέκτημα στην απόδοση - όπως οι μεταβολές εξασθένισης-μαύρου, ξεθωριασμού και διασταύρωσης.

- Ένα σχέδιο κωδικοποίησης εντροπίας που περιλαμβάνει:
 - Κωδικοποίηση δυαδικής αριθμητικής (Context-Adaptive Binary Arithmetic Coding) [36], ένας αλγόριθμος για την χωρίς απώλειες συμπίεση στοιχείων σύνταξης στην ροή του βίντεο, γνωρίζοντας τις πιθανότητες των στοιχείων σύνταξης σε ένα δεδομένο πλαίσιο. Το CABAC συμπιέζει δεδομένα πιο αποτελεσματικά από το CAVLC, αλλά απαιτεί πολύ περισσότερη επεξεργασία για να αποκωδικοποιηθεί.
 - Κωδικοποίηση μεταβλητού μήκους (Context-Adaptive Variable-Length Coding), η οποία είναι μια εναλλακτική λύση χαμηλότερης πολυπλοκότητας σε σχέση με την CABAC για την κωδικοποίηση κβαντισμένων τιμών συντελεστών μετασχηματισμού. Αν και με μικρότερη πολυπλοκότητα από το CABAC, το CAVLC είναι πιο περίπλοκο και πιο αποτελεσματικό από τις μεθόδους που χρησιμοποιούνται συνήθως για να κωδικοποιηθούν οι συντελεστές σε άλλα προηγούμενα σχέδια.
 - Μια απλή και πολύ δομημένη τεχνική κωδικοποίησης μεταβλητού μήκους (Variable Length Coding) για πολλά από τα στοιχεία σύνταξης που δεν κωδικοποιούνται από το CABAC ή το CAVLC, που αναφέρεται ως κωδικοποίηση Exponential-Golomb (ή Exp-Golomb).
- Λειτουργίες απώλειας ανθεκτικότητας (Loss resilience) που περιλαμβάνουν:
 - Ο ορισμός του Network Abstraction Layer (NAL) που επιτρέπει η ίδια σύνταξη βίντεο να χρησιμοποιείται σε πολλά περιβάλλοντα δικτύου. Μια πολύ θεμελιώδης ιδέα σχεδιασμού του H.264 είναι η δημιουργία αυτοτελών πακέτων, για την αφαίρεση του διπλασιασμού της επικεφαλίδας όπως στον κώδικα προέκτασης κεφαλίδας (Header Extension Code) του MPEG-4. Αυτό επιτεύχθηκε με την αποσύνδεση πληροφοριών σχετικών με περισσότερα από ένα slices από τη ροή πολυμέσων. Ο συνδυασμός των παραμέτρων υψηλότερου επιπέδου ονομάζεται σετ παραμέτρων (parameter set). Το πρότυπο H.264

περιλαμβάνει δύο τύπους παραμέτρων: Σει παραμέτρων ακολουθίας (Sequence Parameter Set) και σετ παραμέτρων εικόνας (Picture Parameter Set). Ένα ενεργό σετ παραμέτρων ακολουθίας παραμένει αμετάβλητο σε μια κωδικοποιημένη ακολουθία βίντεο και ένα ενεργό σετ παραμέτρων εικόνων παραμένει αμετάβλητο σε μια κωδικοποιημένη εικόνα. Οι δομές των δύο αυτών σετ (SPS, PPS) περιέχουν πληροφορίες όπως το μέγεθος της εικόνας, τους προαιρετικούς τρόπους κωδικοποίησης που χρησιμοποιούνται κ.λ.π..

- Flexible macroblock ordering (FMO), επίσης γνωστό και ως ομάδες slices, και Arbitrary slice ordering (ASO), οι οποίες είναι τεχνικές για την αναδιάρθρωση της σειράς της απεικόνισης των θεμελιωδών περιοχών (macroblocks) στις εικόνες. Αν και συνήθως θεωρούνται σαν χαρακτηριστικά σφάλματος/απώλειας ευστάθειας, τα FMO και ASO μπορούν επίσης να χρησιμοποιηθούν για άλλους σκοπούς.
- Η κατανομή δεδομένων (Data Partitioning), ένα χαρακτηριστικό που παρέχει τη δυνατότητα διαχωρισμού των πιο σημαντικών και λιγότερο σημαντικών στοιχείων σύνταξης σε διαφορετικά πακέτα δεδομένων, επιτρέποντας την εφαρμογή της άνισης προστασίας σφάλματος (Unequal Error Protection) και άλλους τύπους βελτίωσης της ανθεκτικότητας σφάλματος/απώλειας.
- Redundant slices (RS), ένα χαρακτηριστικό ανθεκτικότητας σφάλματος/απώλειας που επιτρέπει σε έναν κωδικοποιητή να στείλει μια επιπλέον αναπαράσταση μιας περιοχής της εικόνας (συνήθως σε χαμηλότερη πιστότητα) και μπορεί να χρησιμοποιηθεί στην περίπτωση όπου η πρωτεύουσα αναπαράσταση είναι κατεστραμμένη ή έχει χαθεί.
- Αρίθμηση καρτέ, ένα χαρακτηριστικό που επιτρέπει τη δημιουργία «υπο-ακολουθιών», επιτρέποντας τη χρονική κλιμάκωση με προαιρετική ενσωμάτωση επιπλέον εικόνων μεταξύ άλλων εικόνων και την ανίχνευση και απόκρυψη απώλειας ολόκληρων εικόνων που μπορεί να προκύψουν λόγω απώλειας πακέτων δικτύου ή σφαλμάτων.
- Η εναλλαγή slices, που ονομάζεται SP και SI slices, επιτρέπει σε έναν κωδικοποιητή να κατευθύνει έναν αποκωδικοποιητή για να μεταπηδήσει σε μια τρέχουσα ροή βίντεο για τέτοιους σκοπούς, όπως η μετάδοση ροής βίντεο

και η λειτουργία trick mode. Όταν ένας αποκωδικοποιητής μεταπηδά στη μέση μιας ροής βίντεο χρησιμοποιώντας τη λειτουργία SP/SI, μπορεί να ταιριάξει επακριβώς με τις αποκωδικοποιημένες εικόνες σε εκείνη τη θέση στην ροή βίντεο παρά τη χρήση διαφορετικών εικόνων ή καθόλου εικόνων, ως παραπομπές πριν την εναλλαγή.

1.2.4 Προφίλ

Το πρότυπο ορίζει ένα σύνολο δυνατοτήτων, τα οποία αναφέρονται ως προφίλ, με στόχο συγκεκριμένες κατηγορίες εφαρμογών. Ορίζονται ως ένα σύνολο περιορισμών που εφαρμόζονται στον κωδικοποιητή και αυτό επιτρέπει στον αποκωδικοποιητή να αναγνωρίσει τις απαιτήσεις για την αποκωδικοποίηση.

Κάποια από τα προφίλ για μη κλιμακούμενες εφαρμογές βίντεο 2D περιλαμβάνουν τα ακόλουθα:

- Constrained Baseline Profile (CBP): Πρωτίστως για εφαρμογές χαμηλού κόστους, αυτό το προφίλ χρησιμοποιείται κατά κανόνα στις εφαρμογές τηλεδιάσκεψης και κινητής τηλεφωνίας.
- Baseline Profile (BP): Κατά κύριο λόγο για εφαρμογές χαμηλού κόστους που απαιτούν πρόσθετη ανθεκτικότητα απώλειας δεδομένων. Το προφίλ αυτό χρησιμοποιείται σε ορισμένες εφαρμογές τηλεδιάσκεψης και κινητής τηλεφωνίας και επίσης περιλαμβάνει όλες τις λειτουργίες που υποστηρίζονται στο CBP, καθώς και τρεις πρόσθετες λειτουργίες
- Extended Profile (XP): Προοριζόμενο ως προφίλ για βίντεο streaming, αυτό το προφίλ έχει σχετικά υψηλή ικανότητα συμπίεσης και επιπλέον λειτουργίες για ανθεκτικότητα σε απώλειες δεδομένων και εναλλαγή διακομιστή.
- Main Profile (MP): Αυτό το προφίλ χρησιμοποιείται στην ψηφιακή τηλεόραση με τυπική ευκρίνεια που χρησιμοποιούν τη μορφή MPEG-4. Δεν χρησιμοποιείται, ωστόσο, για υψηλής ευκρίνειας, καθώς αναπτύχθηκε το 2004 το High Profile για αυτή την εφαρμογή.
- High Profile (HiP): Το πρωτεύον προφίλ για εφαρμογές αναμετάδοσης και αποθήκευσης δίσκων, ιδιαίτερα για εφαρμογές υψηλής ευκρίνειας.
- Progressive High Profile (PHiP): Παρόμοιο με το High Profile.

- Constrained High Profile: Παρόμοιο με το Progressive High Profile, αλλά χωρίς την υποστήριξη των B-slices (bi-predictive).
- High 10 Profile (Hi10P): Το προφίλ αυτό βασίζεται στο High Profile, προσθέτοντας υποστήριξη έως και 10 bits ανά δείγμα αποκωδικοποιημένης ακρίβειας εικόνας.
- High 4:2:2 Profile (Hi422P): Στοχεύει κυρίως σε επαγγελματικές εφαρμογές. Το προφίλ αυτό βασίζεται στο High 10 Profile, προσθέτοντας υποστήριξη για 4:2:2 chroma subsampling ενώ χρησιμοποιεί έως και 10 bits ανά δείγμα ακρίβειας αποκωδικοποιημένης εικόνας.
- High 4:4:4 Predictive Profile (Hi444PP): Αυτό το προφίλ βασίζεται στο High 4:2:2 Profile, υποστηρίζοντας 4:4:4 chroma subsampling, έως και 14 bit ανά δείγμα και υποστηρίζοντας επιπλέον την αποτελεσματική κωδικοποίηση χωρίς απώλειες και την κωδικοποίηση κάθε εικόνας ως τριών διαφορετικών χρωμάτων.

Για τις βιντεοκάμερες, την επεξεργασία και τις επαγγελματικές εφαρμογές, το πρότυπο περιέχει τέσσερα πρόσθετα ενδοπλαισιακά (intra-frame) προφίλ, τα οποία ορίζονται ως απλά υποσύνολα άλλων αντίστοιχων προφίλ:

- High 10 Intra Profile: Το High 10 Profile, περιορίζεται σε all-Intra χρήση.
- High 4:2:2 Intra Profile: Το High 4:2:2 Profile, περιορίζεται σε all-Intra χρήση.
- High 4:4:4 Intra Profile: Το High 4:4:4 Predictive Profile περιορίζεται σε All-Intra χρήση.
- CAVLC 4:4:4 Intra Profile: Το High 4:4:4 Predictive Profile περιορίζεται σε All-Intra χρήση σ και στην κωδικοποίηση εντροπίας CAVLC.

1.3 High Efficiency Video Coding (HEVC)

Το HEVC [35], επίσης γνωστό ως H.265 και MPEG-H Part 2, είναι ένα πρότυπο συμπίεσης βίντεο, ένα από τους πολλούς πιθανούς διαδόχους του πιο ευρέως χρησιμοποιούμενου AVC (H.264/MPEG-4 Part 10). Σε σύγκριση με το AVC, το HEVC προσφέρει διπλάσια αναλογία συμπίεσης για το ίδιο επίπεδο ποιότητας βίντεο ή περισσότερο βελτιωμένη ποιότητα βίντεο με τον ίδιο ρυθμό μετάδοσης δεδομένων (bit rate). Υποστηρίζει αναλύσεις μέχρι 8192x4320, συμπεριλαμβανομένου και 8K UHD.

Σε κάθε περίπτωση, το HEVC είναι μια επέκταση της ιδέας του H.264/MPEG-4 AVC. Και τα δύο λειτουργούν συγκρίνοντας διάφορα τμήματα ενός καρέ ενός βίντεο με σκοπό την εύρεση περιττών κομματιών, τόσο μέσα σε ένα καρέ όσο και σε επόμενα. Αυτά τα περιττά κομμάτια αντικαθιστούνται στη συνέχεια με μια μικρή περιγραφή αντί για τα αρχικά pixels (εικονοστοιχεία).

Μια από τις κύριες αλλαγές στο HEVC περιλαμβάνει την επέκταση της αλληλουχίας σύγκρισης και κωδικοποίησης διαφορών από 16x16 pixels σε μέγεθος έως και 64x64 pixels, βελτιωμένη τμηματοποίηση μεταβλητής-block-μεγέθους (variable-block-size segmentation), βελτιωμένη πρόβλεψη “intra” μέσα στην ίδια εικόνα, βελτιωμένη πρόβλεψη με motion vector (διάνυσμα κίνησης: στη συμπίεση βίντεο είναι το βασικό στοιχείο στη διαδικασία εκτίμησης κίνησης. Χρησιμοποιείται για να αντιπροσωπεύσει ένα macroblock σε μια εικόνα και βασίζεται στη θέση αυτού του macroblock, ή ενός παρόμοιου, σε μια άλλη εικόνα, που ονομάζεται εικόνα αναφοράς) και συγχώνευση της περιοχής κίνησης, βελτιωμένο φίλτράρισμα αντιστάθμισης κίνησης και ένα επιπλέον βήμα φιλτραρίσματος που ονομάζεται sample-adaptive offset filtering (φίλτρο αντιστάθμισης προσαρμοστικού δείγματος). Η αποτελεσματική χρήση αυτών των βελτιώσεων απαιτεί πολύ περισσότερη δυνατότητα επεξεργασίας σήματος για τη συμπίεση του βίντεο, αλλά έχει μικρότερη επίπτωση στο ποσό υπολογισμού που απαιτείται για αποσυμπίεση.

Το HEVC αναπτύχθηκε από την JCT-VC (Joint Collaborative Team on Video Coding), μια συνεργασία μεταξύ των ISO/IEC MPEG και ITU-T VCEG. Η ομάδα ISO/IEC αναφέρεται σε αυτήν ως MPEG-H Part 2 και η ομάδα ITU-T ως H.265. Η πρώτη έκδοση του προτύπου HEVC επικυρώθηκε τον Ιανουάριο του 2013 και δημοσιεύθηκε τον Ιούνιο του 2013. Η δεύτερη έκδοση, με επεκτάσεις Multiview (MV-HEVC), επεκτάσεις RExt (range extensions) και επεκτάσεις SHVC (scalability extensions), ολοκληρώθηκε και εγκρίθηκε το 2014 και δημοσιεύθηκε στις αρχές του 2015. επεκτάσεις για 3D βίντεο (3D-HEVC) ολοκληρώθηκαν στις αρχές του 2015, και επεκτάσεις για την κωδικοποίηση του περιεχόμενου οθόνης (SCC, screen content coding) ολοκληρώθηκαν στις αρχές του 2016 και δημοσιεύθηκε στις αρχές του 2017, καλύπτοντας βίντεο με περιεχόμενο γραφικών, κειμένου, ή animation (κινούμενα σχέδια) καθώς και (ή αντί για) σκηνές βίντεο που έχουν τραβηχτεί με κάμερα. Τον Οκτώβριο του 2017, το πρότυπο αναγνωρίστηκε από τα βραβεία Primetime Emmy Engineering Awards ως έχοντας επηρεάσει σημαντικά την τεχνολογία της τηλεόρασης.

1.3.1 Ιστορία

Το 2004, η Ομάδα ITU-T VCEG (Video Coding Experts Group) ξεκίνησε μια σημαντική μελέτη των τεχνολογικών εξελίξεων που θα μπορούσαν να επιτρέψουν τη δημιουργία ενός νέου προτύπου συμπίεσης βίντεο (ή σημαντικών βελτιώσεων με γνώμονα την συμπίεση του προτύπου H.264 / MPEG-4 AVC) . Από τον Οκτώβριο του 2004, εξετάστηκαν διάφορες τεχνικές για την πιθανή βελτίωση του προτύπου H.264.

Δύο προσεγγίσεις επικράτησαν για την τυποποίηση της τεχνολογίας ενισχυμένης συμπίεσης: είτε δημιουργώντας ένα νέο πρότυπο είτε δημιουργώντας επεκτάσεις του H.264 / MPEG-4 AVC. Το έργο είχε προσωρινά ονόματα H.265 και H.NGVC (Next Generation Video Coding) και ήταν ένα σημαντικό μέρος της δουλειάς της ομάδας VCEG έως την εξέλιξή του στο πρόγραμμα HEVC από την κοινή προσπάθεια μαζί με την ομάδα MPEG το 2010.

Οι απαιτήσεις για το NGVC ήταν η δυνατότητα μείωσης του bit rate κατά 50% με την ίδια ποιότητα εικόνας σε σύγκριση με το H.264/MPEG-4 AVC High profile και η υπολογιστική πολυπλοκότητα όπου κυμαίνεται από 1/2 έως 3 φορές εκείνη του High Profile. Το NGVC θα έπρεπε να είναι σε θέση να προσφέρει μείωση του bit rate κατά 25% μαζί με μείωση της πολυπλοκότητας κατά 50% με την ίδια ποιότητα εικόνας όπως αυτή του High Profile ή να προσφέρει μεγαλύτερη μείωση του bit rate με κάπως μεγαλύτερη πολυπλοκότητα.

