



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΝΑΛΥΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ
ΠΑΡΑΛΛΗΛΗΣ ΚΩΔΙΚΟΠΟΙΗΣΗΣ ΣΕ
ΕΠΙΠΕΔΟ ΔΙΑΜΕΡΙΣΜΟΥ ΤΩΝ CTS_s ΣΤΟ
ΠΡΟΤΥΠΟ ΚΩΔΙΚΟΠΟΙΗΣΗΣ ΒΙΝΤΕΟ VVC

ΣΕΦΕΡΛΗΣ ΧΑΡΑΛΑΜΠΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Μαρία Κοζύρη
Επίκουρη Καθηγήτρια

Λαμία, 3/2020



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΝΑΛΥΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ
ΠΑΡΑΛΛΗΛΗΣ ΚΩΔΙΚΟΠΟΙΗΣΗΣ ΣΕ
ΕΠΙΠΕΔΟ ΔΙΑΜΕΡΙΣΜΟΥ ΤΩΝ CTUs ΣΤΟ
ΠΡΟΤΥΠΟ ΚΩΔΙΚΟΠΟΙΗΣΗΣ ΒΙΝΤΕΟ VVC

ΣΕΦΕΡΛΗΣ ΧΑΡΑΛΑΜΠΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Μαρία Κοζύρη
Επικουρη Καθηγήτρια

Λαμία, 3/2020



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

ANALYSIS AND EVALUATION OF CODING
PARALLELIZATION ON CTU PARTITIONING
LEVEL IN VVC VIDEO CODING STANDARD

SEFERLIS CHARALAMPOS

FINAL THESIS

ADVISOR

Maria Koziri
Assistant Professor

Lamia, 3/2020

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία:/...../20.....

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Με τον αριθμό των streaming υπηρεσιών συνεχώς να αυξάνεται και με τα μέσα κοινωνικής δικτύωσης να εισάγουν όλο και περισσότερες λειτουργίες που έχουν ως βάση την καταγραφή και την αναπαραγωγή βίντεο, έχουμε ως αποτέλεσμα έναν τεράστιο όγκο δεδομένων στο διαδίκτυο που αν τον αφήσουμε ανεξέλεγκτο στο μέλλον θα μας δημιουργήσει προβλήματα. Η λύση στο πρόβλημα αυτό είναι η μείωση του όγκου των δεδομένων χωρίς να υπάρχουν απώλειες στις πληροφορίες, πράγμα το οποίο μπορεί να επιτευχθεί με την συμπίεση των βίντεο. Με την χρήση προτύπων συμπίεσης όπως το VVC που θα αναλύσουμε σε αυτή την πτυχιακή εργασία μπορούμε να μειώσουμε δραματικά τον όγκο που καταλαμβάνουν τα βίντεο. Με το μεγάλο κόστος επεξεργασίας όμως που απαιτείται για την κωδικοποίηση και αποκωδικοποίηση, μας είναι αδύνατον να χρησιμοποιήσουμε αυτή την τεχνολογία χωρίς την εφαρμογή της λειτουργίας παράλληλης εκτέλεσης στο VVC. Σε αυτή την πτυχιακή εργασία θα αναλύσουμε την σπουδαιότητα της παράλληλης εκτέλεσης του αλγορίθμου, προκειμένου να αυξηθούν οι επιδόσεις του VVC. Επιπλέον μέσα από γραφικές παραστάσεις παρατίθενται τα αποτελέσματα της έρευνας και των πειραμάτων που διεξήχθησαν σε δύο διαφορετικές ακολουθίες βίντεο, για την συγκρότηση αρκετών πληροφοριών ώστε να είναι δυνατή η σύγκριση της παράλληλης με τη σειριακή εκτέλεση της κωδικοποίησης.

ABSTRACT

With the number of streaming services constantly growing and social media introducing more and more features based on video recording and playback, the result is a massive amount of data circulated in the internet, that might cause some damages if it won't get restrained. The solution to this problem is a reduction of data volumes, that can be achieved by the compression of video sequences. Video coding standards like VVC can reduce drastically the size of video samples. The usage of this technology without the parallelization mode in VVC, is impossible because of its huge processing cost required for the encoding and decoding procedures. In this thesis we will discuss the importance of parallelization in order to achieve better performance in VVC. Furthermore, in order to gather the information about the comparison between parallel and serial coding, some experiments were conducted using two video sequences. Additionally, in this thesis there are some diagrams where the experiment results are shown.

Table of Contents

ΠΕΡΙΛΗΨΗ	I
ABSTRACT	III
<u>ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ</u>	<u>1</u>
<u>ΚΕΦΑΛΑΙΟ 2 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ</u>	<u>2</u>
<u>ΚΕΦΑΛΑΙΟ 3 ΤΟ ΠΡΟΤΥΠΟ ΣΥΜΠΙΕΣΗΣ VIDEO VVC</u>	<u>3</u>
ΥΒΡΙΔΙΚΗ ΚΩΔΙΚΟΠΟΙΗΣΗ VIDEO	3
ΤΟ ΠΡΟΤΥΠΟ ΣΥΜΠΙΕΣΗΣ VVC	7
ΕΙΣΑΓΩΓΗ	7
ΔΙΑΜΕΡΙΣΜΟΣ ΕΙΚΟΝΑΣ - PARTITIONING	7
INTRA PREDICTION	10
INTER PREDICTION	11
ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ	17
ΚΒΑΝΤΙΣΜΟΣ	18
ΦΙΛΤΡΑΡΙΣΜΑ	18
ΚΩΔΙΚΟΠΟΙΗΣΗ ΕΝΤΡΟΠΙΑΣ	19
<u>ΚΕΦΑΛΑΙΟ 4 SPLIT PARALLELISM</u>	<u>20</u>
<u>ΚΕΦΑΛΑΙΟ 5 ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ</u>	<u>25</u>
<u>ΚΕΦΑΛΑΙΟ 6 ΣΥΜΠΕΡΑΣΜΑΤΑ</u>	<u>34</u>
<u>ΒΙΒΛΙΟΓΡΑΦΙΑ</u>	<u>35</u>

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

Από το H.261 που αποτέλεσε το πρώτο αποδεκτό πρότυπο συμπίεσης μέχρι τους διαδόχους του στη σημερινή εποχή, έχουν γίνει σημαντικά βήματα εξέλιξης στην κωδικοποίηση βίντεο. Η ανάπτυξη στον τομέα της κωδικοποίησης είναι ανάλογη της συνεχούς αυξανόμενης ζήτησης για ανώτερη ποιότητα βίντεο, με υψηλότερες αναλύσεις και νέες τεχνολογίες που οδηγούν την εμπειρία θέασης σε όλο και ανώτερα επίπεδα. Το VVC προκειμένου να μπορεί να ανταπεξέλθει σε αυτές τις απαιτήσεις που με το πέρασμα του χρόνου αυξάνονται, παρέχει υψηλά επίπεδα συμπίεσης χωρίς σημαντικές απώλειες σε ποιότητα, ώστε το μέγεθος των βίντεο υψηλής ποιότητας (π.χ. 4K, 8K) να μην είναι απαγορευτικό για να χρησιμοποιηθεί. Τέτοια επίπεδα συμπίεσης όμως έχουν ως συνέπεια την αυξημένη υπολογιστική δαπάνη για να πραγματοποιηθεί η κωδικοποίηση και η αποκωδικοποίηση βίντεο. Οι περισσότερες συσκευές που κάνουν χρήση βίντεο δεν έχουν τους υπολογιστικούς πόρους για να πραγματοποιήσουν αυτή την διαδικασία μέσα σε λογικό χρόνο για να μπορεί να χρησιμοποιηθεί, γεγονός που οδήγησε στην ανάπτυξη της παραλληλοποίησης. Στο VVC με την χρήση της παράλληλης κωδικοποίησης επιτυγχάνεται σε ικανοποιητικό επίπεδο η πτώση του χρόνου συμπίεσης και αποσυμπίεσης του βίντεο από κοινές συσκευές και η εξέλιξή του συνεχίζεται με στόχο την αύξηση των επιδόσεων του αλγορίθμου.