Η ομάδα ISO/IEC MPEG (Moving Picture Experts Group) ξεκίνησε ένα παρόμοιο έργο το 2007, ονομάζοντάς το High Performance Video Coding. Ο στόχος του σχεδίου, τον Ιούλιο του 2007, αποφασίστηκε να είναι η επίτευξη μείωσης του bit rate κατά 50%. Οι πρώτες αξιολογήσεις έγιναν με τροποποιήσεις του κωδικοποιητή που αναπτύχθηκε από την VCEG. Μέχρι τον Ιούλιο του 2009, τα πειραματικά αποτελέσματα έδειξαν μείωση του bit rate κατά περίπου 20% σε σύγκριση με το AVC High Profile. Αυτά τα αποτελέσματα ώθησαν την MPEG να ξεκινήσει την προσπάθεια τυποποίησης σε συνεργασία με την VCEG.

Οι VCEG και MPEG εξέτασαν και αξιολόγησαν αρκετές προτάσεις κατά την πρώτη συνεδρίαση της κοινής ομάδας συνεργασίας που πραγματοποιήθηκε τον Απρίλιο 2010. Υποβλήθηκαν συνολικά 27 πλήρεις προτάσεις. Οι αξιολογήσεις έδειξαν ότι ορισμένες προτάσεις θα μπορούσαν να φτάσουν την ίδια οπτική ποιότητα με το AVC με μόλις το μισό bit rate σε πολλές από τις δοκιμές, με κόστος 2-10 φορές αύξηση της υπολογιστικής πολυπλοκότητας και μερικές προτάσεις πέτυχαν αποτελέσματα με υποκειμενικά καλή ποιότητα και bit rate με χαμηλότερη υπολογιστική πολυπλοκότητα

από τις κωδικοποιήσεις με το πρότυπο AVC High Profile. Κατά τη συνάντηση αυτή, υιοθετήθηκε το όνομα High Efficiency Video Coding (HEVC) για το κοινό σχέδιο.

Στις 25 Ιανουαρίου 2013 το HEVC έλαβε την πρώτη έγκριση (έγκριση πρώτου σταδίου) και στις 13 Απριλίου 2013 εγκρίθηκε ως πρότυπο. Το πρότυπο δημοσιεύθηκε επισήμως από την ITU-T στις 7 Ιουνίου 2013 και από την ISO/IEC MPEG στις 25 Νοεμβρίου 2013.

Στις 11 Ιουλίου 2014, η MPEG ανακοίνωσε ότι η 2η έκδοση του HEVC θα περιέχει τρεις πρόσφατα ολοκληρωμένες επεκτάσεις, οι οποίες είναι οι επεκτάσεις multiview (MV-HEVC), οι επεκτάσεις Rext (range extensions) και οι επεκτάσεις SHVC (scalability extensions).

Στις 29 Οκτωβρίου 2014, η 2^η έκδοση HEVC/H.265 εγκρίθηκε ως πρότυπο. Στη συνέχεια δημοσιεύθηκε επισήμως στις 12 Ιανουαρίου 2015.

Στις 29 Απριλίου 2015, η 3^η έκδοση του HEVC/H.265 εγκρίθηκε ως πρότυπο, ενώ στις 3 Ιουνίου 2016, σε μια πρώτη προσπάθεια έγκρισης, η 4^η έκδοση HEVC/H.265 απέτυχε, επιστρέφοντας στις 22 Δεκεμβρίου 2016 όπου κατάφερε και εγκρίθηκε ως πρότυπο.

1.3.2 Αποτελεσματικότητα της κωδικοποίησης

Ο σχεδιασμός των περισσότερων προτύπων κωδικοποίησης βίντεο στοχεύει πρωτίστως στην επίτευξη της υψηλότερης απόδοσης κωδικοποίησης. Η αποδοτικότητα της κωδικοποίησης είναι η δυνατότητα κωδικοποίησης του βίντεο με το χαμηλότερο δυνατό bit rate ενώ διατηρείται ένα ορισμένο επίπεδο ποιότητας βίντεο. Υπάρχουν δύο τρόποι μέτρησης της αποτελεσματικότητας της κωδικοποίησης ενός προτύπου και αυτοί είναι είτε η χρήση αντικειμενικής μέτρησης (objective metric), όπως το PSNR (peak signal-to-noise ratio: η αναλογία μεταξύ της μέγιστης δυνατής ισχύος ενός σήματος και της ισχύος του θορύβου που επηρεάζει την πιστότητα/ποιότητα της αναπαράστασης), ή η χρήση υποκειμενικής αξιολόγησης της ποιότητας του βίντεο. Η υποκειμενική αξιολόγηση της ποιότητας του βίντεο θεωρείται ο πιο σημαντικός τρόπος μέτρησης ενός προτύπου κωδικοποίησης βίντεο, δεδομένου ότι οι άνθρωποι αντιλαμβάνονται υποκειμενικά την ποιότητα του βίντεο.

Το HEVC επωφελείται από τη χρήση μεγαλύτερου μεγέθους CTU (coding tree unit: βασική μονάδα επεξεργασίας του προτύπου κωδικοποίησης βίντεο HEVC και εννοιολογικά αντιστοιχεί σε δομή μακροστοιχείων/macroblock units που

χρησιμοποιήθηκαν σε πολλά προηγούμενα πρότυπα βίντεο). Αυτό αποδείχθηκε μετά από δοκιμές με PSNR και με χρήση κωδικοποιητή HM-8.0 HEVC όπου αναγκάστηκε να χρησιμοποιήσει προοδευτικά μικρότερα σε μέγεθος CTU [22], [24]. Οι δοκιμές έδειξαν ότι τα μεγαλύτερα σε μέγεθος CTU αυξάνουν την αποτελεσματικότητα της κωδικοποίησης μειώνοντας παράλληλα τον χρόνο αποκωδικοποίησης.

Στη σύγκριση κωδικοποιητών βίντεο που έγινε στο [9], ο κωδικοποιητής HM-10.0 HEVC συγκρίθηκε με τον κωδικοποιητή x264 (έκδοση r2334) και τον κωδικοποιητή VP9 (έκδοση v1.2.0-3088-ga81bd12). Η σύγκριση χρησιμοποίησε τη μέθοδο μέτρησης bit-rate Bjøntegaard-Delta (BD-BR), στην οποία οι αρνητικές τιμές δείχνουν πόσο πιο χαμηλά έχει μειωθεί το bit rate και οι θετικές τιμές δείχνουν πόσο αυξήθηκε το bit rate για το ίδιο PSNR. Στη σύγκριση, ο κωδικοποιητής HM-10.0 HEVC είχε την υψηλότερη αποτελεσματικότητα κωδικοποίησης και, κατά μέσο όρο, για να επιτύχει την ίδια αντικειμενική ποιότητα, ο κωδικοποιητής x264 χρειάζεται να αυξήσει το bit rate κατά 66,4%, ενώ ο κωδικοποιητής VP9 χρειάζεται να αυξήσει το bit rate κατά 79,4%.

1.3.3 Χαρακτηριστικά

Το HEVC σχεδιάστηκε για να βελτιώσει σημαντικά την αποτελεσματικότητα της κωδικοποίησης σε σύγκριση με την H.264 / MPEG-4 AVC HP, δηλαδή να μειώσει τις απαιτήσεις bitrate κατά το ήμισυ με συγκρίσιμη ποιότητα εικόνας, εις βάρος της αυξημένης υπολογιστικής πολυπλοκότητας. Το HEVC σχεδιάστηκε με στόχο να επιτρέψει στο περιεχόμενο βίντεο να έχει λόγο συμπίεσης δεδομένων έως 1000:1. Ανάλογα με τις απαιτήσεις, οι κωδικοποιητές HEVC μπορούν να «ανταλλάξουν» την υπολογιστική πολυπλοκότητα, το ποσοστό συμπίεσης, την ανθεκτικότητα σε σφάλματα και τον χρόνο καθυστέρησης της κωδικοποίησης. Δύο από τα βασικά χαρακτηριστικά που βελτίωσε το HEVC σε σύγκριση με το H.264 / MPEG-4 AVC ήταν η υποστήριξη βίντεο υψηλής ανάλυσης και βελτιωμένων μεθόδων παράλληλης επεξεργασίας.

Το HEVC απευθύνεται κυρίως για εικόνες HDTV νέας γενιάς και συστήματα συλλογής περιεχομένου, τα οποία περιλαμβάνουν προοδευτική σάρωση frame rate και αναλύσεις από QVGA (320x240) έως 4320p (7680x4320), καθώς και βελτιωμένη ποιότητα εικόνας όσον αφορά το επίπεδο θορύβου, τους χρωματικούς χώρους και δυναμικό εύρος.

1.3.3.1 Επίπεδα κωδικοποίησης βίντεο

Τα επίπεδα κωδικοποίησης βίντεο HEVC χρησιμοποιούν την ίδια «υβριδική» προσέγγιση που χρησιμοποιούνται σε όλα τα σύγχρονα πρότυπα βίντεο, ξεκινώντας από το H.261, στο ότι χρησιμοποιεί διαπλαισιακή/ενδοπλαισιακή (inter-/intra-picture) πρόβλεψη και κωδικοποίηση μετασχηματισμού 2D. Ένας κωδικοποιητής HEVC προχωρά αρχικά διαιρώντας μια εικόνα σε block για την πρώτη εικόνα (ή την πρώτη εικόνα ενός τυχαίου σημείου πρόσβασης) που χρησιμοποιεί την ενδοπλαισιακή πρόβλεψη. Η ενδοπλαισιακή πρόβλεψη είναι όταν η πρόβλεψη των block στην εικόνα βασίζεται μόνο στις πληροφορίες της εικόνας αυτής. Για όλες τις άλλες εικόνες, χρησιμοποιείται η διαπλαισιακή πρόβλεψη, στην οποία χρησιμοποιούνται πληροφορίες πρόβλεψης από άλλες εικόνες. Αφού ολοκληρωθούν οι μέθοδοι πρόβλεψης και η εικόνα περάσει από τα φίλτρα βρόχου, η τελική απεικόνιση της εικόνας αποθηκεύεται στο decoded picture buffer. Οι εικόνες αυτές μπορούν να χρησιμοποιηθούν για την πρόβλεψη άλλων εικόνων.

Το HEVC σχεδιάστηκε με την ιδέα ότι θα χρησιμοποιηθεί η προοδευτική σάρωση βίντεο (progressive scanning) και δεν προστέθηκαν εργαλεία κωδικοποίησης. Ειδικά εργαλεία κωδικοποίησης του Interlaced, όπως το MBAFF και το PAFF, δεν υποστηρίζονται στο HEVC. Το HEVC στέλνει αντ' αυτού metadata που αναφέρουν τον τρόπο αποστολής του επεξεργασμένου βίντεο. Το επεξεργασμένο βίντεο μπορεί να αποστέλλεται είτε με την κωδικοποίηση κάθε frame ως ξεχωριστή εικόνα είτε με την κωδικοποίηση κάθε πεδίου ως χωριστής εικόνας. Για το επεξεργασμένο βίντεο (Interlaced video), το HEVC μπορεί να αλλάξει μεταξύ κωδικοποίησης frame και κωδικοποίησης πεδίων χρησιμοποιώντας το SAFF (Sequence Adaptive Frame Field), το οποίο επιτρέπει την αλλαγή της λειτουργίας κωδικοποίησης για κάθε ακολουθία βίντεο. Αυτό επιτρέπει την αποστολή του επεξεργασμένου βίντεο (interlaced video) με HEVC χωρίς να χρειάζεται προσθήκη ειδικών διεργασιών αποκωδικοποίησης στους αποκωδικοποιητές HEVC.

1.3.3.2 Color spaces

Το πρότυπο HEVC υποστηρίζει συστήματα χρωμάτων όπως generic film, NTSC, PAL, Rec. 601, Rec. 709, Rec. 2020, Rec. 2100, SMPTE 170M, SMPTE 240M, sRGB,

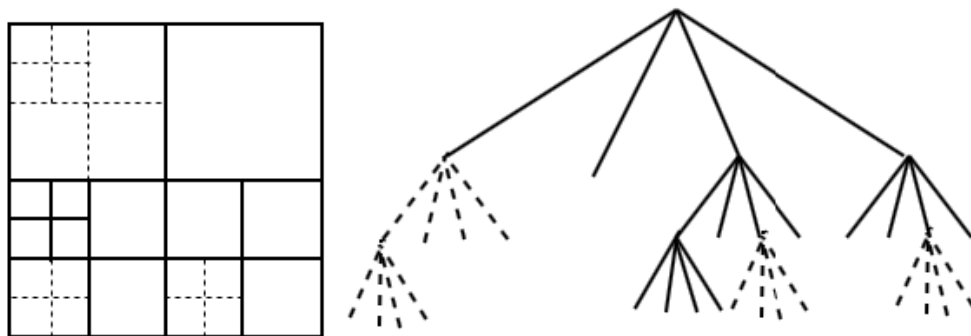
sYCC, xvYCC και XYZ. Το HEVC επίσης υποστηρίζει αναπαραστάσεις κωδικοποίησης χρώματος όπως RGB, YCbCr και YCoCg.

1.3.4 Εργαλεία κωδικοποίησης

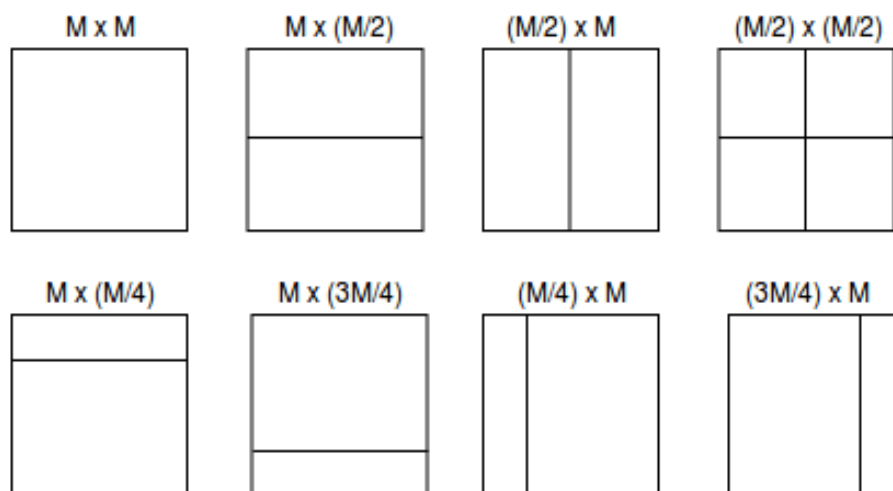
1.3.4.1 Coding Tree Unit

Το CTU (Coding tree unit) είναι η βασική μονάδα επεξεργασίας του προτύπου HEVC και εννοιολογικά αντιστοιχεί σε δομή με μονάδες macroblock που χρησιμοποιήθηκαν σε αρκετά προηγούμενα πρότυπα βίντεο.

Το HEVC αντικαθιστά τα macroblocks 16x16 pixel, τα οποία χρησιμοποιήθηκαν με προηγούμενα πρότυπα, με τα CTUs που μπορούν να χρησιμοποιήσουν μεγαλύτερες δομές block έως τα 64x64 και μπορούν να υποδιαιρέσουν καλύτερα την εικόνα σε δομές μεταβλητού μεγέθους. Το HEVC αρχικά διαιρεί την εικόνα σε CTU που μπορεί να είναι 64x64, 32x32 ή 16x16 με μεγαλύτερο μέγεθος pixel block συνήθως αυξάνοντας την απόδοση της κωδικοποίησης. Η Εικόνα 3 δείχνει παράδειγμα διάσπασης ενός CTU σε πολλαπλά CU (Coding Units) ακολουθώντας Quad tree. Η Εικόνα 4 δείχνει τα πιθανά Prediction Units (PUs) βάσει των οποίων μπορεί να γίνει η πρόβλεψη ενός CU.



Εικόνα 3: Δόμηση CTU ως QuadTree.



Εικόνα 4: Επιτρεπτές διαιρέσεις ενός μπλοκ CU σε υπομπλόκ PU για λόγους πρόβλεψης.

1.3.4.2 Εργαλεία παράλληλης επεξεργασίας

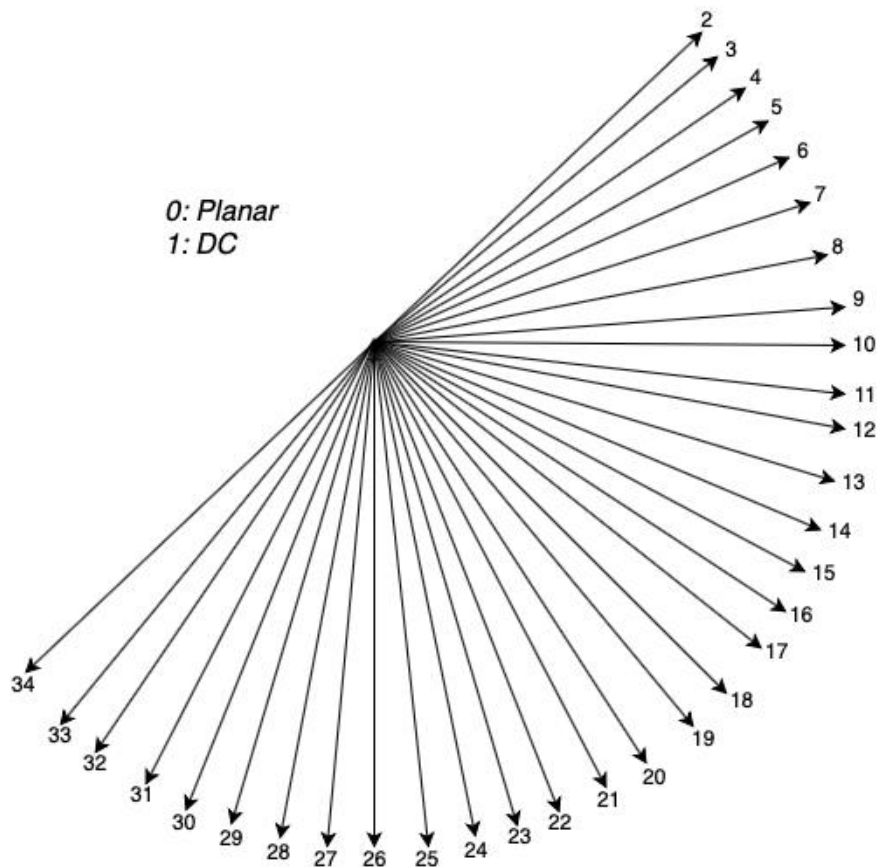
- Τα πλακάκια/tiles επιτρέπουν στην εικόνα να διαιρεθεί σε ένα πλέγμα ορθογώνιων περιοχών που μπορούν ανεξάρτητα να αποκωδικοποιηθούν / κωδικοποιηθούν. Ο κύριος σκοπός είναι να επιτρέπεται η παράλληλη επεξεργασία. Τα πλακάκια/tiles μπορούν να αποκωδικοποιηθούν ανεξάρτητα και να επιτρέψουν την τυχαία πρόσβαση σε συγκεκριμένες περιοχές μιας εικόνας σε μια ροή βίντεο.
- Η παράλληλη επεξεργασία Wavefront (WPP) είναι όταν ένα slice χωρίζεται σε σειρές CTU στις οποίες η πρώτη σειρά αποκωδικοποιείται κανονικά, αλλά κάθε πρόσθετη σειρά απαιτεί να λαμβάνονται αποφάσεις στην προηγούμενη σειρά. Το WPP έχει τις πληροφορίες χρήσης του κωδικοποιητή εντροπίας από την προηγούμενη σειρά CTU και επιτρέπει μια μέθοδο παράλληλης επεξεργασίας που μπορεί να επιτρέψει την καλύτερη συμπίεση από τα πλακάκια/tiles.
- Τα tiles και το WPP επιτρέπονται, αλλά είναι προαιρετικά. Εάν υπάρχουν tiles, θα πρέπει να έχουν ύψος τουλάχιστον 64 pixels και πλάτος 256 pixels, με ειδικό όριο για τον αριθμό των tiles που επιτρέπονται.
- Τα slices μπορούν, ως επί το πλείστον, να αποκωδικοποιούνται ανεξάρτητα το ένα από το άλλο, ώστε να επιτυγχάνεται ο εκ νέου συγχρονισμός σε περίπτωση απώλειας δεδομένων στην ροή βίντεο. Τα slices μπορούν να οριστούν ως αυτοτελή τμήματα στα οποία η πρόβλεψη δεν γίνεται πέρα από τα όριά τους. Όταν

διενεργείται φιλτράρισμα εντός-βρόχου σε μια εικόνα όμως, μπορεί να απαιτούνται πληροφορίες πέρα των ορίων. Τα slices είναι ομάδες CTUs και χωρίζονται σε 3 τύπους I, P και οι τύποι B.

- I-Slices: Slices για τα οποία γίνεται αποκλειστικά ενδοπλαισιακή πρόβλεψη (intra-picture prediction).
 - P-Slices: Slices για τα οποία γίνεται εκτός από ενδοπλαισιακή πρόβλεψη και διαπλαισιακή πρόβλεψη (inter-picture prediction) χρησιμοποιώντας μόνο προηγούμενα frames ως σημείο αναφοράς.
 - B-Slices: Slices για τα οποία γίνεται εκτός από ενδοπλαισιακή πρόβλεψη και διαπλαισιακή πρόβλεψη, χρησιμοποιώντας προηγούμενα καθώς και μελλοντικά frames ως σημείο αναφοράς.
- Τα εξαρτημένα slices μπορούν να επιτρέψουν την πρόσβαση των δεδομένων που σχετίζονται με τα tiles ή WPP πιο γρήγορα από το σύστημα απ'ό,τι εάν έπρεπε να αποκωδικοποιηθεί ολόκληρο το slice. Ο κύριος σκοπός των εξαρτημένων slices είναι να επιτρέπουν την κωδικοποίηση βίντεο low-delay λόγω της μικρότερης καθυστέρησης.

1.3.4.3 Κωδικοποίηση εντροπίας

Το HEVC χρησιμοποιεί έναν αλγόριθμο δυαδικής αριθμητικής κωδικοποίησης (CABAC) που βασικά είναι παρόμοιος με τον CABAC στο H.264 / MPEG-4 AVC. Το CABAC είναι η μόνη μέθοδος κωδικοποιητή εντροπίας που επιτρέπεται στο HEVC ενώ υπάρχουν δύο μέθοδοι κωδικοποιητή εντροπίας που επιτρέπονται από το H.264 / MPEG-4 AVC. Το CABAC και η κωδικοποίηση εντροπίας στο HEVC σχεδιάστηκαν για υψηλότερη διακίνηση από το H.264 / MPEG-4 AVC διατηρώντας παράλληλα υψηλότερη απόδοση συμπίεσης για μεγαλύτερα σε μέγεθος block μετασχηματισμού σε σχέση με απλές επεκτάσεις. Μια άλλη βελτίωση με το HEVC είναι ότι οι εξαρτήσεις μεταξύ των κωδικοποιημένων δεδομένων έχουν αλλάξει για να αυξήσουν περαιτέρω την διακίνηση.



Εικόνα 5: Οι κατευθύνσεις της ενδοπλαισιακής πρόβλεψης στο πρότυπο HEVC.

1.3.4.4 Ενδοπλαισιακή πρόβλεψη

Το HEVC καθορίζει 33 τρόπους κατεύθυνσης (βλέπε Εικόνα 5) για την ενδοπλαισιακή πρόβλεψη σε σύγκριση με τους 8 που καθορίζονται από το H.264. Το HEVC επίσης καθορίζει τους τρόπους ενδοπλαισιακή πρόβλεψη DC και επίπεδης πρόβλεψης (planar prediction). Η ενδοπλαισιακή πρόβλεψη DC παράγει μια μέση τιμή από τον μέσο όρο των δειγμάτων αναφοράς και μπορεί να χρησιμοποιηθεί για επίπεδες επιφάνειες. Η λειτουργία επίπεδης πρόβλεψης στο HEVC υποστηρίζει όλα τα μεγέθη μπλοκ που ορίζονται στο HEVC ενώ στο H.264 περιορίζεται σε μέγεθος block 16x16 pixels. Οι τρόποι ενδοπλαισιακής πρόβλεψης χρησιμοποιούν δεδομένα από γειτονικά block πρόβλεψης που έχουν προηγουμένως αποκωδικοποιηθεί από την ίδια εικόνα.

1.3.4.5 Φίλτρα βρόχου (Loop filters)

Το HEVC ορίζει δύο φίλτρα βρόχου που εφαρμόζονται διαδοχικά, με το deblocking filter (DBF) να εφαρμόζεται πρώτο και το sample adaptive offset filter (SAO) να εφαρμόζεται στη συνέχεια. Και τα δύο φίλτρα βρόχου εφαρμόζονται στον βρόχο διαπλαισιακής πρόβλεψης, δηλ. η φιλτραρισμένη εικόνα αποθηκεύεται στο decoded picture buffer (DPB) ως σημείο αναφοράς για την διαπλαισιακή πρόβλεψη.

- **Deblocking filter:** Το DBF που χρησιμοποιεί το HEVC είναι παρόμοιο με αυτό που χρησιμοποιείται από το H.264, αλλά με έναν απλούστερο σχεδιασμό και καλύτερη υποστήριξη για παράλληλη επεξεργασία. Στο HEVC το DBF εφαρμόζεται μόνο σε ένα πλέγμα δειγμάτων 8x8 ενώ το H.264 εφαρμόζεται σε ένα πλέγμα δειγμάτων 4x4. Το DBF χρησιμοποιεί ένα πλέγμα δειγμάτων 8x8, καθώς δεν προκαλεί αξιοσημείωτη υποβάθμιση και βελτιώνει σημαντικά την παράλληλη επεξεργασία επειδή το DBF δεν προκαλεί πλέον αλληλεπιδράσεις με άλλες λειτουργίες. Το HEVC απαιτεί επίσης το DBF να εφαρμόζει πρώτα οριζόντιο φιλτράρισμα για τα κατακόρυφα άκρα στην εικόνα και μόνο μετά από αυτό εφαρμόζει το κάθετο φιλτράρισμα για τις οριζόντιες άκρες της εικόνας. Αυτό επιτρέπει τη χρήση πολλαπλών παράλληλων threads.
- **Sample adaptive offset filter:** Το φίλτρο SAO εφαρμόζεται μετά το DBF και έχει σχεδιαστεί για να επιτρέπει την καλύτερη ανασυγκρότηση του αρχικού σήματος. Το φίλτρο SAO μπορεί να απενεργοποιηθεί ή να εφαρμοστεί με έναν από τους δύο τρόπους: με λειτουργία άκρου (edge offset mode) ή λειτουργία ζώνης (band offset mode). Η λειτουργία άκρου συγκρίνει την τιμή ενός δείγματος σε δύο από τους οκτώ γείτονές του χρησιμοποιώντας ένα από τα τέσσερα μοτίβα κατεύθυνσης. Με βάση τη σύγκριση, το δείγμα ταξινομείται σε μία από τις πέντε κατηγορίες: ελάχιστο, μέγιστο, ένα άκρο με τη χαμηλότερη τιμή, ένα άκρο με την υψηλότερη τιμή ή μονοτονικό. Για κάθε μία από τις τέσσερις πρώτες κατηγορίες εφαρμόζεται μια αντιστάθμιση. Η λειτουργία ζώνης εφαρμόζει μια μετατόπιση που βασίζεται στο πλάτος ενός μόνο δείγματος. Ένα δείγμα ταξινομείται από το πλάτος του σε μία από τις 32 ζώνες (δοχεία ιστογράμματος). Οι μετατοπίσεις καθορίζονται για τέσσερις διαδοχικές από τις 32 ζώνες.

1.3.5 Προφίλ

Το πρότυπο HEVC χρησιμοποιεί προφίλ κατά τον ίδιο τρόπο με το πρότυπο H.264. Στην πρώτη έκδοση του προτύπου τα προτεινόμενα προφίλ είναι:

- **Main profile:** Το main profile επιτρέπει ένα βάθος 8 bits ανά δείγμα με δειγματοληψία χρώματος 4:2:0, το οποίο είναι ο συνηθέστερος τύπος βίντεο που χρησιμοποιείται.
- **Main 10 profile:** Το main 10 profile επιτρέπει ένα βάθος 8-bit έως 10-bit ανά δείγμα με δειγματοληψία χρώματος 4:2:0. Επιτρέπει την βελτιωμένη ποιότητα βίντεο, καθώς μπορεί να υποστηρίξει βίντεο με μεγαλύτερο βάθος bit από αυτό που υποστηρίζεται από το main profile. Επιπλέον, στο main 10, ένα βίντεο 8-bit μπορεί να κωδικοποιηθεί σε μεγαλύτερο βάθος bit των 10-bit, το οποίο επιτρέπει βελτιωμένη απόδοση κωδικοποίησης σε σύγκριση με το main profile.
- **Main still picture profile:** Το main still picture profile επιτρέπει την κωδικοποίηση μιας ενιαίας φωτογραφίας με τους ίδιους περιορισμούς με το main profile. Ως υποσύνολο του main profile, το main still picture profile επιτρέπει ένα βάθος 8 bits ανά δείγμα με δειγματοληψία χρωμάτων 4:2:0.

Κεφάλαιο 2: Αλγόριθμοι για Παραλληλοποίηση HEVC με χρήση slices

2.1 Στόχοι

Το νέο πρότυπο HEVC κωδικοποίησης βίντεο έχει σχεδιαστεί για να προσφέρει αυξημένες αναλογίες συμπίεσης σε σύγκριση με τον προκάτοχό του H.264 / AVC, ιδιαίτερα για βίντεο εξαιρετικά υψηλής ανάλυσης όπως 4K. Η επεξεργασία τέτοιων ροών απαιτεί σημαντική υπολογιστική ισχύ η οποία μπορεί να υλοποιηθεί μόνο με πολλούς πυρήνες CPU και / ή GPU. Το πόσο σημαντική είναι η κλιμάκωση της διαδικασίας συμπίεσης βίντεο γίνεται αντιληπτό όταν αναλογιστούμε τα οικονομικά μεγέθη των εταιρειών που δραστηριοποιούνται στον τομέα της διακωδικοποίησης (video transcoding), πχ., YouTube. Για το λόγο αυτό στη σχετική βιβλιογραφία προτείνονται διάφορες τεχνικές διαμοιρασμού του σχετικού φόρτου σε περιβάλλον Υπολογιστικού Νέφους, πχ., [8] και [43], ενώ η διαστασιοποίηση του προβλήματος τόσο σε υπολογιστικούς όσο και σε οικονομικούς όρους μελετάται στις [18] και [16] αντίστοιχα. Τέλος, ιδιαίτερο ενδιαφέρον σε περιβάλλον Νέφους έχουν οι Scalable επεκτάσεις του HEVC [32] καθώς απαλείφεται με αυτόν τον τρόπο η ανάγκη να υπάρχουν ξεχωριστά αρχεία για τα διαφορετικά επίπεδα ποιότητας και ανάλυσης (αφού στην ουσία πολυπλέκονται όλα τα επιθυμητά επίπεδα σε μία ενιαία ροή), αν και η εύρεση των κοινών παραμέτρων όλων των ροών είναι ιδιαίτερος δύσκολο πρόβλημα βελτιστοποίησης [17].

Σε αυτή την εργασία εστιάζουμε την προσοχή μας στην επιτάχυνση της διαδικασίας συμπίεσης μίας ροής βίντεο χρησιμοποιώντας την παραλληλοποίηση σε επίπεδο πυρήνων CPU με χρήση slices. Συγκεκριμένα, προτείνουμε αλγόριθμους που μπορούν να εξισορροπήσουν αποτελεσματικά το φορτίο του πυρήνα προσαρμόζοντας τα μεγέθη των τεμαχίων μεταξύ των frames. Αξιολογούμε τους προτεινόμενους αλγορίθμους για την περίπτωση της ιεραρχικής κωδικοποίησης βίντεο και τις συγκρίνουμε με άλλες εναλλακτικές λύσεις της βιβλιογραφίας. Τα αποτελέσματα καταδεικνύουν ότι οι αλγόριθμοι που εκμεταλλεύονται την ιεραρχική δομή GOP (Group of Pictures) μπορούν να μειώσουν τις ανισορροπίες φόρτισης κατά λιγότερο

από 15%, ειδικά όταν χρησιμοποιείται η κωδικοποίηση χαμηλής καθυστέρησης, ξεπερνώντας με μεγάλο περιθώριο την απόδοση των υπαρχουσών εναλλακτικών λύσεων. Δείχνεται επίσης ότι μπορεί να επιτευχθεί βελτίωση ταχύτητας μέχρι 25% σε σύγκριση με την ομοιόμορφη ταξινόμηση στατικού μεγέθους slice.

2.2 Συναφής έρευνα στην παράλληλη κωδικοποίηση βίντεο

Η αξιολόγηση της απόδοσης του HEVC και η σύγκριση με το H.264 / AVC έγινε στα [4], [9] και [30] για να αναφέρουμε μερικά. Συνολικά, το συμπέρασμα από αυτές τις εργασίες είναι ότι το HEVC είναι σε θέση να μειώσει το μέγεθος bitstream που παράγεται από το H.264/AVC κατά τουλάχιστον 30%. Παρ' όλα αυτά, για να υλοποιηθεί η δυνατότητα στην πράξη, χρειάζονται αποδοτικές παράλληλες υλοποιήσεις του κωδικοποιητή και του αποκωδικοποιητή [4]. Στην αναφορά [5] τα τρία διαφορετικά εργαλεία παραλληλοποίησης στο HEVC, δηλαδή το wavefront, τα tiles και τα slices συζητούνται με ιδιαίτερο ενδιαφέρον για το πρώτο, ενώ στην αναφορά [6] επικεντρώνεται στην wavefront παραλληλοποίηση, για το κομμάτι της αποκωδικοποίησης.