Σε αυτή την πτυχιακή θα αναλύσουμε την αύξηση των επιδόσεων του VVC όταν το εκτελούμε με ενεργοποιημένη την λειτουργία της παράλληλης εκτέλεσης σε επίπεδο CTU σε σχέση με την σειριακή εκτέλεση του αλγορίθμου, καθώς και τις διαφορές που συναντούμε στην ροή του κώδικα στις δύο περιπτώσεις. Επιπλέον θα παρουσιάσουμε μια σύντομη περιγραφή του (υπό διαμόρφωση) προτύπου συμπίεσης βίντεο VVC αναλύοντας μερικές από τις βασικές τεχνολογίες που χρησιμοποιεί στην κωδικοποίηση, καθώς και τις εξελιγμένες τεχνικές που εισάγει και το κάνουν να ξεχωρίζει από προγενέστερα πρότυπα χαμηλότερων επιδόσεων. Επίσης θα παρουσιάσουμε την κεντρική ιδέα και τις βασικές αρχές της συμπίεσης βίντεο, που έχει θεοπίσει η υβριδική κωδικοποίηση και ακολουθείται από το VVC αλλά και από όλα τα πρότυπα που έχουν αναπτυχθεί τα τελευταία χρόνια.

Μέσα από την έρευνα και τα πειράματα που πραγματοποιήσαμε μπορούμε να δούμε και με πραγματικά νούμερα τις διαφορές που υπάρχουν στην ταχύτητα και γενικότερα στις επιδόσεις του αλγορίθμου όταν χρησιμοποιείται η παράλληλη εκτέλεση, σε δυο ακολουθίες βίντεο. Επιπλέον μπορούμε να διακρίνουμε την επίδραση που έχουν στον αλγόριθμο οι διαφορετικές ρυθμίσεις της παράλληλης εκτέλεσης και γενικότερα της κωδικοποίησης καθώς και να ανακαλύψουμε τα δυνατά και τα αδύναμα σημεία του, για κάθε διαφορετική ρύθμιση και βίντεο. Με την χρήση γραφικών παραστάσεων αναλύουμε τα αποτελέσματα που μας έδωσαν τα πειράματα που πραγματοποιήσαμε και κάνουμε την σύγκρισή τους ευκολότερη.

Πιο συγκεκριμένα η δομή αυτής της πτυχιακής εργασίας είναι οργανωμένη όπως ακολουθεί στην συνέχεια. Στο Κεφάλαιο 2 γίνεται μια μικρή αναφορά σε μέρος του ερευνητικού έργου που υπάρχει στη βιβλιογραφία και αφορά την παραλληλοποίηση της κωδικοποίησης βίντεο. Το Κεφάλαιο 3 παρουσιάζει μια γενική ιδέα για την υβριδική κωδικοποίηση και αναλύει κάποιες από τις νέες τεχνολογίες που εισάγει το VVC. Στο Κεφάλαιο 4 παρουσιάζονται οι διαφορές μεταξύ σειριακής και παράλληλης εκτέλεσης του VVC. Το Κεφάλαιο 5 παραθέτει τα αποτελέσματα των πειραμάτων με γραφικές παραστάσεις. Τέλος στο κεφάλαιο 6 αναφέρονται τα συμπεράσματα που εξάγουμε από αυτή την πτυχιακή εργασία.

ΚΕΦΑΛΑΙΟ 2 Βιβλιογραφική Επισκόπηση

Κάθε νέο πρότυπο κωδικοποίησης βίντεο πρέπει να παρουσιάζει μια βελτίωση στην απόδοση συμπίεσης τουλάχιστον 30% σε σχέση με τον προκάτοχό του. Τα περισσότερα πρότυπα επιτυγχάνουν βελτιώσεις γύρω στο 50%. Για να επιτευχθεί αυτός ο στόχος κάθε νέο πρότυπο εισάγει καινούργια εργαλεία τα οποία όμως αυξάνουν την πολυπλοκότητα του αποκωδικοποιητή και πολύ περισσότερο αυτήν του κωδικοποιητή. Καθώς λοιπόν τα σύγχρονα πρότυπα κωδικοποίησης γίνονται πιο πολύπλοκα, αυξάνει η ανάγκη να βρεθούν αποδοτικοί τρόποι παραλληλοποίησης της όλης διαδικασίας [7]. Στην παρούσα εργασία θα ασχοληθούμε με την μελέτη της παραλληλοποίησης σε επίπεδο CTU στο νέο πρότυπο συμπίεσης VVC [1]. Καθώς το VVC δεν έχει γίνει ακόμα επισήμως πρότυπο, αλλά βρίσκεται ακόμα στο στάδιο της δημιουργίας δεν υπάρχει ακόμα επαρκής βιβλιογραφία σχετικά με την παραλληλοποίησή του. Δεν μπορούμε να πούμε όμως ότι ισχύει το ίδιο για παλαιότερα πρότυπα. Στη συνέχεια ακολουθεί μια περιληπτική αναφορά μερικών σημαντικών ερευνητικών εργασιών που ασχολήθηκαν με το θέμα της παραλληλοποίησης σε προηγούμενα πρότυπα, και κυρίως στο HEVC [4].

Στην [11] παρουσιάζονται και αναλύονται οι τρεις βασικοί τρόποι παραλληλοποίησης που προτείνονται από το ίδιο το πρότυπο HEVC [5], οι οποίοι είναι ονομαστικά οι: wavefront, tiles και slices. Στη συγκεκριμένη εργασία δίνεται αρκετή βαρύτητα στον πρώτο από τους τρεις τρόπους παραλληλοποίησης. Από την άλλη μεριά η εργασία [12] επικεντρώνεται στην παραλληλοποίηση με wavefront από την μεριά όμως του αποκωδικοποιητή. Η παραλληλοποίηση σε επίπεδο tile αναλύεται στην [13]. Ένα σχήμα κωδικοποίησης με περιοχές ενδιαφέροντος (ROI – Region Of Interest) χρησιμοποιήθηκε στη συγκεκριμένη εργασία, όπου μικρότερα tiles αντιστοιχήθηκαν σε ROI για την οποία αναμενόταν υψηλός χρόνος κωδικοποίησης. Άλλες εργασίες που αναφέρονται στην παραλληλοποίηση σε επίπεδο tiles, και πώς μπορούμε να προσαρμόσουμε τα όριά τους έτσι ώστε να πετύχουμε καλύτερη εξισορρόπηση φορτίου είναι οι [8], [9] και [9].

Σε επίπεδο slice παραλληλοποίησης αρκετές ερευνητικές εργασίες είχαν ήδη γίνει από την εποχή του H.264/AVC [14]. Μερικές από αυτές είναι οι [15], [16], [17] και [18]. Στην [18] γίνεται προσαρμοζόμενη ανάθεση Macroblock στα slices. Η τεχνική βασίζεται στον υπολογισμό του WPA (Weighted Past Average) με έναν παράγοντα 0.5 προκειμένου να υπολογιστεί το κόστος του Macroblock για το επόμενο καρέ. Στη συνέχεια τα Macroblocks διανέμονται στα slices έτσι ώστε να μειωθούν οι διαφορές στα συνολικά κόστη. Στην [15] το πρόβλημα της εξισορρόπησης των slices αντιμετωπίστηκε με το να ανατίθενται περισσότερα slices από υπάρχοντες πυρήνες. Στην [16] παρουσιάζεται ένας αλγόριθμος ο οποίος προσαρμόζει το μέγεθος των slices προκειμένου να βελτιωθεί η εξισορρόπηση φορτίου, ενώ στην [17] προτείνεται ιεραρχική παραλληλοποίηση δυο επιπέδων.

Τέλος, ένας άλλος τρόπος παραλληλοποίησης, ο οποίος είναι πλέον διαχρονικός στην τομέα κωδικοποίησης βίντεο καθώς πρωτοεμφανίστηκε τη δεκαετία του '90 ήδη από το MPEG2 [19] είναι αυτή με τη χρήση SIMD εντολών. Πιο πρόσφατα παραδείγματα εργασιών που κάνουν χρήση εντολών SIMD για την επίτευξη της παραλληλοποίησης είναι τα [20], [21], [22], [23] και [24]. Αξίζει να σταθούμε λίγο στην [23] όπου παρουσιάζεται ένας συνδυασμός από GPU και πολυπύρηνων CPU προκειμένου να επιτευχθεί η παραλληλοποίηση στην εκτίμηση κίνησης. Στην [24] παρουσιάζεται ένα πλαίσιο για την ανάλυση των εξαρτήσεων των γειτονικών CTUs. Τα CTUs δημιουργούν έναν DAG ο οποίος στη συνέχεια δρομολογείται για παράλληλη υλοποίηση. Διάφοροι βαθμοί παραλληλοποίησης συζητούνται που κυμαίνονται από ένα μόνο CTU σε σύνολα από CTUs. Μια παρόμοια προσέγγιση ακολουθείται και στην [25], μόνο που εκεί μελετούν μόνο την περίπτωση του intra coding χρησιμοποιώντας των κωδικοποιητή ανοικτού λογισμικού x265 [26].