Η παραλληλοποίηση με tiles αξιολογήθηκε στην αναφορά [25]. Ένα σύστημα κωδικοποίησης με περιοχή ενδιαφέροντος (Region Of Interest) υποκινήθηκε με το οποίο τα μικρότερα tiles αποδίδονται σε ένα ROI που αναμένεται να έχει υψηλό χρόνο κωδικοποίησης (πιθανώς λόγω υψηλής κίνησης). Η αξιολόγηση με τη χρήση προκαθορισμένων κατατμήσεων tiles με βάση το ROI έδειξε την αποτελεσματικότητα της μεθόδου. Στην αναφορά [33] το πλήθος των tiles και η μορφή τους καθορίζεται στην αρχή της ροής βίντεο, με γνώμονα την επίτευξη προκαθορισμένου βαθμού παραλληλίας και επιθυμητής ταχύτητας επεξεργασίας. Ο επανακαθορισμός του μεγέθους των tiles πριν την κωδικοποίηση κάθε frame ώστε να επιτυγχάνεται εξισορρόπηση φόρτου στους επεξεργαστές βρέθηκε στο επίκεντρο πολλών εργασιών. Στις [19] και [34] προτείνεται μία επαναλαμβανόμενη διαδικασία βελτιστοποίησης των ορίων των tiles δοθέντος εκτιμήσεων για το κόστος κωδικοποίησης των CTUs στο επερχόμενο frame. Στην [20] αξιοποιούνται θεωρητικά αποτελέσματα από το πρόβλημα κατάτμησης πίνακα με στόχο τον βέλτιστο επανακαθορισμό των ορίων των tiles. Ο επανακαθορισμός των ορίων των tiles είναι και το αντικείμενο της [37] με τη διαφορά ότι η συνάρτηση βελτιστοποίησης περιλαμβάνει τόσο τη μείωση του χρόνου εκτέλεσης

όσο και την ποιότητα του τελικού βίντεο. Για την περίπτωση που χρησιμοποιούνται περισσότερα tiles από τους διαθέσιμους επεξεργαστές στην [28] προτείνεται αλγόριθμος τόσο επανακαθορισμού των ορίων των tiles όσο και ανάθεσης των tiles στα CPU cores, ενώ στην [23] γίνεται πειραματική εύρεση του αριθμού των tiles που προσφέρει την καλύτερη σχέση ταχύτητας και ποιότητας (υποθέτοντας στατικά όρια).

Οι τεχνικές παραλληλοποίησης SIMD (Single Instruction Multiple Data) εφαρμόστηκαν ευρέως στην κωδικοποίηση βίντεο από την εμφάνιση του MPEG-2 πίσω στη δεκαετία του '90, βλέπε για παράδειγμα [2]. Πιο πρόσφατα παραδείγματα τέτοιων έργων περιλαμβάνουν: [1], [14], [26] και [38] για να αναφέρουμε μερικά. Στο [14] συζητείται μια προσέγγιση που περιλαμβάνει τη χρήση των οδηγιών SIMD στον πυρήνα της CPU για την αποτελεσματικότερη υλοποίηση των περισσότερων στοιχείων κωδικοποίησης βίντεο. Η DCT και η συνάρτηση κόστους παραλληλοποίησης συζητείται (μεταξύ άλλων) στο [1]. Στην [38] παρουσιάζεται μια συνδυασμένη προσέγγιση GPU - πολλαπλών πυρήνων CPU για την εκτίμηση παράλληλης κίνησης, ενώ σε [26] παρέχεται συγκριτική αξιολόγηση μεταξύ της εφαρμογής GPU με CUDA και αντίστοιχων υλοποιήσεων χρησιμοποιώντας MPI και OpenMP.

Η παραλληλοποίηση της διαδικασίας εκτίμησης κίνησης είναι και το αντικείμενο της [41] όπου στόχος είναι η εκμετάλλευση των δυνατοτήτων των multi/many core CPUs. Στην αναφορά [42] εισάγεται ένα πλαίσιο για την ανάλυση των εξαρτήσεων γειτονικών CTU. Τα CTU σχηματίζουν ένα DAG το οποίο στη συνέχεια προγραμματίζεται για παράλληλο υπολογισμό. Διαφορετικοί βαθμοί παραλληλοποίησης συζητούνται ποικίλλοντας από ένα CTU σε ομάδες CTU. Μια παρόμοια προσέγγιση ακολουθείται και στο [44], αλλά για την ενδο-κωδικοποίηση, χρησιμοποιώντας τον κωδικοποιητή x265 ανοιχτού κώδικα [40].

Αν και το επίκεντρο της εργασίας έγκειται στη παραλληλοποίηση σε επίπεδο slices (σε επίπεδο πυρήνα της CPU), οι αλγόριθμοι που παρουσιάζονται στην εργασία μπορούν να ενσωματωθούν σε ένα συνδυασμένο σχήμα που χρησιμοποιεί επίσης λεπτομερή παραλληλισμό για την εκτίμηση της κίνησης (σε επίπεδο πυρήνα GPU).

2.3 Συναφής έρευνα στην παραλληλοποίηση σε επίπεδο slices

Οι προηγούμενες εργασίες του Κεφ. 2.2, διαφέρουν από αυτή την εργασία ως προς τη θεωρούμενη μονάδα εργασίας αλλά και το βάθος παραλληλοποίησης. Πιο

κοντά στην παραλληλοποίηση σε επίπεδο slices είναι οι εργασίες που έγιναν στο H.264/AVC [7], [12], [31] και [45]. Στην [45] εξετάζεται η ανάθεση προσαρμοστικών Macroblock σε slices. Η τεχνική βασίζεται στον υπολογισμό του παρελθοντικού σταθμισμένου μέσου όρου (Weighted Past Average) με συντελεστή 0,5 προκειμένου να εκτιμηθεί το κόστος του Macroblock για το επόμενο frame. Στη συνέχεια, τα macroblock κατανέμονται σε slices έτσι ώστε να ελαχιστοποιούνται οι διαφορές στο συνολικό κόστος. Ο αλγόριθμος TSLB-C δανείζει την ιδέα της χρήσης του πραγματικού χρόνου κωδικοποίησης του Macroblock (CTU σε HEVC) ως εκτιμητή χωρίς όμως να χρησιμοποιεί WPA.

Στο [7] το πρόβλημα της εξισορρόπησης των slices αντιμετωπίστηκε με χρήση περισσότερων slices από τους υφιστάμενους πυρήνες σε μια προσπάθεια να μειωθεί η λεπτομερής παραλληλοποίηση, επιτυγχάνοντας έτσι καλύτερη ισορροπία. Ο δυναμικός ορισμός του αριθμού των slices υπερβαίνει το πεδίο αυτής της εργασίας. Στο [12] προτείνεται ένας αλγόριθμος που προσαρμόζει το μέγεθος των slices για τη βελτίωση της ισορροπίας φόρτωσης. Το εγχείρημα χρησιμοποιεί ένα βήμα προεπεξεργασίας για γρήγορη εκτίμηση κίνησης και στη συνέχεια το εφαρμόζει στα Macroblocks ανάλογα με τα αποτελέσματα. Ως εκ τούτου, δεν έχει άμεση εφαρμογή στο HEVC. Στην [31] εφαρμόζεται ιεραρχική παραλληλοποίηση σε δύο επίπεδα. Σε πρώτο επίπεδο, διανέμονται διαφορετικά GOP (Group of Pictures) σε κόμβους υπολογισμού. Κάθε frame ενός GOP κωδικοποιείται χρησιμοποιώντας παραλληλοποίηση επιπέδου slice. Η προσαρμογή του μεγέθους του slice δεν λαμβάνεται υπόψη.

Όσον αφορά το HEVC, οι συγγραφείς του [29] αξιολόγησαν την παραλληλοποίηση με βάση τα slices σε διαφορετικά σενάρια κωδικοποίησης. Ωστόσο, η εξισορρόπηση φόρτου δεν ελήφθη υπόψη. Για την περίπτωση που τα slices ορίζονται ως άθροισμα από tiles, στην [21] προτείνονται αλγόριθμοι επανακαθορισμού ορίων των tiles και συνακολούθως των slices ώστε να επιτυγχάνεται τόσο η μείωση του χρόνου συμπίεσης όσο και η εξισορρόπηση του μεγέθους των slices με στόχο την υποβοήθηση της διαδικασίας μετάδοσης του βίντεο. Στην παρούσα διπλωματική εργασία περιορίζουμε τη μελέτη μας στην περίπτωση που τα slices ορίζονται ως σύνολο CTUs και όχι tiles.

Ίσως, το πιο κοντινό στην εργασία μας είναι το [1] όπου παρουσιάζεται παραλληλοποίηση με βάση SIMD όπως επίσης και παραλληλοποίηση σε επίπεδο slice με την εφαρμογή προσαρμοστικών slices. Στο πείραμα αυτής της εργασίας συγκρίνουμε επίσης την απόδοση των αλγορίθμων μας με εκείνη της προαναφερθείσας

εργασίας. Μερικά προκαταρκτικά αποτελέσματα αυτής της εργασίας που περιέχουν μόνο ένα υποσύνολο των αλγορίθμων και πειραματικών σεναρίων που εξετάστηκαν εδώ εμφανίστηκαν στο [15].

2.4 Αλγόριθμοι βιβλιογραφίας

Σε αυτήν την ενότητα περιγράφουμε τους αλγορίθμους που χρησιμοποιούνται στην πειραματική αξιολόγηση του Κεφ. 3. Ξεκινάμε με τους αλγόριθμους που δεν συμπεριλαμβάνουν τη δομή του GOP και συνεχίζουμε με αυτούς που το κάνουν.

Στατική ομοιόμορφη εκχώρηση (Static)

Σε αυτό το σχήμα τα CTUs κατανέμονται ομοιόμορφα σε slices και η κατανομή παραμένει σταθερή για όλα τα frame. Αυτή η μέθοδος χρησιμοποιείται ως δείκτης απόδοσης και είναι ισοδύναμη με εκείνη που αξιολογείται στο [29].

Αλγόριθμος βασισμένος στο βάρος (Weight)

Ο αλγόριθμος που προτείνεται στο [1] βασίζεται στην εφαρμογή ενός κόστους βάρους σε κάθε CU ανάλογα με το εάν το συνεγκατεστημένο CU στο προηγούμενο frame κωδικοποιήθηκε ως Skip, Inter ή Intra και το αντίστοιχο βάθος του quadtree. Ο Πίνακας 1 αναπαράγει τα βάρη για ευκολία.

Πίνακας 1: Βάρη ανάθεσης στον αλγόριθμο Weight.

Μέγεθος CU	Skip	Inter	Intra
64×64	109	760	52
32×32	42	280	16
16×16	9	71	3
8×8	2	19	1

Ο αλγόριθμος υπολογίζει κάθε CTU weight ως άθροισμα των αντίστοιχων CU weight και το slice weight ως άθροιση των σχετικών CTU weight. Στη συνέχεια αναθέτει τις CTU σε κάθε slice έτσι ώστε να εξισορροπούνται τα βάρη των slices.

Παρελθοντικός σταθμισμένος μέσος όρος (WPA)

Ο αλγόριθμος που προτείνεται στο [45] λειτουργεί ως εξής. Χρησιμοποιεί τον πραγματικό χρόνο που χρειάστηκε για να κωδικοποιηθεί κάθε CTU στο προηγούμενο frame και προβλέπει τον εκτιμώμενο χρόνο του στο επόμενο frame χρησιμοποιώντας την τεχνική WPA (Weight Past Average) με τον πραγματικό και εκτιμώμενο χρόνο να σταθμίζονται με 0,5 αντίστοιχα.

2.5 Προτεινόμενοι αλγόριθμοι που δε βασίζονται στο GOP

Χρονική εξισορρόπηση φόρτου slice με χρήση του μέσου όρου CTU σε slices (TSLB-Avg)

Το TSLB-Avg λειτουργεί σε επίπεδο slice. Το S_i δηλώνει το i -οστό slice ($0 \leq i \leq S-1$) όπου S είναι ο συνολικός αριθμός slices. Το T_{ij} είναι ο πραγματικός χρόνος λειτουργίας για να συμπιεστεί το S_i στο j -οστό frame και C_{ij} είναι ο συνολικός αριθμός CTUs στο S_i . Το TSLB-Avg διανέμει τα CTU σε slices αναλογικά με τους πραγματικούς χρόνους συμπίεσης των slices (του αντίστοιχου slice στο προηγούμενο frame) ως εξής. Πρώτα για κάθε slice, η διαφορά μεταξύ του χρόνου και του μέσου χρόνου slice υπολογίζεται ως (1).

$$D_{ij} = T_{ij} - \left(\sum_{x=0}^{S-1} T_{xj} / S \right) \quad (1)$$

Αν η διαφορά είναι θετική, το slice πρέπει να εγκαταλείψει CTUs για να πλησιάσει το χρονικό μέσο όρο, διαφορετικά θα πρέπει να πάρει περισσότερα. Ο αριθμός των CTU που πρέπει να αφεθούν ή να αποκτηθούν δίνεται από:

$$A_{ij} = \begin{cases} D_{ij} C_{ij} / T_{ij} & , D_{ij} > 0 \\ D_{ij} C_{(i+1)j} / T_{(i+1)j} & , D_{ij} < 0 \end{cases} \quad (2)$$

Στη (2) εάν το S_i πρέπει να αφήσει μερικά από τα CTU του, τότε ο μέσος χρόνος CTU στο S_i (δηλ. T_{ij}/C_{ij}) θα πρέπει να χρησιμοποιηθεί για να υπολογίσει πόσα CTU πρέπει να αφήσει ώστε το S_i να έχει υπολογιστικό χρόνο ίσο με τον μέσο όρο όλων των slices. Διαφορετικά, εάν θα πρέπει να πάρει CTU, αυτά τα CTU θα προέλθουν από το

επόμενο slice, οπότε, χρησιμοποιείται ο μέσος χρόνος CTU στο slice S_{i+1} . Ο αριθμός των CTU που χρειάζεται να αφήσει ή να αποκτήσει ορίζεται ως $\lfloor |A_{ij}| \rfloor$.



Εικόνα 6: Παράδειγμα διαμοίρασμού frame σε 4 slices (βίντεο: Bosphorus, frame: 5).

Όταν το S_i αφήνει $\lfloor |A_{ij}| \rfloor$ CTUs (δηλ., $D_{ij} > 0$ στο (2)), αυτά τα CTUs θα ανατεθούν στο επόμενο slice S_{i+1} . Αυτό θα πρέπει να ληφθεί υπόψη στον υπολογισμό της (1) για S_{i+1} με την προσθήκη του κόστους που προκύπτει από τα $\lfloor |A_{ij}| \rfloor$ CTU που κληρονομούνται από το S_i . Μια παρόμοια παρατήρηση ισχύει όταν το S_i πρέπει να αποκτήσει CTU που ανήκουν στο S_{i+1} . Οι παραπάνω παρατηρήσεις ενσωματώνονται στις (3) και (4).

$$D'_{ij} = \begin{cases} D_{ij} & , i = 0 \\ D_{ij} + \frac{\lfloor |A'_{(i-1)j}| \rfloor T_{(i-1)j}}{C_{(i-1)j}} & , i > 0 \wedge A_{(i-1)j} > 0 \\ D_{ij} - \frac{\lfloor |A'_{(i-1)j}| \rfloor T_{ij}}{C_{ij}} & , i > 0 \wedge A_{(i-1)j} < 0 \end{cases} \quad (3)$$

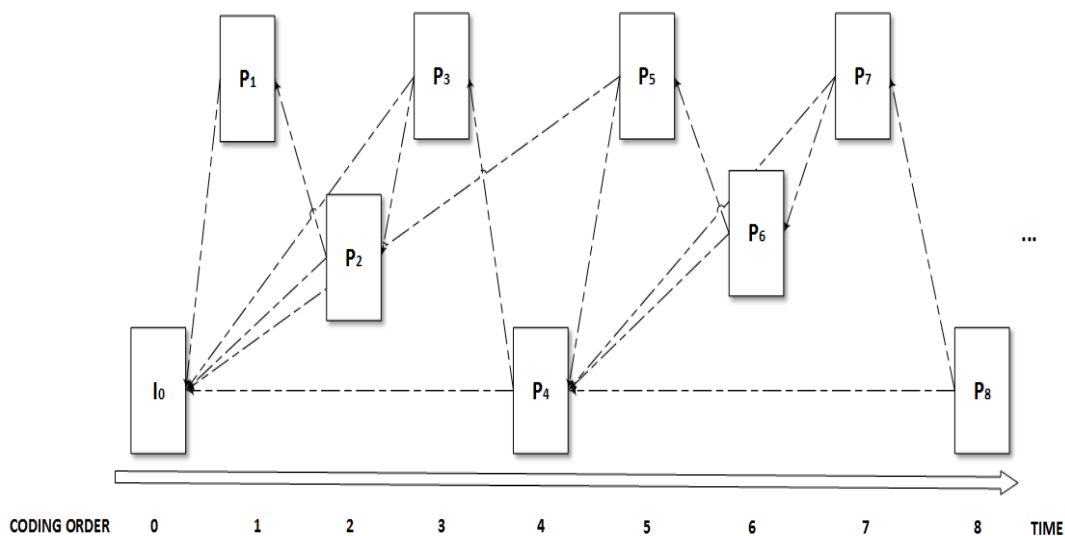
$$A'_{ij} = \begin{cases} D'_{ij} C_{ij} / T_{ij} & , D'_{ij} > 0 \\ D'_{ij} C_{(i+1)j} / T_{(i+1)j} & , D'_{ij} < 0 \end{cases} \quad (4)$$

Ξεκινώντας από το πρώτο slice (S_0) και συνεχίζοντας μέχρι το S_{S-2} με επαναληπτικό τρόπο, ο αλγόριθμος χρησιμοποιεί τις (1), (3) και (4) για να υπολογίσει πόσα CTU πρέπει να πάρει ή να αφήσει ένα slice. Το τελευταίο slice S_{S-1} λαμβάνει τα

υπόλοιπα CTU που δεν έχουν επιλεγεί. Για να έχουμε μια οπτική αναπαράσταση του τρόπου εκτέλεσης του TSLB-Avg, η Εικόνα 6 δείχνει την αντιστοίχιση μεγέθους 4 slices στο 5ο frame της ακολουθίας Bosphorus [13]. Παρατηρήστε ότι το τρίτο slice που περιλαμβάνει το μεγαλύτερο μέρος της κίνησης του σκάφους είναι μικρότερο σε σύγκριση με τα υπόλοιπα.

Χρονική εξισορρόπηση φόρτου slice με χρήση των πραγματικών χρόνων CTU (TSLB-C)

Το TSLB-C λειτουργεί με παρόμοιο τρόπο με το TSLB-Avg. Η διαφορά είναι ότι αντί των μέσων χρόνων CTU στις (3) και (4) χρησιμοποιεί τους πραγματικούς χρόνους κωδικοποίησης CTU.



Εικόνα 7: Παράδειγμα ιεραρχικής δομής 2 GOP από P frames (Low Delay).

2.6 Προτεινόμενοι αλγόριθμοι βασισμένοι στο GOP

Χρονική εξισορρόπηση φόρτου slice χρησιμοποιώντας τους μέσους χρόνους CTU σε slices και τη δομή του GOP (TSLB*)

Στόχος των αλγορίθμων της κατηγορίας είναι να εκμεταλλευτούν την ιεραρχική δομή των GOP (στην περίπτωση που χρησιμοποιούνται ιεραρχικά GOP όπως προτάθηκαν στην [11]) με στόχο την καλύτερη πρόβλεψη του χρόνου συμπίεσης των CTU.

Η σκέψη πίσω από το TSLB* είναι ότι η χρονική πολυπλοκότητα των frame που ανήκουν στο βασικό στρώμα όπως τα P4 και P8 στην Εικόνα 7 θα είναι καλύτερο να προβλεφθούν από το προηγούμενο frame του βασικού στρώματος και όχι από το προηγούμενο frame αριθμητικά. Στο παράδειγμα, αυτό σημαίνει ότι οι χρόνοι των CTU στο P8 θα προβλεφθούν χρησιμοποιώντας το P4 και όχι το P7.

Παρατηρήστε ότι όλοι οι αλγόριθμοι του Κεφ. 2.5 θα χρησιμοποιήσουν αντί του P4 το P7. Μια άλλη αλλαγή που εισάγει ο αλγόριθμος TSLB*, αφορά στην εκτίμηση του frame που ακολουθεί ένα frame βασικού στρώματος. Αντί να χρησιμοποιεί το frame βασικού στρώματος, χρησιμοποιεί το frame που άμεσα προηγείται. Για παράδειγμα, η εκτίμηση του P9 (που δεν φαίνεται στο σχήμα 7) θα γίνει από το P7 αντί του P8. Η διαδικασία ανάθεσης του TSLB* συνοψίζεται και γενικεύεται για αυθαίρετα μεγέθη GOP (ως G) στις ακόλουθες εξισώσεις:

$$E_{ij} = \begin{cases} T_{ij} & , \quad 1 + j \bmod G \in [2, G - 1] \\ C_{ij} T_{i(j-G+1)} / C_{i(j-G+1)} & , \quad 1 + j \bmod G = G \\ C_{ij} T_{i(j-1)} / C_{i(j-1)} & , \quad 1 + j \bmod G = 1 \end{cases} \quad (5)$$

$$D_{ij} = E_{ij} - (\sum_{x=0}^{S-1} E_{xj} / S) \quad (6)$$

$$D'_{ij} = \begin{cases} D_{ij} & , i = 0 \\ D_{ij} + \frac{\lfloor |A'_{(i-1)j}| \rfloor E_{(i-1)j}}{C_{(i-1)j}} & , i > 0 \wedge A_{(i-1)j} > 0 \\ D_{ij} - \frac{\lfloor |A'_{(i-1)j}| \rfloor E_{ij}}{C_{ij}} & , i > 0 \wedge A_{(i-1)j} < 0 \end{cases} \quad (7)$$

$$A'_{ij} = \begin{cases} D'_{ij} C_{ij} / E_{ij} & , D'_{ij} > 0 \\ D'_{ij} C_{(i+1)j} / E_{(i+1)j} & , D'_{ij} < 0 \end{cases} \quad (8)$$

Ο υπόλοιπος αλγόριθμος είναι παρόμοιος με τον TSLB-Avg, δηλαδή σε κάθε frame j , ο TSLB* ξεκινά τον υπολογισμό της εκχώρησης από το S_0 χρησιμοποιώντας τις (5)-(8) προσθέτοντας ή αφαιρώντας $\lfloor |A'_{ij}| \rfloor$ CTU στην τρέχουσα αντιστοίχιση και προχωρά μέχρι το S_{S-2} με έναν επαναληπτικό τρόπο. Τα CTU που δεν έχουν εκχωρηθεί κατανέμονται στο S_{S-1} . Κατά την εφαρμογή του αλγόριθμου επιλέξαμε να χρησιμοποιήσουμε το TSLB-Avg για το πρώτο GOP και τις εκτιμήσεις του TSLB* από το δεύτερο GOP και μετά.

Βελτιωμένη εξισορρόπηση φόρτου με βάση το χρόνο χρησιμοποιώντας τις μέσες τιμές CTU σε slices και δομή GOP (TSLB+)

Ο TSLB+ βελτιώνει τον TSLB*, ταυτόχρονα κάνοντας το σύστημα εκτίμησης πιο απλό. Στο TSLB* ορισμένα από τα frame υπολογίζονται από frame που ανήκουν στο ίδιο GOP ενώ άλλα από το προηγούμενο GOP. Ο TSLB+ υπολογίζει τον εκτιμώμενο φόρτο των slices χρησιμοποιώντας το αντίστοιχο frame του προηγούμενου GOP. Λαμβάνοντας υπόψη τη δομή GOP της Εικόνας 7, ο TSLB+ θα εκτιμήσει τα πλαίσια P5, P6, P7 και P8, από τα P1, P2, P3 και P4 αντίστοιχα. Ο υπόλοιπος αλγόριθμος είναι ταυτόσημος με τον TSLB*, δηλαδή οι μέσοι χρόνοι CTU χρησιμοποιούνται στους υπολογισμούς. Αν και είναι απλούστερος σε σύγκριση με τον TSLB*, ο TSLB+ επιτυγχάνει καλύτερες επιδόσεις όπως φαίνεται στα πειράματα.

Κεφάλαιο 3: Πειράματα

3.1 Πλαίσιο διεξαγωγής πειραμάτων

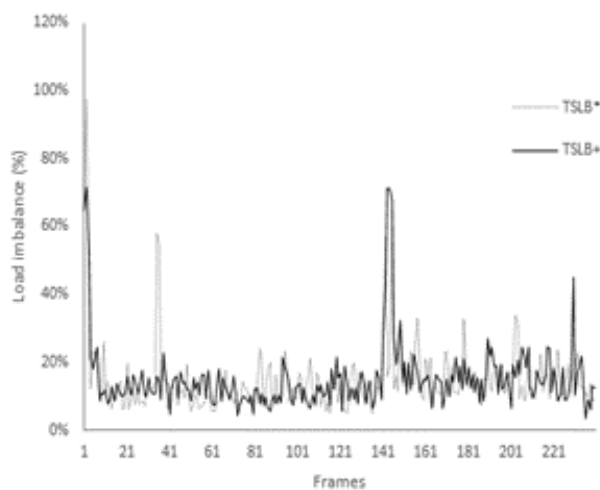
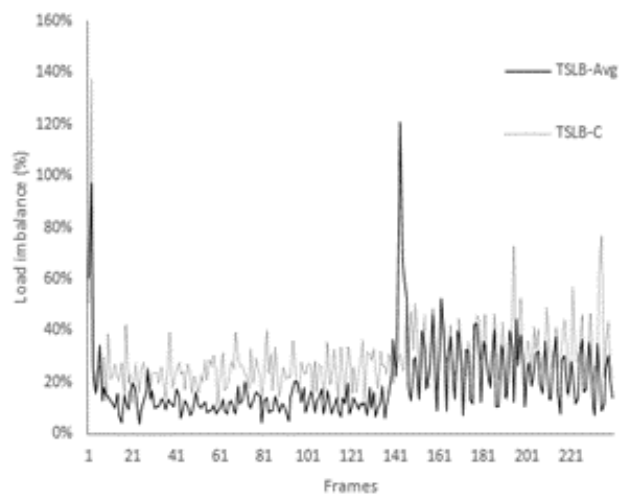
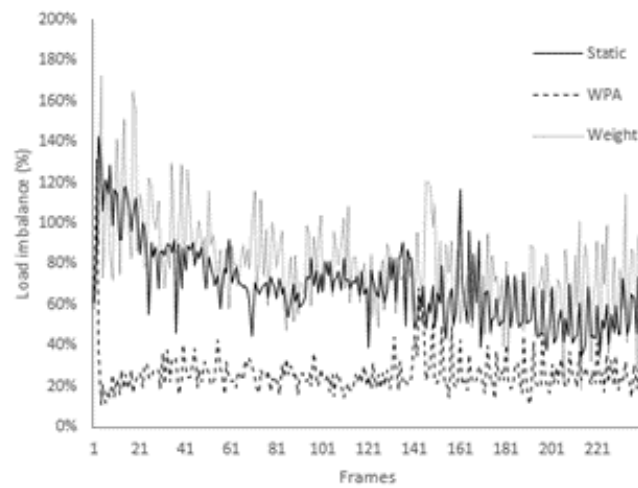
Χρησιμοποιήθηκε το λογισμικό HM 16.7 [10] που είναι ο κωδικοποιητής αναφοράς για το HEVC και υλοποιήθηκαν η παραλληλοποίηση σε επίπεδο slices και οι αλγόριθμοι του Κεφ. 2 με χρήση του OpenMP [27]. Πραγματοποιήθηκαν πειράματα σε έναν Linux server με δύο 6-πύρηνους επεξεργαστές Intel Core Xeon E5-2630 στα 2,3 GHz με χρήση hyper threading. Χρησιμοποιήθηκαν τρεις αλληλουχίες βίντεο (συνοψισμένες στον Πίνακα 2) που αντιστοιχούν σε αναλύσεις FullHD, 2K και 4K. Προκειμένου να εξοικονομηθεί χρόνος στα πειράματα χρησιμοποιήθηκαν τα πρώτα 200 frame της ακολουθίας του Bosphorus αντί της πλήρους. Το QP ορίστηκε σε 32, το βάθος των bit ήταν 8, το μέγεθος CTU 64×64 , το μέγιστο βάθος διαμέρισης CTU ορίστηκε στο 4 και ο τρόπος αναζήτησης στο TZ. Τα πειράματα που διεξήχθησαν αφορούσαν τόσο σε Low Delay όσο και σε Random Access μεθόδους κωδικοποίησης όπως αυτές περιγράφονται στο [3].

Πίνακας 2: Χαρακτηριστικά των βίντεο που χρησιμοποιήθηκαν στα πειράματα.

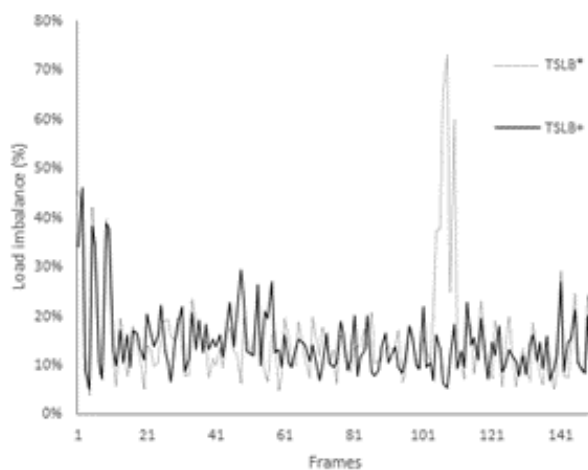
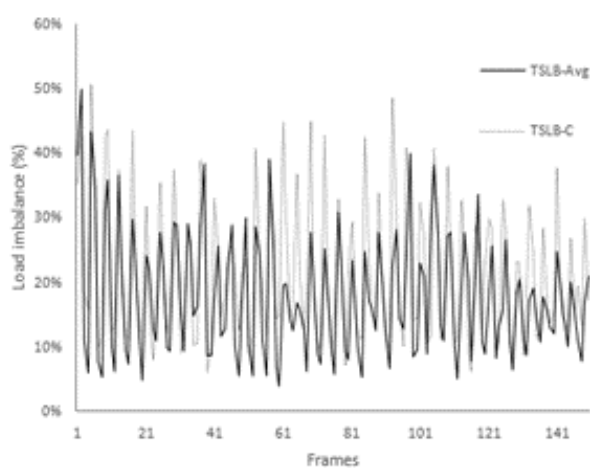
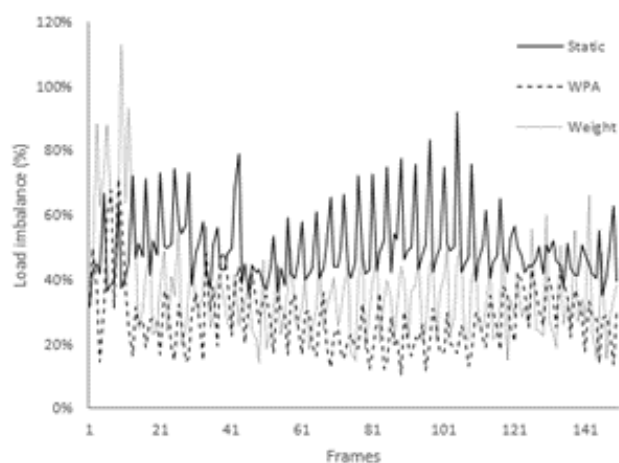
Όνομα	Ανάλυση	Frames ανά sec (fps)	Πλήθος frames	CTUs ανά frame
Bosphorus	3840×2160	120	200/600	2040
Traffic	2560×1600	30	150	1000
Kimono	1920×1080	24	240	510

Σε όλα τα πειράματα, κάθε slice κωδικοποιήθηκε από ένα ξεχωριστό thread στη βάση του ένα προς ένα. Καταγράψαμε τον χρόνο κωδικοποίησης κάθε frame, την ποιότητά του σαν PSNR (Peak Signal to Noise Ratio), το συμπιεσμένο μέγεθος (bit rate), καθώς και την απόκλιση εξισορρόπησης φόρτου μεταξύ των threads ως το ακόλουθο ποσοστό:

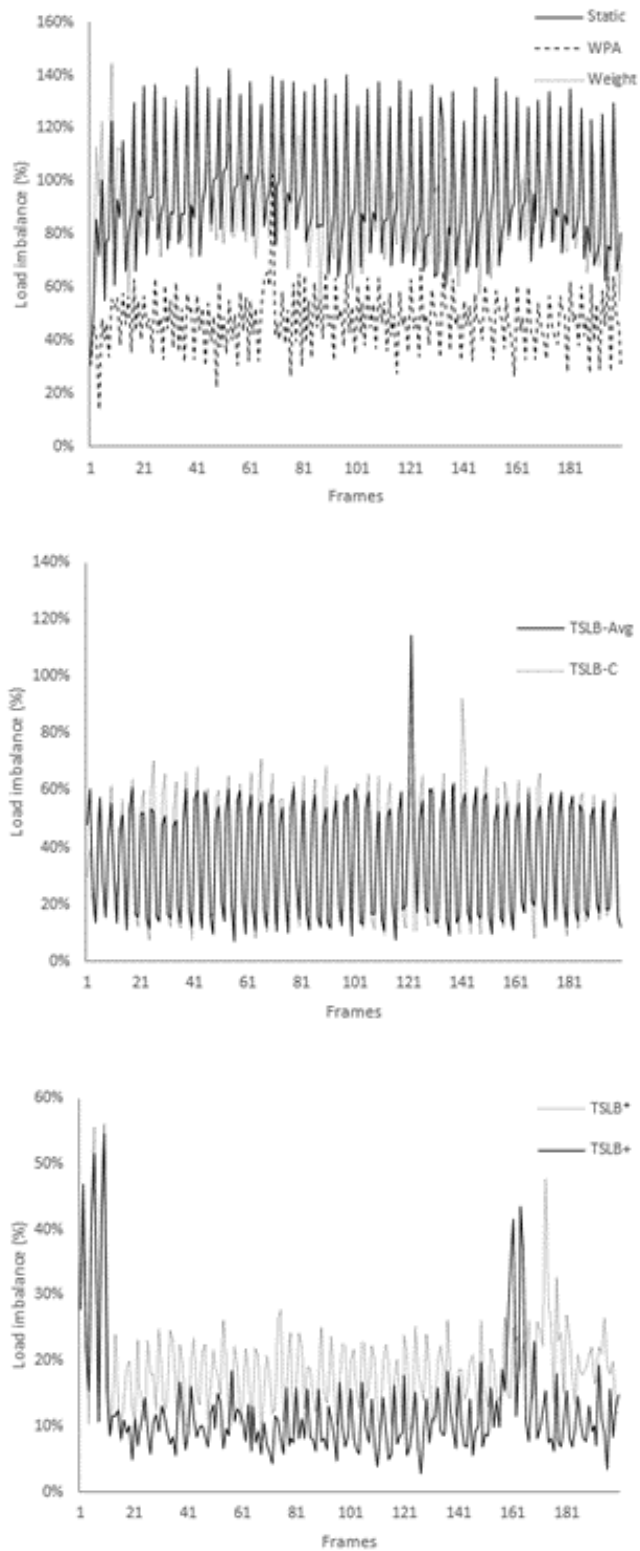
$$100 \text{ (ΜέγιστοςΧρόνοςThread-ΕλάχιστοςΧρόνοςThread) / ΕλάχιστοςΧρόνοςThread}$$