ΚΕΦΑΛΑΙΟ 3 Το πρότυπο συμπίεσης video VVC

Υβριδική κωδικοποίηση video

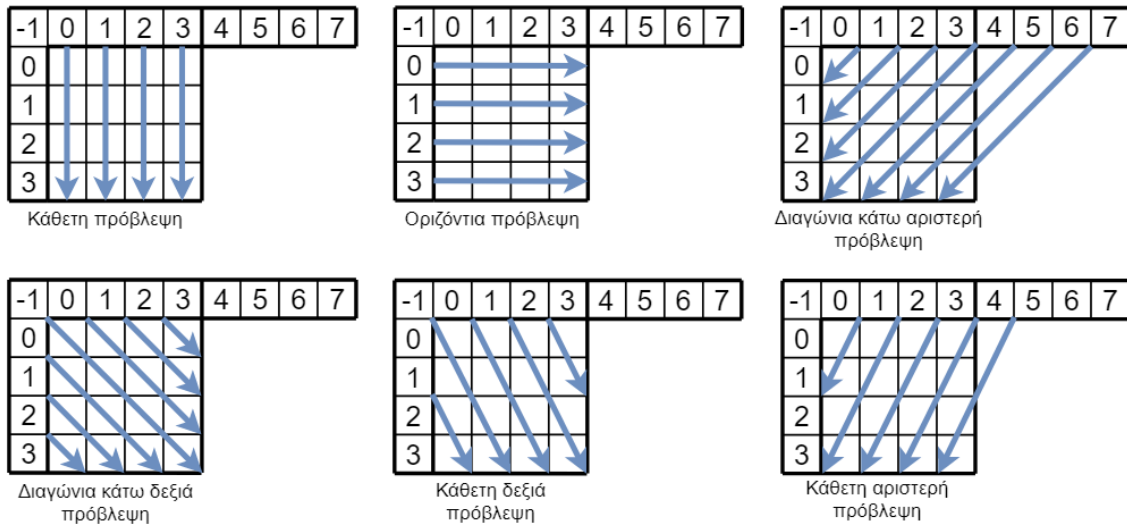
Η Υβριδική κωδικοποίηση βίντεο αποτελεί την βάση πάνω στην οποία αναπτύσσονται τα πρότυπα συμπίεσης βίντεο που έχουν δημιουργηθεί τα τελευταία χρόνια. Είναι γνωστή για τις τεχνικές που ακολουθεί προκειμένου να είναι δυνατή η συμπίεση βίντεο με τις λιγότερες δυνατές απώλειες σε ποιότητα εικόνας, αλλά και με την ταχύτερη και αποδοτικότερη ολοκλήρωση της κωδικοποίησης. Το VVC, όπως και κάθε πρότυπο συμπίεσης βίντεο, ακολουθεί αυτές τις τεχνικές και τις αρχές που έχει θέσει η υβριδική κωδικοποίηση. Ακολουθώντας αυτές τις αρχές κάθε καρέ ενός βίντεο πρέπει για αρχή να σπάει σε μικρότερα κομμάτια, να πραγματοποιείται πρόβλεψη κομματιών του βίντεο, να μετασχηματίζεται και να κβαντίζεται και τέλος να περνάει από την μηδενικής απώλειας κωδικοποίηση εντροπίας.

Τα κομμάτια στα οποία σπάει η εικόνα ονομάζονται μπλοκ, το καθένα από τα οποία μπορεί να διασπαστεί σε μικρότερα μπλοκ αν χρειαστεί και το μέγιστο και ελάχιστο μέγεθος που μπορούν να φτάσουν εξαρτάται από το πρότυπο συμπίεσης βίντεο που εκτελεί την κωδικοποίηση. Από τα πρότυπα συμπίεσης εξαρτάται και το αν τα μπλοκ που δημιουργούνται θα διαχωριστούν σε ομάδες δίνοντας διαφορετικές ιδιότητες και ονομασίες στα μπλοκ της κάθε ομάδας ή όχι.

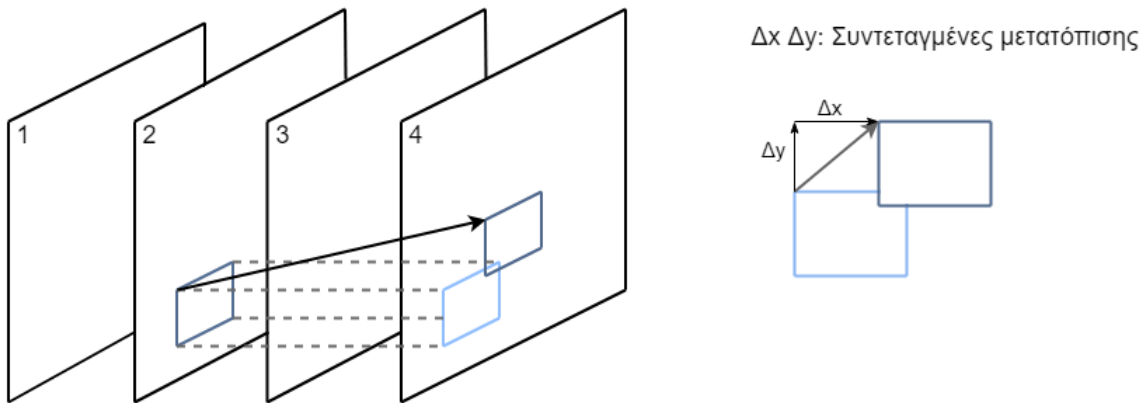
Η αρχή της πρόβλεψης κομματιών του βίντεο είναι θεμελιώδης και εξαιρετικά σημαντική για την κωδικοποίηση καθώς χωρίς αυτή δεν θα μπορούσαμε να μειώσουμε σημαντικά το μέγεθος των βίντεο κατά την συμπίεση. Με την πρόβλεψη υπάρχει δυνατότητα να αφαιρείται από το κάθε καρέ του βίντεο ένας μεγάλος όγκος δεδομένων ο οποίος έχει την δυνατότητα να προβλεφθεί με την χρήση συγκεκριμένων διαδικασιών. Είναι δηλαδή δυνατή η ανασυγκρότηση των καρέ ακόμα και αν αρκετά κομμάτια τους έχουν αφαιρεθεί από το τελικό αποτέλεσμα της κωδικοποίησης. Προκειμένου να καλυφθούν όσο το δυνατόν περισσότερα ενδεχόμενα πάνω στην πρόβλεψη, η υβριδική κωδικοποίηση βίντεο εμπεριέχει έναν μεγάλο αριθμό διαδικασιών και λειτουργιών πρόβλεψης οι οποίες χωρίζονται σε δύο κατηγορίες, την intra prediction και την inter prediction.

Η πρόβλεψη intra prediction εκμεταλλεύεται την συσχέτιση γειτονικών μπλοκ του τρέχοντος καρέ που έχουν ήδη ανασχηματιστεί, προκειμένου να γίνει η πρόβλεψη. Υπάρχουν πολλές διαφορετικές λειτουργίες intra prediction από τις οποίες ο αλγόριθμος πρέπει να επιλέξει ποια θα χρησιμοποιήσει για το κάθε μπλοκ που κωδικοποιείται, ώστε να έχουμε τα καλύτερα δυνατά αποτελέσματα. Στην Εικόνα 1 μπορούμε να δούμε την γενική λογική που ακολουθείται για την πραγματοποίηση της intra prediction. Φυσικά οι λειτουργίες παρόλο που ακολουθούν την ίδια λογική για την πραγματοποίηση της πρόβλεψης, διαφέρουν σημαντικά μεταξύ των προτύπων συμπίεσης βίντεο.