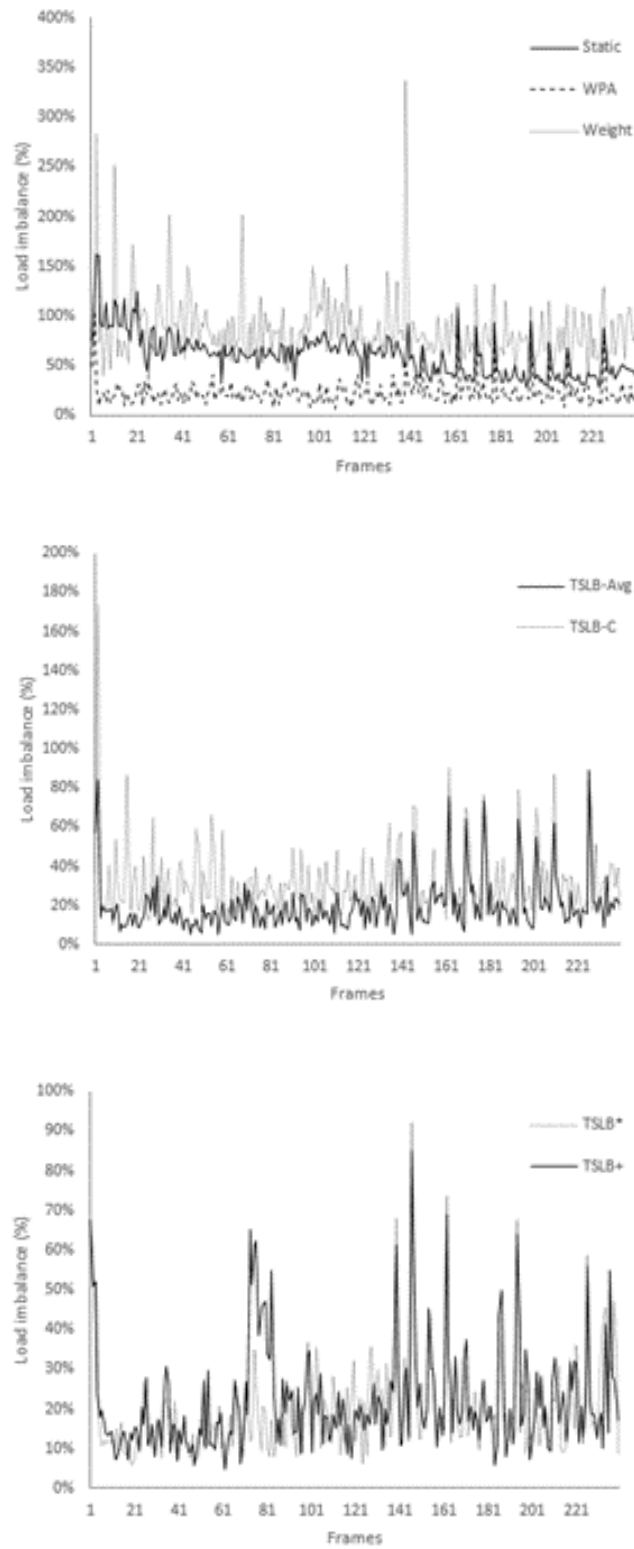
Εικόνα 8: Απόκλιση εξισορρόπησης φόρτου (Low Delay, Kimono, 12 slices).



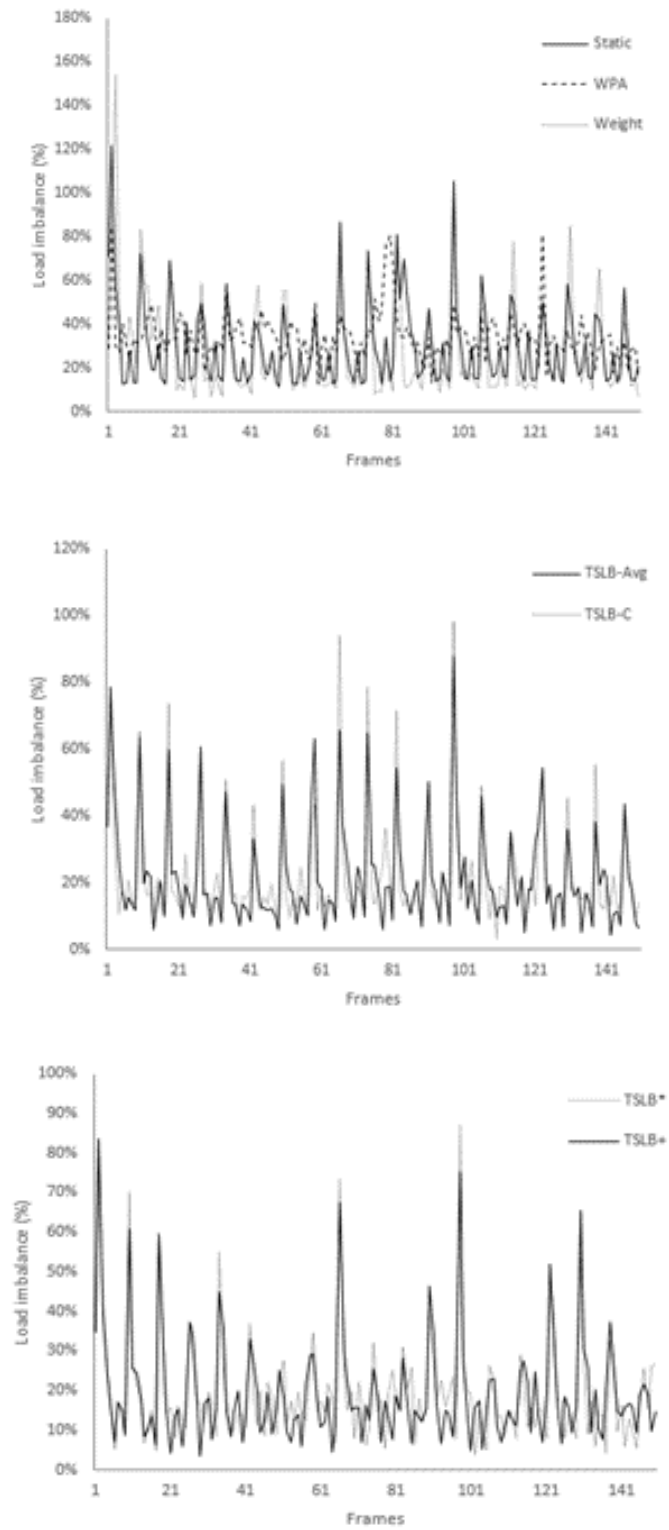
Εικόνα 9: Απόκλιση εξισορρόπησης φόρτου (Low Delay, Traffic, 12 slices).



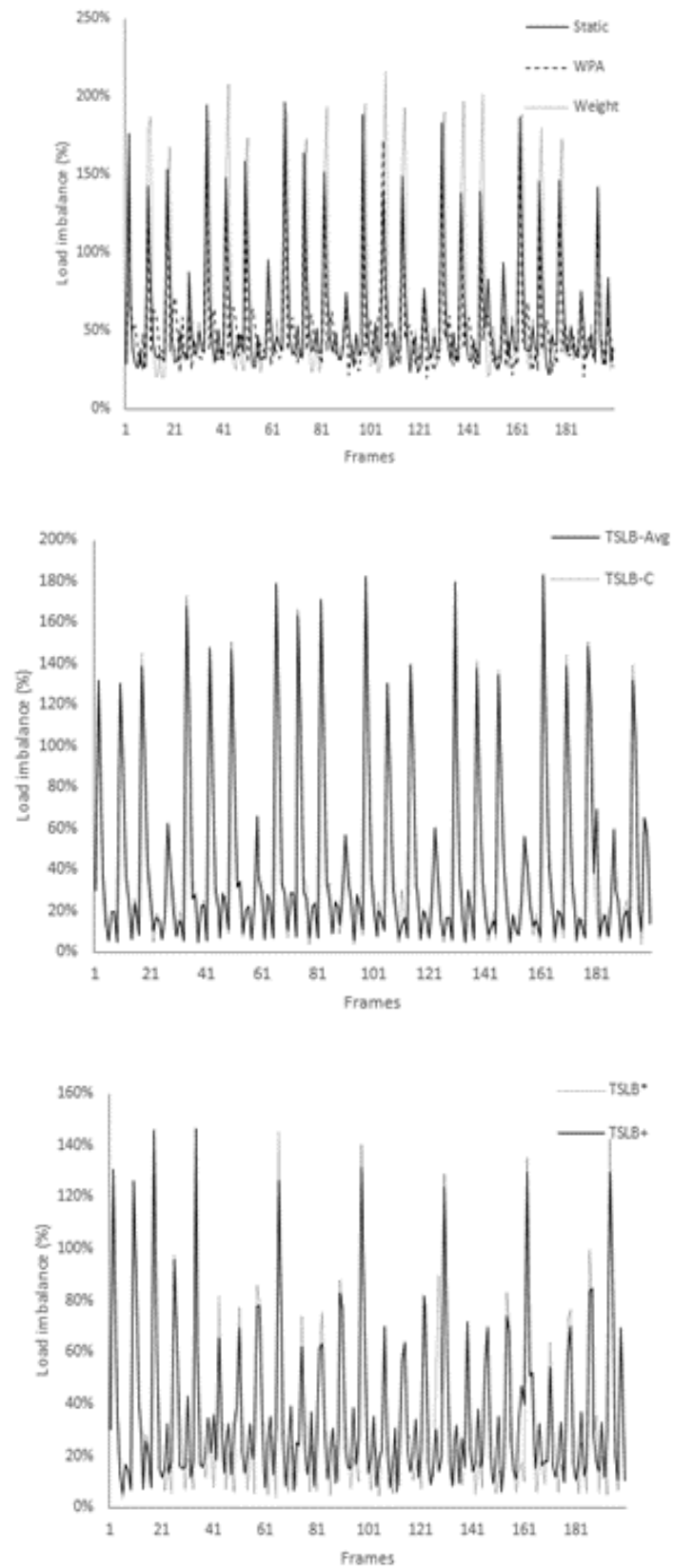
Εικόνα 10: Απόκλιση εξισορρόπησης φόρτου (Low Delay, Bosphorus, 12 slices).



Εικόνα 11: Απόκλιση εξισορρόπησης φόρτου (Random Access, Kimono, 12 slices).



Εικόνα 12: Απόκλιση εξισορρόπησης φόρτου (Random Access, Traffic, 12 slices).



Εικόνα 13: Απόκλιση εξισορρόπησης φόρτου (Random Access, Bosphorus, 12 slices).

3.2 Αξιολόγηση της απόκλισης εξισορρόπησης φόρτου

Αρχικά αξιολογήθηκε η απόδοση των αλγορίθμων χρησιμοποιώντας δομή LowDelay-P GOP. Αποτελέσματα για τις τρεις ακολουθίες και 12 slices φαίνονται στις Εικόνες 8, 9 και 10. Για λόγους αναγνωσιμότητας, κάθε εικόνα χωρίζεται σε τρία μέρη: (α) το πρώτο μέρος απεικονίζει την απόδοση των αλγορίθμων του Κεφ. 2.4, (β) το δεύτερο μέρος τους αλγορίθμους του Κεφ. 2.5 και (γ) το τρίτο μέρος την απόδοση των αλγορίθμων του Κεφ. 2.6. Αρχικά παρατηρούμε ότι όλοι οι αλγόριθμοι παρουσιάζουν περιοδικές κορυφώσεις στην απόκλιση, οι οποίες είναι ιδιαίτερα εμφανείς στις γραφικές παραστάσεις (α) και (β). Αναλύοντας περισσότερο τα αποτελέσματα, μπορέσαμε να εντοπίσουμε ότι η περίοδος ισούται με το μέγεθος του GOP. Αυτό ουσιαστικά υπαγορεύει την αναγκαιότητα αλγορίθμων που βασίζονται στο GOP, όπως ο TSLB* και TSLB+.

Όσον αφορά στην καθαυτή απόδοση των αλγορίθμων, μπορούμε να αναφέρουμε ότι το WPA είναι καλύτερο σε σύγκριση με το Static και τις προσεγγίσεις που βασίζονται σε στατική ανάθεση βαρών (Weight), το TSLB-Avg είναι οριακά ανώτερο σε σύγκριση με το TSLB-C και ανώτερο σε σύγκριση με το WPA (παρατηρήστε τις διαφορές στην κλίμακα άξονα y). Τέλος, οι αλγόριθμοι που βασίζονται στο GOP υπερέχουν όλων των ομολόγων τους. Μεταξύ των δύο επιλογών ο TSLB+ φαίνεται να είναι μια καλύτερη γενικότερα επιλογή. Τέλος, θα θέλαμε να σημειώσουμε ότι στην ακολουθία Kimono υπάρχει μια αλλαγή σκηνής στο frame 141. Παρά το γεγονός ότι το TSLB+ βιώνει ξαφνική κορύφωση, ήταν σε θέση να προσαρμόσει γρήγορα την εξισορρόπηση των slices σχεδόν μέσα σε διάστημα ενός GOP.

Στη συνέχεια μετρήθηκε η απόκλιση φόρτου με την δομή Random Access GOP. Τα αποτελέσματα φαίνονται στις Εικόνες 11, 12 και 13, χρησιμοποιώντας παρόμοιο με τον παραπάνω τρόπο προβολής των αποτελεσμάτων. Συγκρίνοντας την απόδοση σε RA έναντι LowDelay-P μπορούμε να πούμε ότι όλοι οι αλγόριθμοι (TSLB* και TSLB+ επίσης) αντιμετωπίζουν μεγαλύτερες κορυφώσεις (ιδιαίτερα στις περιπτώσεις Traffic και Bosphorus που έχουν μεγαλύτερη κίνηση). Αυτές οι κορυφώσεις φαίνεται να είναι περιοδικές, όμοια όπως και στο LowDelay-P, αλλά με μεγαλύτερη περίοδο. Με την ανάλυση των αποτελεσμάτων μπορούσαμε να εντοπίσουμε ότι η αιτία αυτών των αιχμών είναι το γεγονός ότι στο RA, κάθε 32 frame εισάγεται ένα I frame. Και εδώ, οι TSLB* και TSLB+ είναι οι νικητές αλλά με μικρότερο χάσμα απόδοσης σε σύγκριση με την περίπτωση LowDelay.

3.3 Αξιολόγηση υπολογιστικού χρόνου

Στη συνέχεια παρουσιάζονται τα αποτελέσματα από πειράματα με τους ακόλουθους αριθμούς slices/threads: 2, 4, 8, 12 και 24. Να σημειωθεί ότι όπως αναφέρθηκε στο Κεφ. 3.1, υπήρχαν 12 υπολογιστικοί πυρήνες στον server στον οποίο διεξήχθησαν τα πειράματα. Παρόλα αυτά, θέλαμε να δοκιμάσουμε πώς θα ανταποκριθούν οι αλγόριθμοι όταν οι διαθέσιμοι πυρήνες είναι λιγότεροι από τα slices. Οι Πίνακες 3 και 4 συνοψίζουν τις σχετικές επιταχύνσεις που επιτυγχάνονται από κάθε αλγόριθμο για το LowDelay και RandomAccess. Η επιτάχυνση μετρήθηκε ως ο λόγος του χρόνου της σειριακής εκτέλεσης με 1 thread προς τον παράλληλο χρόνο. Ιδεατά η μέγιστη επιτάχυνση που μπορεί να επιτευχθεί ισούται με το πλήθος των πυρήνων αν και άλλοι λόγοι (αρχιτεκτονική μνήμης) μπορεί να οδηγήσουν σε μεγαλύτερες επιταχύνσεις (κάτι που παρατηρήθηκε σε κάποια από τα πειράματά μας), περίπτωση η οποία αναφέρεται στη βιβλιογραφία ως *superlinear speedup*. Ακολουθούν οι πίνακες. Οι γκρι ενδείξεις υποδεικνύουν τον αλγόριθμο που πετυχαίνει την καλύτερη επίδοση στο συγκεκριμένο πειραματικό σενάριο.