Η πρόβλεψη inter prediction αντισταθμίζει την κίνηση των πραγματικών αντικειμένων που απεικονίζονται στα βίντεο, προκειμένου να εκμεταλλευτεί την κίνηση αυτή μεταξύ των καρέ ώστε να πραγματοποιήσει την πρόβλεψη. Χρησιμοποιεί δηλαδή μπλοκ γειτονικών καρέ που έχουν ήδη ανασχηματιστεί και με την χρήση συντεταγμένων για να υποδειχθεί η πορεία που ακολουθούν τα αντικείμενα πραγματοποιείται τελικά η πρόβλεψη. Η inter prediction δεν είναι ανάγκη να γίνει από το αμέσως προηγούμενο καρέ. Μπορεί να γίνει από οποιοδήποτε καρέ κρίνει ο κωδικοποιητής ότι είναι καταλληλότερο για να χρησιμοποιηθεί στην προκειμένη περίπτωση. Οι δυνατότητες και τα κριτήρια επιλογής καρέ διαφέρουν από πρότυπο σε πρότυπο συμπίεσης. Το ίδιο ισχύει και για τις διάφορες λειτουργίες inter prediction παρόλο που ακολουθούν την ίδια λογική. Στην Εικόνα 2 αναπαριστάτε η λογική αυτή.



Εικόνα 1 -Intra prediction



Εικόνα 2 -Inter prediction

Στο τέλος της διαδικασίας πρόβλεψης δημιουργείται το μπλοκ διαφορών (residual block) το οποίο είναι αποτέλεσμα της αφαίρεσης των προβλεπόμενων μπλοκ από το τρέχον μπλοκ που κωδικοποιείται. Στην υβριδική κωδικοποίηση βίντεο το residual μπλοκ που παίρνει την μορφή δυοδιάστατου πίνακα $N \times N$ περνάει από την διαδικασία του μετασχηματισμού. Το αποτέλεσμα του μετασχηματισμού είναι και πάλι ένας δυοδιάστατος πίνακας $N \times N$ που ονομάζεται transform coefficients (coeff) και με την σειρά του περνάει στην διαδικασία κβαντισμού (Qstep). Ο δυοδιάστατος πίνακας $N \times N$ που θα παραχθεί μετά τον κβαντισμό στην συνέχεια θα αναδιαταχθεί σε γραμμική συστοιχία μέσω της σάρωσης ζιγκ-ζαγκ (Εικόνα 3) του πίνακα που θα ξεκινήσει από το πάνω αριστερά στοιχείο του. Έπειτα η γραμμική συστοιχία αυτή θα υποστεί επεξεργασία ώστε τα μηδενικά της στοιχεία να συμπυκνωθούν με τα μη μηδενικά της προκειμένου να γίνει εξοικονόμηση χώρου. Η διαδικασία αυτή αν και

χρονοβόρα μειώνει κατά πολύ το μέγεθος του κάθε μπλοκ μετά την διαδικασία της πρόβλεψης. Η υβριδική κωδικοποίηση έρχεται στο τέλος της με την κωδικοποίηση εντροπίας που και πάλι διαφέρει ανάλογα με το πρότυπο συμπίεσης βίντεο που εκτελείται.

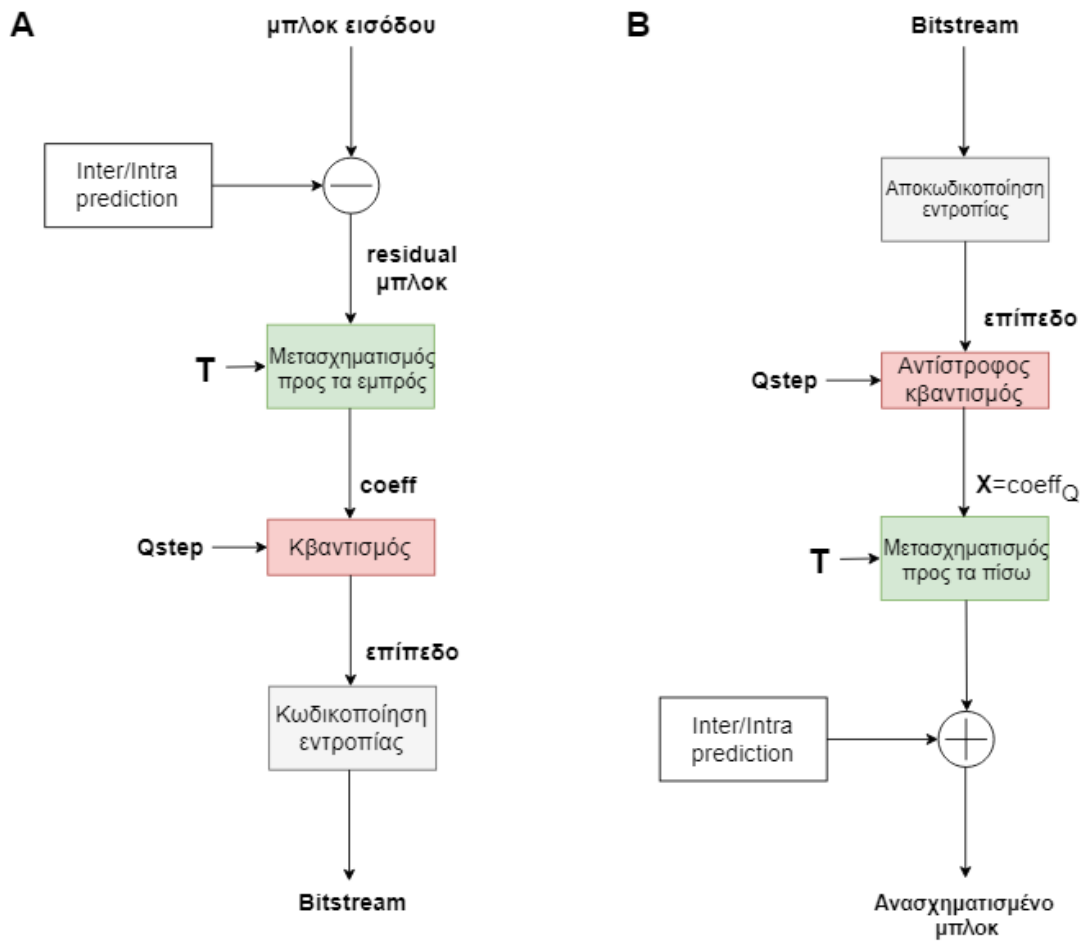
2	2	0	1
-1	1	0	1
1	-1	0	0
0	0	0	0

Γραμμική συστοιχία: 2, 2, -1, 1, 1, 0, -1, 0, -1, 0, 0, 0, 1, 0, 0, 0

Εικόνα 3 - Σάρωση ζικκ-ζαγκ και γραμμική συστοιχία

Ολόκληρη αυτή η διαδικασία θα πρέπει να πραγματοποιηθεί αντίστροφα προκειμένου να ανασχηματιστεί το βίντεο. Για να γίνει αυτό εκτός του συμπιεσμένου βίντεο, ο κωδικοποιητής δίνει και σαν αποτέλεσμα στοιχεία που είναι αναγκαία για να γίνει σωστά και με όσο το δυνατόν λιγότερες απώλειες δεδομένων η αντίστροφη διαδικασία από τον αποκωδικοποιητή. Τέτοια στοιχεία είναι το είδος μετασχηματισμού και ο πίνακας μετασχηματισμού που χρησιμοποιήθηκε στο κάθε μπλοκ, το είδος inter prediction και οι συντεταγμένες μαζί με τον αριθμό του καρτέ από το οποίο έγινε η πρόβλεψη για κάθε μπλοκ, καθώς και το είδος της intra prediction που έγινε στο κάθε μπλοκ. Επιπλέον χρειάζεται τις πληροφορίες για τον πίνακα κβαντισμού που χρησιμοποιήθηκε προκειμένου να επιλεχθεί η αντιστοιχη διαδικασία αντίστροφου κβαντισμού στον αποκωδικοποιητή. Με την χρήση αυτών των δεδομένων σε συνδυασμό με το συμπιεσμένο βίντεο ο αποκωδικοποιητής είναι σε θέση να κάνει την αντίστροφη διαδικασία, προκειμένου να ανασχηματιστεί το βίντεο.