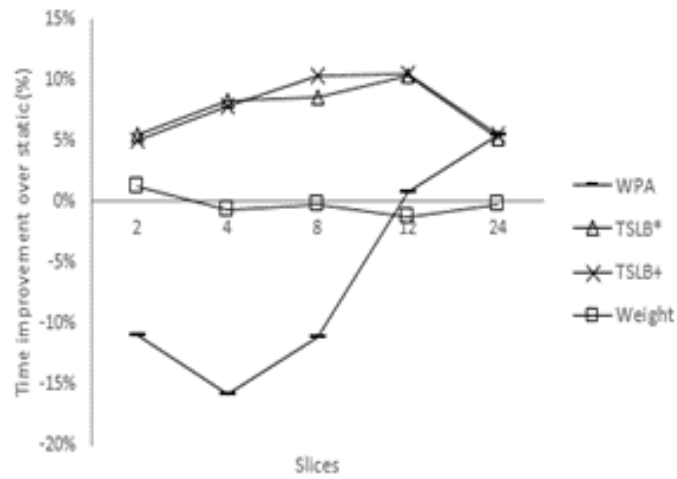
Πίνακας 3: Επιταχύνσεις (speedup) στο LowDelay σενάριο.

		2	4	8	12	24
Bosphorus	Static	1.74	3.35	5.83	8.09	10.44
	TSLB-Avg	1.92	3.67	6.90	10.03	12.16
	TSLB-C	1.93	3.66	6.88	9.90	12.15
	WPA	1.69	3.12	6.19	9.30	12.19
	TSLB*	1.94	3.76	7.32	10.63	12.45
	TSLB+	1.97	3.87	7.57	11.01	12.34
	Weight	1.74	3.29	5.80	8.14	10.63
Traffic	Static	1.94	3.43	6.45	9.26	11.33
	TSLB-Avg	1.92	3.71	7.24	10.41	11.95
	TSLB-C	1.93	3.72	7.18	10.52	11.94
	WPA	1.72	3.27	6.26	9.62	12.02
	TSLB*	1.95	3.79	7.36	10.48	11.71
	TSLB+	1.94	3.82	7.31	10.51	11.44
	Weight	1.91	3.57	6.85	9.89	11.64
Kimono	Static	1.85	3.56	6.76	9.69	11.46
	TSLB-Avg	1.96	3.81	7.35	10.67	12.05
	TSLB-C	1.95	3.79	7.35	10.64	11.43
	WPA	1.67	3.07	6.09	9.77	12.12
	TSLB*	1.96	3.88	7.39	10.81	12.10
	TSLB+	1.95	3.86	7.54	10.82	12.13
	Weight	1.88	3.53	6.74	9.57	11.44

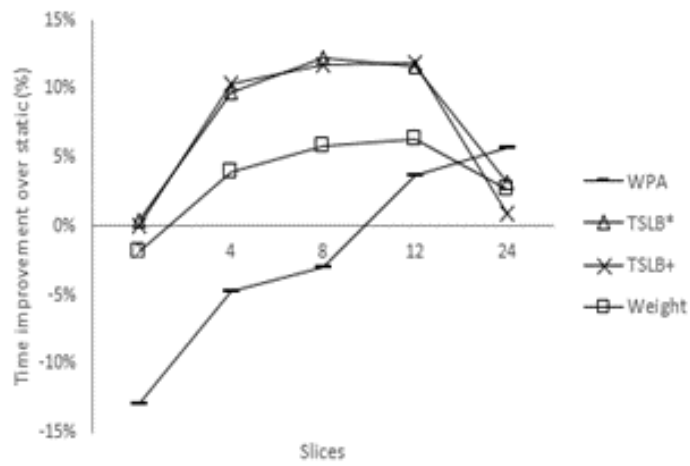
Πίνακας 4: Επιταχύνσεις (speedup) στο Random Access σενάριο.

		2	4	8	12	24
Bosphorus	Static	1.88	3.57	6.79	9.71	11.62
	TSLB-Avg	1.85	3.61	6.86	10.01	12.13
	TSLB-C	1.85	3.56	6.84	9.98	12.04
	WPA	1.74	3.29	6.41	9.36	12.08
	TSLB*	1.88	3.66	6.98	10.07	12.10
	TSLB+	1.87	3.63	6.94	10.01	12.06
	Weight	1.84	3.51	6.50	9.44	11.50
Traffic	Static	1.95	3.62	6.95	10.01	11.97
	TSLB-Avg	1.94	3.70	7.13	10.37	12.02
	TSLB-C	1.94	3.71	7.17	10.38	12.05
	WPA	1.73	3.21	6.12	9.42	12.32
	TSLB*	1.95	3.78	7.25	10.51	12.09
	TSLB+	1.94	3.75	7.14	10.52	12.09
	Weight	1.90	3.65	7.02	10.22	11.92
Kimono	Static	1.86	3.58	6.84	9.91	11.35
	TSLB-Avg	1.92	3.75	7.30	10.60	11.99
	TSLB-C	1.92	3.77	7.20	10.40	11.33
	WPA	1.70	3.17	6.33	10.21	11.89
	TSLB*	1.93	3.76	7.24	10.57	11.88
	TSLB+	1.93	3.75	7.26	10.37	12.03
	Weight	1.85	3.54	6.73	9.62	12.81

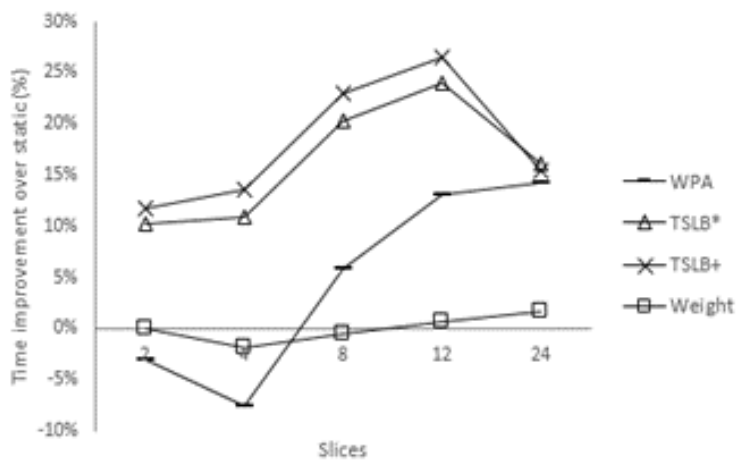
Οι αλγόριθμοι που βασίζονται στο GOP αποδεικνύονται σαφείς νικητές τόσο στην περίπτωση LowDelay όσο και στην περίπτωση RandomAccess (παρατηρούμε ότι ακόμη και στις καταχωρήσεις του Kimono με RA όπου χάνουν, οι διαφορές με το TSLB-Avg και το TSLB-C που επιτυγχάνουν την καλύτερη απόδοση, είναι αμελητέες). Η απόδοση του TSLB+ στην ακολουθία 4K είναι εντυπωσιακή, επιτυγχάνοντας με 12 slices μια σχεδόν +3 επιτάχυνση σε σύγκριση με τον Static. Καταγράφει επίσης σημαντική υπεροχή σε σχέση με το Weight και WPA, αυξάνοντας ταυτόχρονα την επιτάχυνση κατά σχεδόν +0,4 σε σύγκριση με τον δεύτερο καλύτερο αλγόριθμο (TSLB*).



(α) Kimono

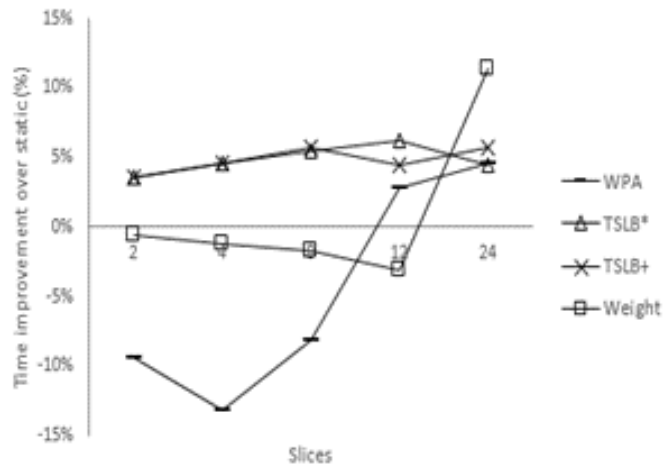


(β) Traffic

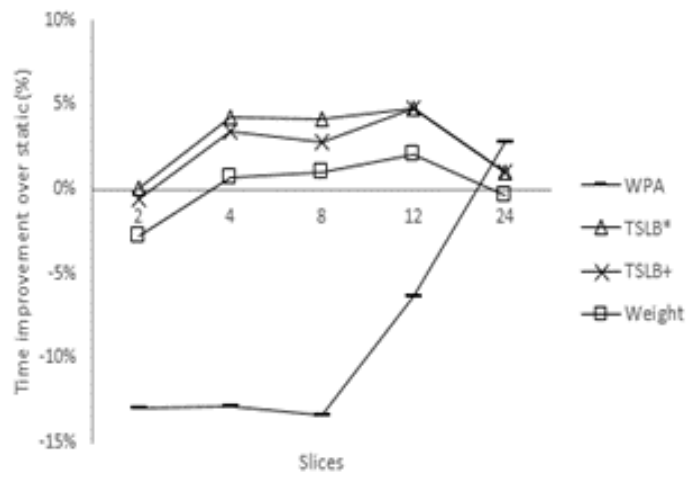


(γ) Bosphorus

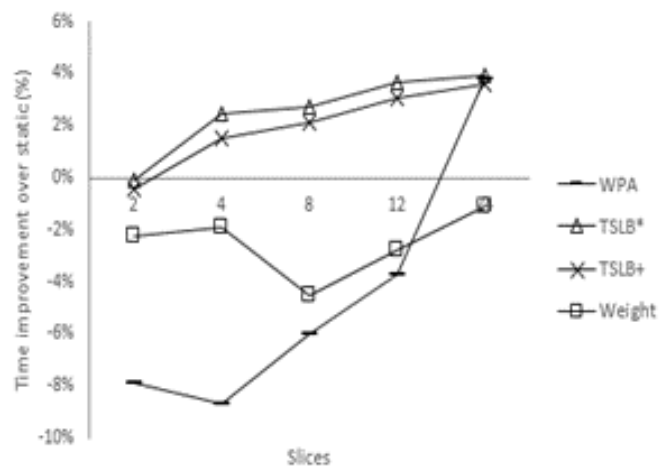
Εικόνα 14: Ποσοστό βελτίωσης του χρόνου του Static (Low Delay).



(α) Kimono



(β) Traffic

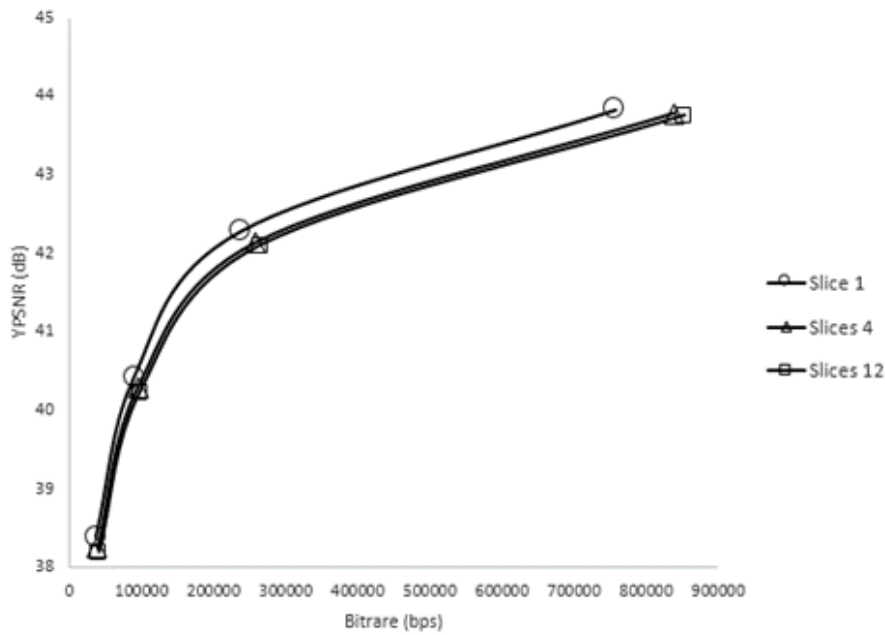


(γ) Bosphorus

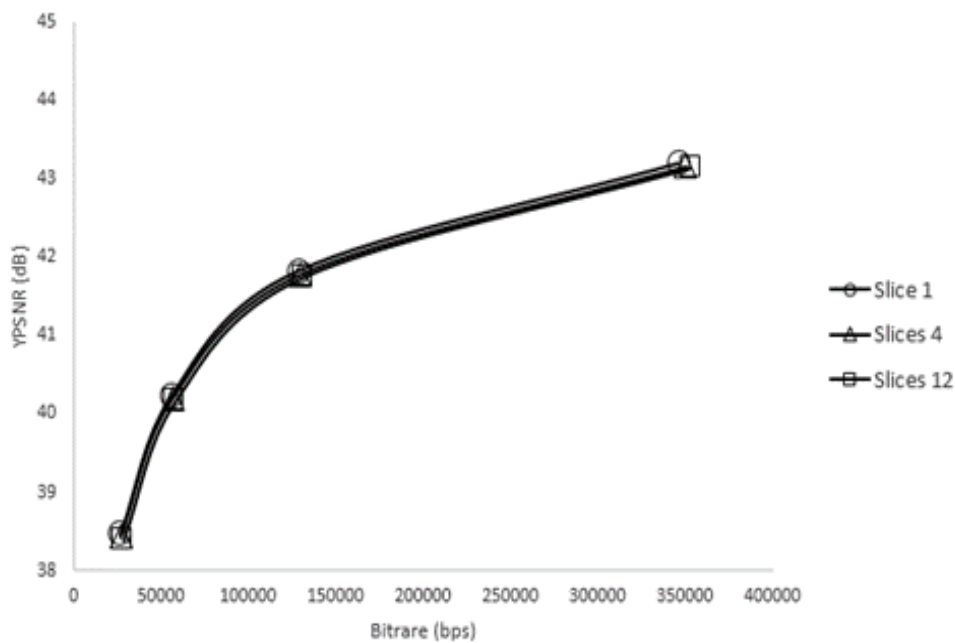
Εικόνα 15: Ποσοστό βελτίωσης του χρόνου του Static (Random Access).

Προκειμένου να έχουμε καλύτερη κατανόηση των αποτελεσμάτων επιτάχυνσης, στις Εικόνες 14 και 15 σχεδιάζουμε την απόδοση των αλγορίθμων χρησιμοποιώντας τον Static ως κριτήριο μέτρησης. Οι γραφικές παραστάσεις απεικονίζουν το ποσοστό βελτίωσης, δηλ. τη μείωση του χρόνου, που επιτυγχάνεται σε σχέση με τον Static. Για λόγους αναγνωσιμότητας δεν περιλαμβάνονται οι γραφικές των TSLB-Avg και TSLB-C, οι οποίοι είχαν δώσει χειρότερα αποτελέσματα σε σχέση με του TSLB* και TSLB+. Μια πρώτη παρατήρηση είναι ότι οι αλγόριθμοι που βασίζονται στο GOP είναι σαφείς νικητές όταν κάθε slice μπορεί να εκχωρηθεί σε διαφορετικό φυσικό πυρήνα. Επιπλέον, τείνουν να αυξάνουν τη διαφορά απόδοσης σε σχέση με τον Static καθώς ο αριθμός των slices αυξάνεται. Αυτό το αποτέλεσμα απεικονίζει περαιτέρω την αποτελεσματικότητα των μεθόδων εξισορρόπησης φόρτου που προτείνονται στην διπλωματική εργασία. Συγκρίνοντας το TSLB+ με το TSLB* φαίνεται ότι το TSLB+ ξεπερνάει τον TSLB* στην περίπτωση του LowDelay και της ακολουθίας 4K ενώ το αντίθετο ισχύει για το RandomAccess σενάριο. Τέλος, στην περίπτωση των 24 slices, η επιτάχυνση όλων των αλγορίθμων αυξάνεται (βλέπε Πίνακες 3 και 4) ακόμη και αν οι διαφορές τους με τον Static (εκτός από το Weight), μειώνονται (βλέπε Εικόνες 14 και 15). Αυτό οφείλεται στο γεγονός ότι η επιτάχυνση του Static επίσης αυξάνεται. Αξίζει επίσης να σημειωθεί ότι ορισμένοι αλγόριθμοι επιτυγχάνουν υπεργραμμική (superlinear) επιτάχυνση (πάνω από 12) σε ορισμένες περιπτώσεις λόγω hyper threading.