Η απόδοση που μπορεί να φτάσει ένα πρότυπο υβριδικής κωδικοποίησης βίντεο εξαρτάται από πολλές πτυχές σχεδιασμού όπως είναι τα χρησιμοποιημένα φίλτρα παρεμβολής για παρεμβολή υπό-δειγμάτων, η αποτελεσματικότητα της κωδικοποίησης εντροπίας και οι τεχνικές in-loop filtering που χρησιμοποιούνται. Κύρια πηγή βελτίωσης στις επιδόσεις των προτύπων συμπίεσης από την μία γενιά στην επόμενη αποτελεί ο αυξημένος αριθμός των δυνατοτήτων που υποστηρίζονται για την βέλτιστη κωδικοποίηση ενός μπλοκ δειγμάτων ή μιας εικόνας. Οι δυνατότητες αυτές περιλαμβάνουν βελτιώσεις όπως, μεγαλύτερος αριθμός λειτουργιών πρόβλεψης intra prediction, επέκταση στο σύνολο διαθέσιμων εικόνων αναφοράς, μεγαλύτερη ευελιξία ως προς την επιλογή της σειράς με την οποία θα πραγματοποιηθεί η κωδικοποίηση των εικόνων, αυξημένη ακρίβεια στα διανύσματα κίνησης, μεγαλύτερος αριθμός προγνωστικών διανυσμάτων κίνησης, αύξηση στον αριθμό υποστηριζόμενων μεγεθών μετασχηματισμού καθώς και αύξηση στον αριθμό των μεγεθών των μπλοκ πρόβλεψης αντιστάθμισης κίνησης. Επομένως οι λειτουργίες της υβριδικής κωδικοποίησης βίντεο ακολουθώντας πάντα την ίδια λογική, τροποποιούνται και διαφέρουν σημαντικά μεταξύ των προτύπων συμπίεσης, πράγμα που οδηγεί σε καλύτερες επιδόσεις κάποιων σε σχέση με τα υπόλοιπα.



Εικόνα 4 - Υβριδική κωδικοποίηση βίντεο. (T): Πίνακας μετασχηματισμού, (A): Κωδικοποιητής, (B): Αποκωδικοποιητής

Το πρότυπο συμπίεσης VVC

Εισαγωγή

Το Versatile Video Codec (VVC) είναι ένα νέο πρότυπο συμπίεσης βίντεο το οποίο ξεπερνάει σε δυνατότητες τον προκάτοχό του HEVC (ITU-T H.265 | ISO/IEC 23008-2) και τις τρέχουσες επεκτάσεις του. Η επίσημη προσπάθεια για την έναρξη της δημιουργίας του VVC ακολουθεί την τεχνική έρευνα που ξεκίνησε να διεξάγεται τον Οκτώβριο του 2015 από ειδικούς από την VCEG (Q6/16) και την ISO/IEC JTC1 SC29/WG11 (MPEG). Αφού είχαν ολοκληρωθεί πρόσφατα οι εργασίες για την επέκταση του HEVC και του AVC, το γκρουπ εμπειρογνομένων κωδικοποίησης βίντεο VCEG (Q6/16) μελέτησε πιθανές τεχνικές κωδικοποίησης βίντεο επόμενης γενιάς που θα ξεπερνούσαν τις δυνατότητες των προτύπων που είχαν αναπτυχθεί μέχρι τότε. Στις 27 Οκτωβρίου του 2017 οι δυο ομάδες ήρθαν σε συμφωνία και μαζί δημιούργησαν μια νέα κοινή ομάδα εμπειρογνομένων πάνω στο βίντεο (JVET) για την ανάπτυξη του νέου αυτού προτύπου.

Το κύριο πρότυπο κωδικοποίησης βίντεο που προηγείται του VVC ήταν το HEVC (ITU-T H.265 | ISO/IEC 23008-2) το οποίο ξεκίνησε να αναπτύσσεται το 2010 και η πρώτη έκδοσή του ολοκληρώθηκε το 2013. Το HEVC έδωσε λύση στα ζητήματα της συνεχούς ζήτησης για υψηλότερων αναλύσεων βίντεο (όπως η ανάλυση 4k x 2k ή 8k x 4k) πράγμα το οποίο ξεπερνούσε τις δυνατότητες του προηγούμενου κύριου προτύπου H.264/MPEG-4 AVC. Με το πέρασμα του χρόνου όμως και με τις streaming υπηρεσίες να πολλαπλασιάζονται, η ανάγκη για καλύτερη κωδικοποίηση βίντεο ολοένα και αυξάνεται με αποτέλεσμα κάποια στιγμή το HEVC να σταματήσει να μας καλύπτει. Το VVC σχεδιάστηκε με πρωταρχικούς στόχους να καθοριστεί μια τεχνολογία κωδικοποίησης βίντεο που να έχει δυνατότητες συμπίεσης που ξεπερνούν τις ικανότητες των προτύπων των προηγούμενων γενιών και να είναι εξαιρετικά ευπροσάρμοστη για αποτελεσματική χρήση σε ένα ευρύτερο φάσμα εφαρμογών. Ορισμένοι βασικοί τομείς στους οποίους μπορεί να γίνει χρήση αυτού του προτύπου περιλαμβάνουν τα βίντεο ανάλυσης ultra-high-definition (όπως με ανάλυση 3840 x 2160 ή 7620 x 4320 και με βάθος bit 10 ή 12), τα βίντεο με υψηλού επιπέδου δυναμική εμβέλεια απεικόνισης και ευρεία γκάμα χρωμάτων και τα βίντεο προβολής 360° με κατεύθυνση προς την κατεύθυνση προβολής, χρησιμοποιώντας μια κοινή μορφή προβολής όπως είναι η προβολή με ισομετρική κατεύθυνση.

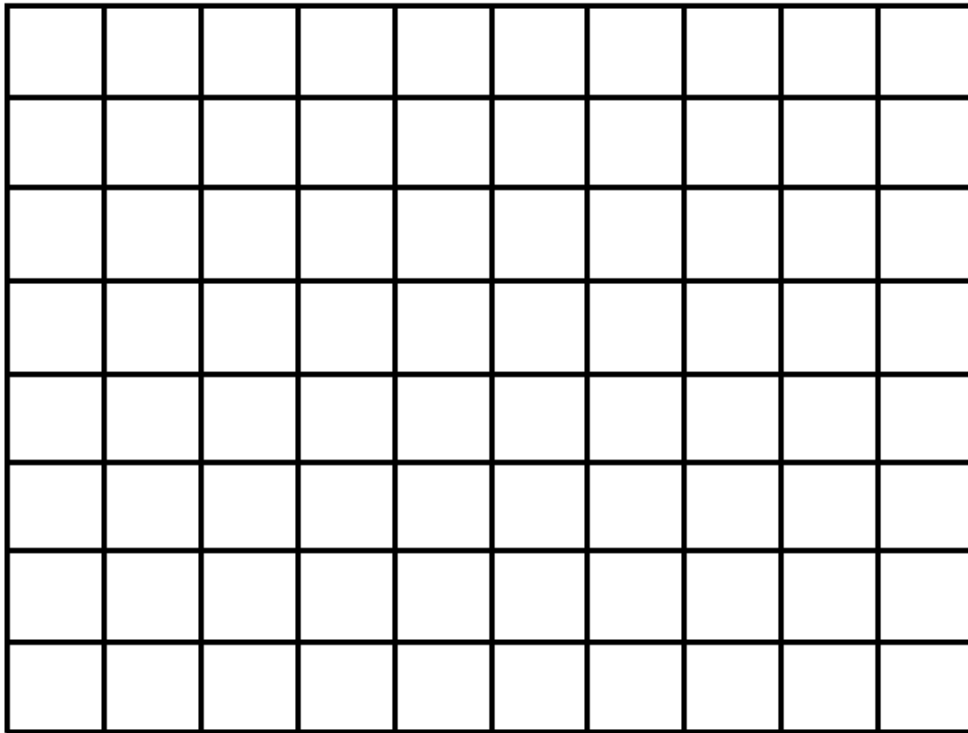
Το VVC περιλαμβάνει πολλά καινούρια στοιχεία σε σχέση με τα παλαιότερα πρότυπα συμπίεσης βίντεο (HEVC, H.264 κ.α.) αλλά περιέχει και αρκετά στοιχεία που παραμένουν ίδια ή έχουν μερικώς τροποποιηθεί ώστε το νέο αυτό πρότυπο να έχει την καλύτερη δυνατή απόδοση. Στη συνέχεια ακολουθεί μια σύντομη περιγραφή των βασικότερων νέων εργαλείων κωδικοποίησης που έχουν προστεθεί στο VVC καθώς και των αλλαγών που έχουν γίνει σε εργαλεία που προϋπήρχαν σε προγενέστερα πρότυπα συμπίεσης.

Διαμερισμός Εικόνας - Partitioning

Διαμερισμός της εικόνας σε CTUs

Στο Partitioning (Διαμερισμός της εικόνας) η έννοια του Coding Tree Unit (CTU) έχει παραμείνει ίδια με αυτή του HEVC. Μια εικόνα περιέχει τρεις ουσιοιχίες δειγμάτων και το CTU αποτελείται από ένα μπλοκ διαστάσεων $N \times N$ δειγμάτων luma μαζί με δύο

αντίστοιχα μπλοκ δειγμάτων chroma (χρώματος). Τα CTUs του VVC είναι μεγαλύτερα από του HEVC αφού επιτρέπεται να έχουν μέγεθος μέχρι και 128 x 128.



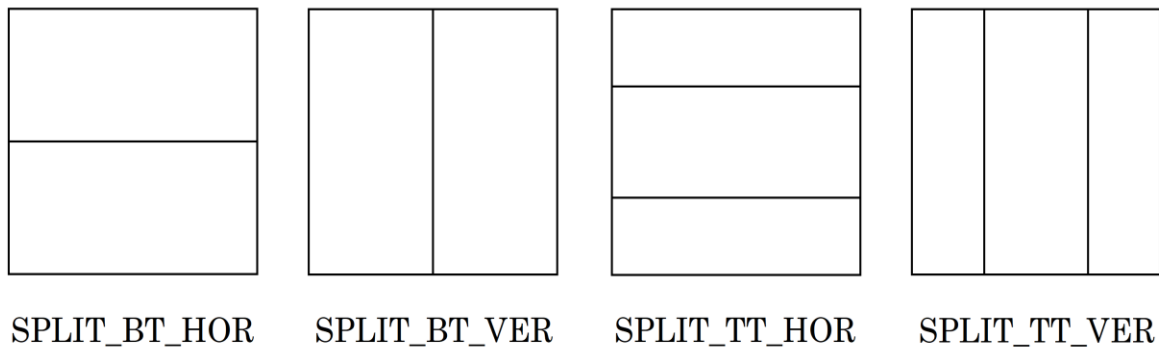
Εικόνα 5- Παράδειγμα εικόνας που έχει χωριστεί σε CTUs

Διαχωρισμός των CTUs με την χρήση μιας δομής δέντρου

Στο HEVC ένα CTU χωρίζεται σε CUs με την χρήση δομής τετραδικών δέντρων (quad-tree unit). Η απόφαση για το αν μια περιοχή της εικόνας θα κωδικοποιηθεί με χρήση της πρόβλεψης inter-picture ή με χρήση της πρόβλεψης intra-picture γίνεται στο επίπεδο φύλλου CU. Κάθε φύλλο CU μπορεί να διαχωριστεί επιπλέον σε ένα, δύο ή τέσσερα PUs (μονάδες πρόβλεψης) σύμφωνα με τον τύπο διαχωρισμού PU. Στο εσωτερικό ενός PU εφαρμόζεται η ίδια διαδικασία πρόβλεψης και οι σχετικές πληροφορίες μεταδίδονται στον αποκωδικοποιητή σε PU βάση. Μετά την απόκτηση του μπλοκ διαφορών (residual block) εφαρμόζοντας τη διαδικασία πρόβλεψης με βάση την διάσπαση τύπου PU, ένα φύλλο CU μπορεί πλέον να χωριστεί σε μονάδες μετασχηματισμού (TUs) σύμφωνα με μια άλλη δομή τετραδικών δέντρων παρόμοια με αυτή του δέντρου κωδικοποίησης για το CU.

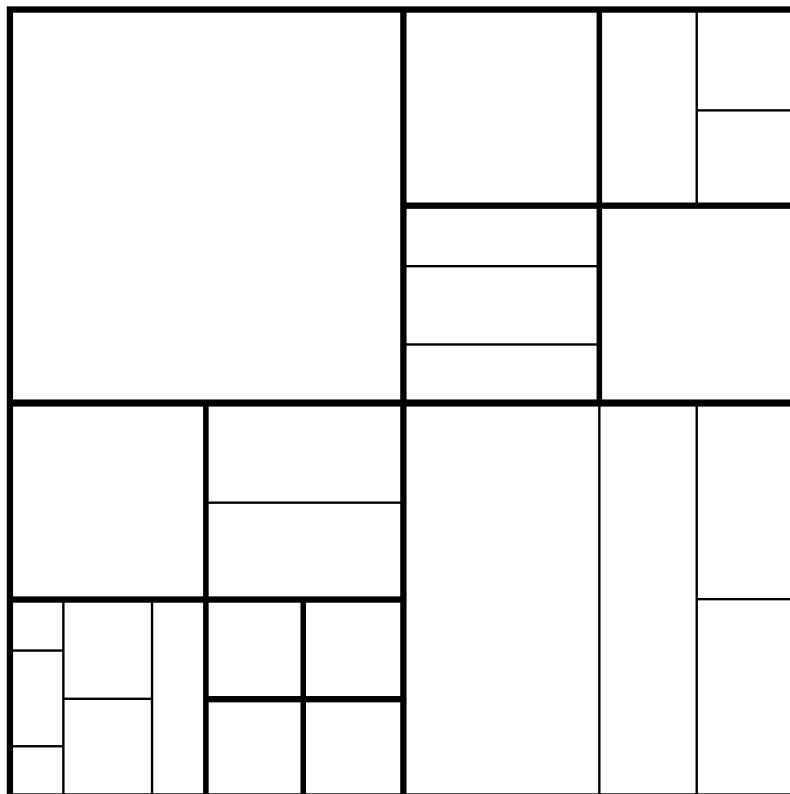
Στο VVC, ένα τετραδικό δέντρο με εμφωλευμένο δέντρο πολλαπλού τύπου που χρησιμοποιεί δυαδική και τριαδική κατάτμηση αντικαθιστά τις έννοιες των μονάδων πολλαπλού τύπου διαμερισμού, αφαιρεί δηλαδή τον διαχωρισμό μεταξύ των εννοιών CU, PU και TU. Αυτό δεν ισχύει στις περιπτώσεις που τα CUs που έχουν μέγεθος πολύ μεγάλο για το μέγιστο μήκος μετασχηματισμού. Η καινούργια δομή, μαζί με την εξάλειψη, στην πλειονότητα των περιπτώσεων, των -διαφορετικών- εννοιών CU, PU και TU, έχει σαν αποτέλεσμα τη μεγαλύτερη ευελιξία για τα σχήματα διαμερισμού CU. Έτσι, ένα CU μπορεί να έχει είτε τετράγωνο είτε ορθογώνιο σχήμα και το μέγεθός του μπορεί να είναι ίσο με του CTU ή τόσο μικρό όσο 4 x 4 σε μονάδες δειγμάτων luma. Ένα CTU διαχωρίζεται αρχικά από μια δομή τετραδικού δέντρου. Στη συνέχεια τα φύλλα κόμβοι του τετραδικού δέντρου μπορούν να διαχωριστούν επιπλέον μέσω μιας δομής δέντρου πολλαπλού τύπου. Υπάρχουν τέσσερις τύποι διαχωρισμού στη δομή δέντρου πολλαπλού τύπου, ο οριζόντιος δυαδικός διαχωρισμός (SPLIT_BT_HOR), ο κάθετος δυαδικός διαχωρισμός (SPLIT_BT_VER), ο οριζόντιος

τριαδικός διαχωρισμός (SPLIT_TT_HOR) και ο κάθετος τριαδικός διαχωρισμός (SPLIT_TT_VER) όπως φαίνεται και στην Εικόνα 6.