Μια τελευταία σημείωση αφορά στην ποιότητα του βίντεο. Οι γραφικές παραστάσεις στην Εικόνα 16 δείχνουν τις καμπύλες παραμόρφωσης (rate distortion RD) για QP = 22, 27, 32, 37 που αφορούν στον Static αλγόριθμο και στην ακολουθία Bosphorus, τόσο για την περίπτωση LowDelay όσο και για την περίπτωση RandomAccess. Παρατηρήθηκε στα πειράματά μας ότι καθώς ο αριθμός των slices αυξήθηκε, η ποιότητα μειώθηκε στην περίπτωση LowDelay, ενώ στην RandomAccess τα αρνητικά αποτελέσματα ήταν ελάχιστα. Αυτή η τάση είναι γνωστή από το H.264/AVC. Παρόλα αυτά, για συγκεκριμένο αριθμό slices τόσο το PSNR όσο και το bitrate εμφάνισαν ελάχιστες διαφορές μεταξύ των αλγορίθμων εξισορρόπησης φόρτου (οι αντίστοιχες γραφικές παραλείπονται καθώς οι καμπύλες «πέφτουν» η μία πάνω στην άλλη). Αυτό είναι ιδιαίτερα ενθαρρυντικό για το TSLB+ καθώς δηλώνει ότι η αυξημένη απόδοσή του, ειδικά σε σχέση με τον Static, έρχεται χωρίς κανένα επιπλέον κόστος σε σχέση με την ποιότητα.



(α) LowDelay



(β) RandomAccess

Εικόνα 16: Καμπύλες παραμόρφωσης Rate Distortion RD (Bosphorus, Static αλγόριθμος).

Ως εκ τούτου, το TSLB + αποτελεί την πλέον βιώσιμη λύση (επί του παρόντος) στο πρόβλημα της εξισορρόπησης των slices, ιδίως όταν τα ιεραρχικά LowDelay P frames χρησιμοποιούνται για την κωδικοποίηση βίντεο υψηλής ανάλυσης με περιοχές

υψηλής κίνησης, ενώ το TSLB* παρέχει μια καλή εναλλακτική λύση στην περίπτωση RandomAccess κωδικοποίησης.

Συνοψίζοντας τα συμπεράσματά μας μπορούμε να δηλώσουμε τα εξής:

1. Η προσέγγιση του Static αλγορίθμου αφήνει πολλά περιθώρια βελτίωσης χρόνου. Τα περιθώρια αυτά εξαρτώνται από την ακολουθία καθώς και από τα slices που χρησιμοποιούνται και τη μέθοδο κωδικοποίησης.
2. Ο πραγματικός χρόνος κωδικοποίησης των slices είναι ένα ανώτερο κριτήριο πρόβλεψης του χρόνου συμπίεσης σε σύγκριση με το στατικό κόστος του Weight στο [1].
3. Το WPA, η άλλη εναλλακτική λύση της βιβλιογραφίας [45], έδωσε μικτά αποτελέσματα χάνοντας σε πολλές δοκιμές, ακόμη και από τον Static αλγόριθμο, αλλά επιτυγχάνοντας σε όλες τις περιπτώσεις (εκτός Kimono και RandomAccess) υπεργραμμική συμπεριφορά με 24 slices σε 12 φυσικούς πυρήνες. Στις ίδιες περιπτώσεις συνήθως οι TSLB* και TSLB+ είχαν επίσης υπεργραμμική επιτάχυνση.
4. Με την ενσωμάτωση της δομής του GOP στο μηχανισμό λήψης αποφάσεων, μπορούν να σχεδιαστούν πολύ αποδοτικοί αλγόριθμοι εξισορρόπησης φόρτου, ιδιαίτερος στην περίπτωση του LowDelay.

Κεφάλαιο 4: Συμπεράσματα

Η γρήγορη και αποτελεσματική κωδικοποίηση βίντεο είναι πρωταρχικής σημασίας κυρίως όσο η ανάλυση μεγαλώνει. Σε αυτήν την εργασία αντιμετωπίσαμε το πρόβλημα της επιτάχυνσης στην παραλληλοποίηση επιπέδου slice στο HEVC, αποφασίζοντας δυναμικά τα μεγέθη των slices. Συγκρίναμε τους αλγόριθμους μας με τρεις εναλλακτικές λύσεις. Η πειραματική αξιολόγηση περιελάμβανε συνήθεις δοκιμαστικές ακολουθίες δύο σεναρίων (Low Delay και Random Access). Διαπιστώθηκε ότι οι αλγόριθμοι που χρησιμοποιούν τη δομή GOP για να υπολογίσουν το φορτίο του slice (TSLB* και TSLB+), οδηγούν σε σημαντικές βελτιώσεις σε σχέση με την στατική, ισομερώς ταξινομημένη τοποθέτηση CTU (αλγόριθμος Static), ενώ παράλληλα υπερέχουν των άλλων εναλλακτικών λύσεων. Μεταξύ των δύο, το TSLB+ προσφέρει την καλύτερη απόδοση, όντας σε θέση να βελτιώσει την επιτάχυνση της επεξεργασίας μιας ακολουθίας 4K κατά 25%.

Αναφορές

- [1] Y.-J. Ahn, T.-J. Hwang, D.-G. Sim, and W.-J. Han, "Implementation of Fast HEVC Encoder Based on SIMD and Data-Level Parallelism," *EURASIP J. of Image and Video Processing*, 2014:16, March 2014, doi:10.1186/1687-5281-2014-16.
- [2] S.M. Akramullah, I. Ahmad, and M.L. Liou, "A Data-Parallel Approach for Real-Time MPEG-2 Video Encoding," *J. of Parallel and Distributed Computing*, vol. 30, no. 2, pp. 129-146, 1995, doi:10.1006/jpdc.1995.1133.
- [3] F. Bossen, "Common Test Conditions and Software Reference Configurations," document JCTVC-H1100, JCT-VC, San Jose, CA, Feb. 2012.
- [4] F. Bossen, B. Bross, K. Sühring, and D. Flynn, "HEVC Complexity and Implementation Analysis," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685-1696, Dec. 2012, doi:10.1109/TCSVT.2012.2221255.
- [5] C.C. Chi, M.A. Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl, "Parallel Scalability and Efficiency of HEVC Parallelization Approaches," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1827-1838, Dec. 2012, doi:10.1109/TCSVT.2012.2223056.
- [6] C.C. Chi, M.A. Mesa, J. Lucas, B. Juurlink, and T. Schierl, "Parallel HEVC Decoding on Multi- and Many-core Architectures - A Power and Performance Analysis," *J. Signal Processing Systems*, vol. 71, no. 3, pp. 247-260, June 2013, doi:10.1007/s11265-012-0714-2.
- [7] J.-F. Franche, and S. Coulombe, "A Multi-Frame and Multi-Slice H.264 Parallel Video Encoding Approach with Simultaneous Encoding of Prediction Frames," *Proc. Int. Conf. Consumer Electronics, Communications and Networks (CECNet 2012)*, pp. 3034-3038, Apr. 2012, doi:10.1109/CECNet.2012.6202018.
- [8] G. Gao, W. Zhang, Y. Wen, Z. Wang, and W. Zhu, "Towards Cost-Efficient Video Transcoding in Media Cloud: Insights Learned From User Viewing Patterns," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1286-1296, Aug. 2015.
- [9] D. Grois, D. Marpe, A. Mulyoff, B. Itzhaky, and O. Hadar, "Performance Comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders," *Proc. Picture Coding Symp. (PCS 2013)*, pp. 394-397, Dec. 2013, doi:10.1109/PCS.2013.6737766.
- [10] HM 16.7 reference software, available at: <http://hevc.hhi.fraunhofer.de>

- [11] D. Hong, M. Horowitz, A. Eleftheriadis, and T. Wiegand, "H.264 Hierarchical P Coding in the Context of Ultra-Low Delay, Low Complexity Applications," *Proc. Picture Coding Symp. (PCS 2010)*, pp. 146-149, Dec. 2010, doi:10.1109/PCS.2010.5702445.
- [12] B. Jung, and B. Jeon, "Adaptive Slice-Level Parallelism for H.264/AVC Encoding Using Pre Macroblock Mode Selection," *J. Visual Communication and Image Representation*, vol. 19, no. 8, pp. 558-572, Dec. 2008, doi:10.1016/j.jvcir.2008.09.004.
- [13] A. Koivula, M. Viitanen, J. Vanne, T.D. Hämäläinen, and L. Fasnacht, "Parallelization of Kvazaar HEVC Intra Encoder for Multi-Core Processors," *Proc. IEEE Workshop Signal Processing Systems (SiSP 2015)*, pp. 1-6, Oct. 2015, doi:10.1109/SiPS.2015.7345015.
- [14] M. Koziri, D. Zacharis, I. Katsavounidis, and N. Bellas, "Implementation of the AVS Video Decoder on a Heterogeneous Dual-Core SIMD Processor," *IEEE Trans. Consumer Electronics*, vol. 57, no. 2, pp. 673-681, May 2011, doi:10.1109/TCE.2011.5955207.
- [15] M. Koziri, P. Papadopoulos, N. Tziritas, A.N. Dadaliaris, T. Loukopoulos, and S.U. Khan, "Slice-Based Parallelization in HEVC Encoding: Realizing the Potential through Efficient Load Balancing," *IEEE Workshop on Multimedia Signal processing (MMSP 2016)*, Sept. 2016, pp. 1-6.
- [16] M. G. Koziri, P. K. Papadopoulos, N. Tziritas, T. Loukopoulos, S. U. Khan, and A. Y. Zomaya, "Efficient Cloud Provisioning for Video Transcoding: Review, Open Challenges and Future Opportunities," *IEEE Internet Computing (IC)*, vol. 22(5), pp. 46-55, Sept. 2018.
- [17] M.G. Koziri and A. Eleftheriadis, "Joint Quantizer Optimization For Scalable Coding," *Proc. 17th IEEE Int. Conf. on Image Processing (ICIP 2010)*, IEEE, Hong Kong, China, Sept. 2010, pp. 1281-1284.
- [18] M.G. Koziri, P. K. Papadopoulos, N. Tziritas, A. N. Dadaliaris, T. Loukopoulos, G. I. Stamoulis, "On Planning the Adoption of New Video Standards in Social Media Networks: A General Framework and its Application to HEVC," *Social Network Analysis and Mining (Springer SNAM)*, vol. 7(1), pp. 1-32, July 2017.
- [19] M.G. Koziri, P. Papadopoulos, N. Tziritas, A.N. Dadaliaris, T. Loukopoulos, S.U. Khan, and C.-Z. Xu, "Adaptive Tile Parallelization for Fast Video Encoding in HEVC," *Proc. 12th Int. Conf. on Green Computing and*

- Communications (GreenCom 2016)*, *IEEE*, Chengdu, China, Dec. 2016, pp. 738-743.
- [20] M. Koziri, P. K. Papadopoulos, N. Tziritas, N. Giachoudis, T. Loukopoulos, S. U. Khan, and G.I. Stamoulis, "Heuristics for Tile Parallelism in HEVC," *Proc. 25th European Signal Processing Conf. (EUSIPCO 2017)*, *IEEE*, Kos, Greece, Aug. 2017, pp. 1514-1518.
- [21] M.G. Koziri, P.K. Papadopoulos, and T. Loukopoulos, "Combining Tile Parallelism with Slice Partitioning in Video Coding," *Proc. SPIE Conferences 2018, Optics and Electronics, SPIE*, San Diego, USA, Aug. 2018.
- [22] S. Ma, and C.-C.J. Kuo, "High-Definition Video Coding with Super-Macroblocks," *Proc. SPIE vol. 6508, Visual Communications and Image Processing 2007*, 650816, Jan. 2007, doi:10.1117/12.707582.
- [23] G. Malossi, D. Palomino, C. Diniz, A. Susin, and S. Bampi, "Adjusting video tiling to available resources in a per-frame basis in high efficiency video coding," *Proc. 14th IEEE International Conference New Circuits and Systems (NEWCAS)*, pp. 1-4, 2016.
- [24] D. Marpe, H. Schwarz, S. Bosse, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, T. Nguyen, S. Oudin, M. Siekmann, K. Sühring, M. Winken, and T. Wiegand, "Highly Efficient Video Compression Using Quadtree Structures and Improved Techniques for Motion Representation and Entropy Coding," *Proc. Picture Coding Symp. (PCS 2010)*, pp. 206-209, Dec. 2010, doi:10.1109/PCS.2010.5702464.
- [25] K.M. Misra, C.A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou, "An Overview of Tiles in HEVC," *IEEE J. Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 969-977, Dec. 2013, doi:10.1109/JSTSP.2013.2271451.
- [26] E. Monteiro, B.B. Vizzotto, C.M. Diniz, M. Maule, B. Zatt, and S. Bampi, "Parallelization of Full Search Motion Estimation Algorithm for Parallel and Distributed Platforms," *Int. J. of Parallel Programming*, vol. 42, no. 2, pp. 239-264, April 2014, doi:10.1007/s10766-012-0216-7.
- [27] OpenMP API, available at: <http://openmp.org>
- [28] P.K. Papadopoulos, M.G. Koziri, and T. Loukopoulos, "A Fast Heuristic for Tile Partitioning and Processor Assignment in HEVC," *Proc. ICIP 2018, IEEE*, Athens, Greece, Oct. 2018.
- [29] P. Piñol, H. Gomis, O.L. Granado, and M.P. Malumbres, "Slice-Based Parallel

- Approach for HEVC Encoder,” *The J. of Supercomputing*, vol. 71, no. 5, pp. 1882-1892, May 2015, doi:10.1007/s11227-014-1371-y.
- [30] M.T. Pourazad, C. Doutre, M. Azimi, and P. Nasiopoulos, “HEVC: The New Gold Standard for video Compression: How Does HEVC Compare with H.264/AVC?,” *IEEE Consumer Electronics Magazine*, vol. 1, no. 3, pp. 36-46, July 2012, doi:10.1109/MCE.2012.2192754.
- [31] A. Rodríguez, A. González, and M. P. Malumbres, “Hierarchical Parallelization of an H.264/AVC Video Encoder,” *Proc. Int. Symp. on Parallel Computing in Electrical Eng. (PARELEC 2006)*, pp. 363-368, Sept. 2006, doi:10.1109/PARELEC.2006.42.
- [32] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of Scalable Video Coding Extension of H.264/AVC Standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17(9), pp. 1103–1120, Sep. 2007.
- [33] M. Shafique, M.U.K. Khan, and J. Henkel, “Power efficient and workload balanced tiling for parallelized high efficiency video coding,” *Proc. 2014 IEEE International Conference on Image Processing (ICIP)*, pp. 1253-1257, 2014.
- [34] I. Storch, D. Palomino, B. Zatt, and L. Agostini, “Speedup-aware history-based tiling algorithm for the hevc standard,” *Proc. 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 824–828, 2016.
- [35] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012, doi:10.1109/TCSVT.2012.2221191.
- [36] V. Sze, and M. Budagavi, “A comparison of CABAC throughput for HEVC/H.265 VS. AVC/H.264,” *Proc. IEEE Workshop on Signal Processing Systems (SiPS 2013)*, pp. 165-170, Oct. 2013, 10.1109/SiPS.2013.6674499.
- [37] W.-J. Tsai, C.-H. Chan, C.-C. Tu, “Improve load balancing and coding efficiency of tiles in high efficiency video coding by adaptive tile boundary,” *Journal of Electronic Imaging*, vol. 26(1), 2017.
- [38] X. Wang, L. Song, M. Chen, and J.-J. Yang, “Paralleling Variable Block Size Motion Estimation of HEVC on Multi-Core CPU plus GPU Platform,” *Proc. IEEE Int. Conf. on Image Processing (ICIP 2013)*, pp. 1836-1839, doi:10.1109/ICIP.2013.6738378.
- [39] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, and A. Luthra, “Overview of the

- H.264/AVC Video Coding Standard,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003, doi:10.1109/TCSVT.2003.815165.
- [40] x265 HEVC encoder, available at: <http://x265.org>.
- [41] C. Yan, Y. Zhang, F. Dai, and L. Li, “Highly Parallel Framework for HEVC Motion Estimation on Many-Core Platform,” *Proc. Data Compression Conf. (DCC 2013)*, pp. 63-72, March 2013, doi:10.1109/DCC.2013.14.
- [42] Q. Yu, L. Zhao, and S. Ma, “Parallel AMVP Candidate List Construction for HEVC,” *Proc. IEEE Visual Communications and Image Processing (VCIP 2012)*, pp. 1-6, Nov. 2012, doi:10.1109/VCIP.2012.6410775.
- [43] M.R. Zakerinasab, and M. Wang, “Dependency-Aware Distributed Video Transcoding in the Cloud,” *Proc. IEEE Local Computer Network (LCN 2015)*, pp. 245-252, Oct. 2015.
- [44] Y. Zhao, L. Song, X. Wang, M. Chen, and J. Wang, “Efficient Realization of Parallel HEVC Intra Encoding,” *Proc. IEEE Int. Conf. on Multimedia and Expo Workshops (ICME Workshops 2013)*, pp. 1-6, July 2013, doi:10.1109/ICMEW.2013.6618415.
- [45] L. Zhao, J. Xu, Y. Zhou, and M. Ai, “A Dynamic Slice Control Scheme for Slice-Parallel Video Encoding,” *Proc. IEEE Int. Conf. on Image Processing (ICIP 2012)*, pp. 713-716, Sept.-Oct. 2012, doi:10.1109/ICIP.2012.6466959.