Εικόνα 6 -Τρόποι διάσπασης δέντρου πολλαπλού τύπου

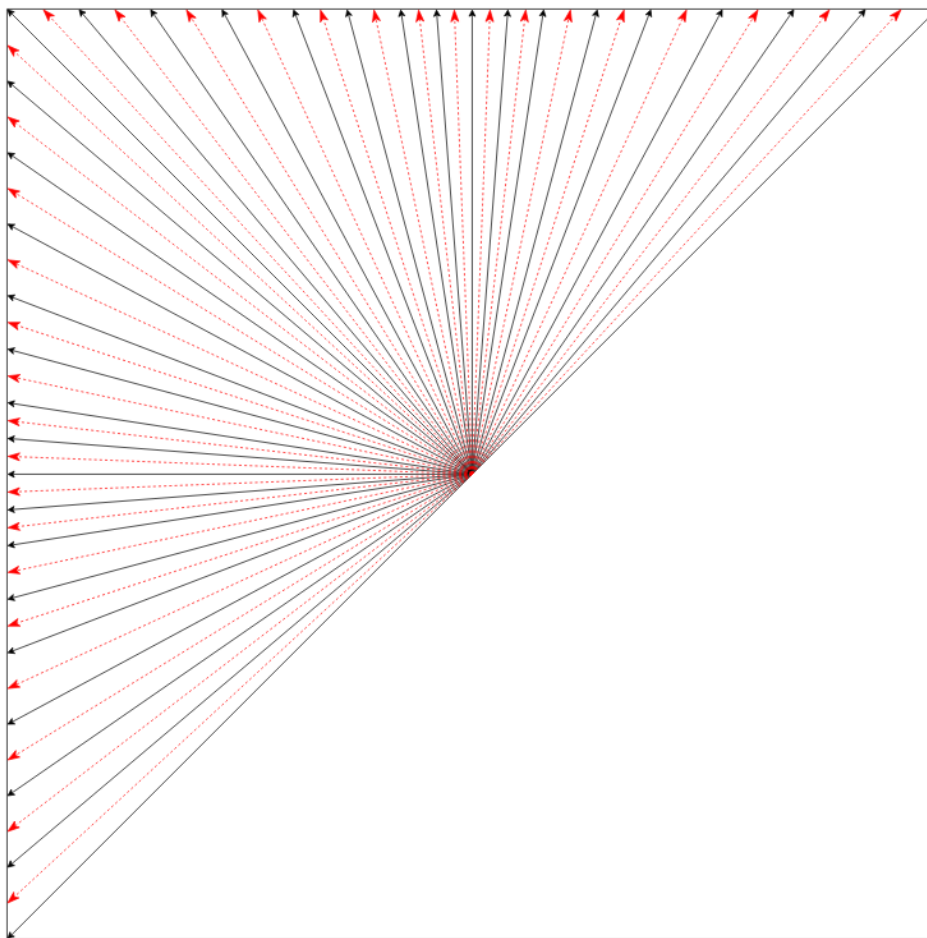
Στις περισσότερες περιπτώσεις, τα CU, PU και TU έχουν το ίδιο μέγεθος μέσα στο τετραδικό δέντρο με εμφωλευμένη δομή δέντρου κωδικοποίησης πολλαπλού τύπου. Εξαιρέση αποτελεί όταν το μέγιστο μήκος μετασχηματισμού που μπορεί να υποστηριχθεί είναι μικρότερο από το ύψος ή το πλάτος των στοιχείων χρώματος του CU. Στην Εικόνα 7 φαίνεται ένα CTU χωρισμένο σε πολλαπλά CUs μέσω ενός τετραδικού δέντρου και δομή εμφωλευμένου πολλαπλού τύπου δέντρου κωδικοποίησης, όπου οι άκρες των μπλοκ που είναι πιο έντονες αναπαριστούν διαχωρισμό τετραδικού δέντρου και οι υπόλοιπες άκρες μπλοκ αναπαριστούν διαχωρισμό δέντρου πολλαπλού τύπου.



Εικόνα 7 -Παράδειγμα τετραδικού δέντρου με δομή εμφωλευμένου δέντρου κωδικοποίησης πολλαπλού τύπου

Λειτουργία intra κωδικοποίησης με 67 intra prediction modes

Για να μπορέσουν να καταγραφούν οι αυθαίρετες κατευθύνσεις των ακμών που βλέπουμε στο φυσικό βίντεο, αυξήθηκε ο αριθμός των κατευθυντήριων intra modes στο VVC από 33 που ήταν στο HEVC στις 65. Επομένως στο VVC μαζί με τις λειτουργίες DC και planar που έχουν παραμείνει ίδιες από το HEVC, το σύνολο των intra prediction modes είναι 67. Τα νέα κατευθυντήρια intra modes του VVC που δεν συμπεριλαμβάνονται στο HEVC απεικονίζονται με κόκκινα διακεκομμένα βέλη στην Εικόνα 8 και οι υπόλοιπες απεικονίζονται με μαύρα βέλη. Στο VVC αρκετές λειτουργίες γωνιακού intra prediction αντικαταστάθηκαν με λειτουργίες intra prediction ευρείας γωνίας για τα μη τετράγωνα μπλοκ.



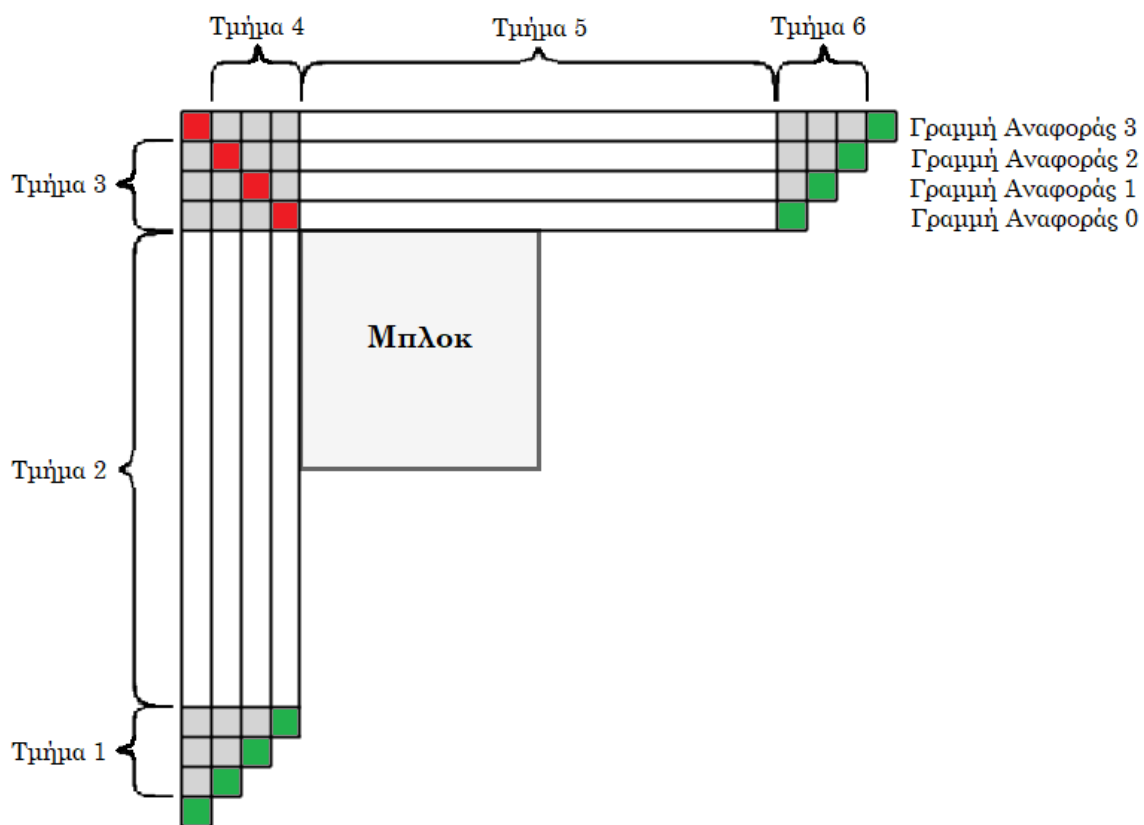
Εικόνα 8 –Κατευθυντήρια intra prediction modes

Intra prediction πολλαπλών γραμμών αναφοράς (MRL)

Ένα ακόμα νέο χαρακτηριστικό του VVC είναι το MRL (πολλαπλές γραμμές αναφοράς) intra prediction το οποίο μπορεί να χρησιμοποιήσει περισσότερες από μια γραμμή αναφοράς για να πραγματοποιήσει το intra prediction. Στην Εικόνα 9 απεικονίζεται ένα παράδειγμα τεσσάρων γραμμών αναφοράς, όπου τα δείγματα των τμημάτων 1 και 6 δεν αντλούνται από ανασκευασμένα γειτονικά δείγματα αλλά από τα πλησιέστερα δείγματα από το τμήμα 2 και 5 αντίστοιχα. Στην intra-picture πρόβλεψη του HEVC χρησιμοποιείται η πλησιέστερη

γραμμή αναφοράς (γραμμή αναφοράς 0). Στο MRL χρησιμοποιούνται δυο επιπλέον γραμμές αναφοράς (η γραμμή αναφοράς 1 και η γραμμή αναφοράς 3).

Ο δείκτης της επιλεγμένης γραμμής αναφοράς (*mrl_idx*) σηματοδοτείται και χρησιμοποιείται για την δημιουργία του *intra predictor*. Ο δείκτης της γραμμής αναφοράς σηματοδοτείται πριν από τις λειτουργίες *intra prediction*. Οι λειτουργίες Planar και DC εξαιρούνται από τις λειτουργίες *intra prediction* σε περίπτωση που σηματοδοτηθεί ένας μη μηδενικός δείκτης γραμμής αναφοράς. Το MRL είναι απενεργοποιημένο για την πρώτη γραμμή των μπλοκ εντός ενός CTU προκειμένου να αποφευχθεί η χρήση εκτεταμένων δειγμάτων αναφοράς που βρίσκονται εκτός των ορίων του CTU.



Εικόνα 9 -Παράδειγμα τεσσάρων γραμμών αναφοράς που γειτονεύουν με ένα μπλοκ πρόβλεψης

Inter Prediction

Πέρα από τα χαρακτηριστικά *inter* κωδικοποίησης του HEVC, στο VVC περιλαμβάνονται και αρκετά νέα και εξελιγμένα εργαλεία *inter prediction* κωδικοποίησης, όπως τα ακόλουθα:

- Πρόβλεψη εκτεταμένης συγχώνευσης
- Λειτουργία συγχώνευσης με MVD (MMVD)
- Συμμετρική κωδικοποίηση MVD
- Πρόβλεψη αντισταθμισμένης συνδετικής κίνησης
- Πρόβλεψη διανύσματος χρονικής κίνησης με βάση υπό-μπλοκ (SbTMVP)
- Προσαρμοσμένη ανάλυση διανυσμάτων κίνησης (AMVR)
- Αποθήκευση πεδίου κίνησης
- Βελτίωση του διανύσματος κίνησης από πλευράς αποκωδικοποιητή

Πρόβλεψη εκτεταμένης συγχώνευσης

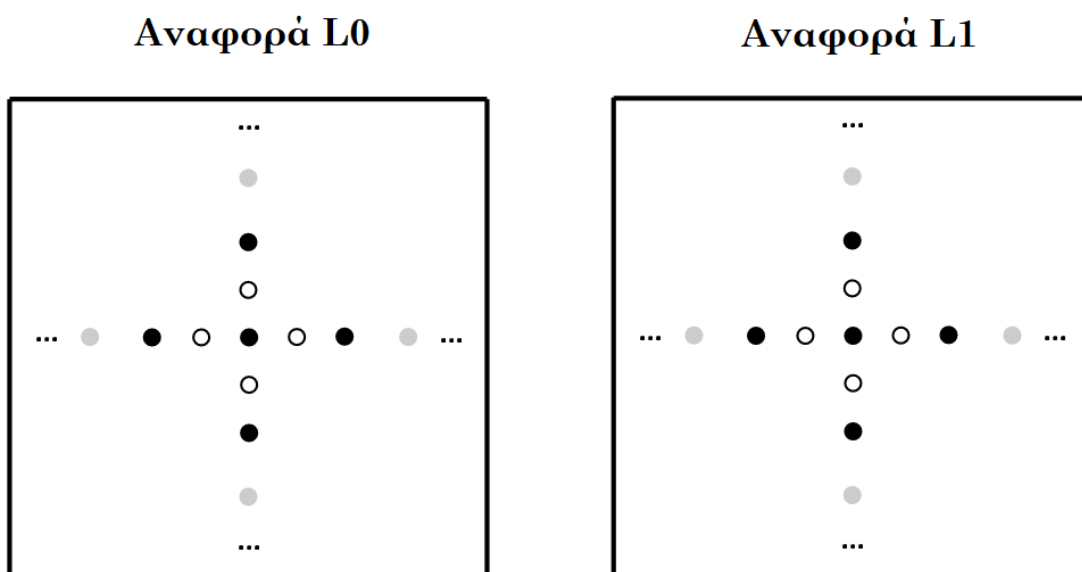
Στο VVC ο κατάλογος υποψηφίων (προβλεπόμενων διανυσμάτων κίνησης) συγχώνευσης κατασκευάζεται συμπεριλαμβάνοντας τους ακόλουθους πέντε τύπους υποψηφίων προβλεπόμενων διανυσμάτων κίνησης (Motion Vector Predictors - MVPs):

1. Το χωρικό MVP από τα γειτονικά CUs
2. Χρονικό MVP από τα collocated CUs
3. History – based MVP από έναν πίνακα FIFO
4. Pairwise average MVP
5. Μηδενικά MVs

Για κάθε κωδικό CU σε λειτουργία συγχώνευσης, ένας δείκτης καλύτερου υποψηφίου συγχώνευσης κωδικοποιείται χρησιμοποιώντας την λειτουργία truncated unary binarization (TU). Το μέγιστο επιτρεπόμενο μέγεθος της λίστας συγχώνευσης είναι έξι και το μέγεθός της σηματοδοτείται στην επικεφαλίδα του slice.

Λειτουργία συγχώνευσης με διαφορές διανύσματος κίνησης MVD (MMVD)

Εκτός από την λειτουργία συγχώνευσης, όπου οι έμμεσα παραγόμενες πληροφορίες κίνησης χρησιμοποιούνται απευθείας για τη δημιουργία δειγμάτων πρόβλεψης του τρέχοντος CU, επιπλέον εισάγεται στο VVC η λειτουργία συγχώνευσης με διαφορές στα διανύσματα κίνησης (MMVD). Μια σημαία MMVD σηματοδοτείται αμέσως μετά την αποστολή μιας σημαίας παράλειψης και μιας σημαίας συγχώνευσης προκειμένου να προσδιοριστεί εάν χρησιμοποιείται η MMVD λειτουργία για ένα CU. Στο MMVD, αφού επιλεγεί ένας υποψήφιος συγχώνευσης, ο υποψήφιος αυτός επαναπροσδιορίζεται περαιτέρω από τις σηματοδοτημένες πληροφορίες MVD οι οποίες περιλαμβάνουν μια σημαία υποψηφίου συγχώνευσης, έναν δείκτη για τον προσδιορισμό της κατεύθυνσης της κίνησης και έναν δείκτη για τον προσδιορισμό του μεγέθους της κίνησης. Στην λειτουργία MMVD ένας από τους πρώτους δύο υποψηφίους στη λίστα συγχώνευσης επιλέγεται για να χρησιμοποιηθεί ως βάση. Η σημαία υποψηφίου συγχώνευσης σηματοδοτείται για να προσδιοριστεί ποιος από όλους χρησιμοποιείται. Ο δείκτης απόστασης προσδιορίζει τις πληροφορίες του μεγέθους της κίνησης και υποδεικνύει την προκαθορισμένη μετατόπιση από το σημείο εκκίνησης. Όπως φαίνεται και από την Εικόνα 10, μια μετατόπιση προστίθεται είτε στην οριζόντια συνιστώσα είτε στην κάθετη συνιστώσα του αρχικού MV (βάση).



Εικόνα 10 - Σημείο Αναζήτησης MMVD

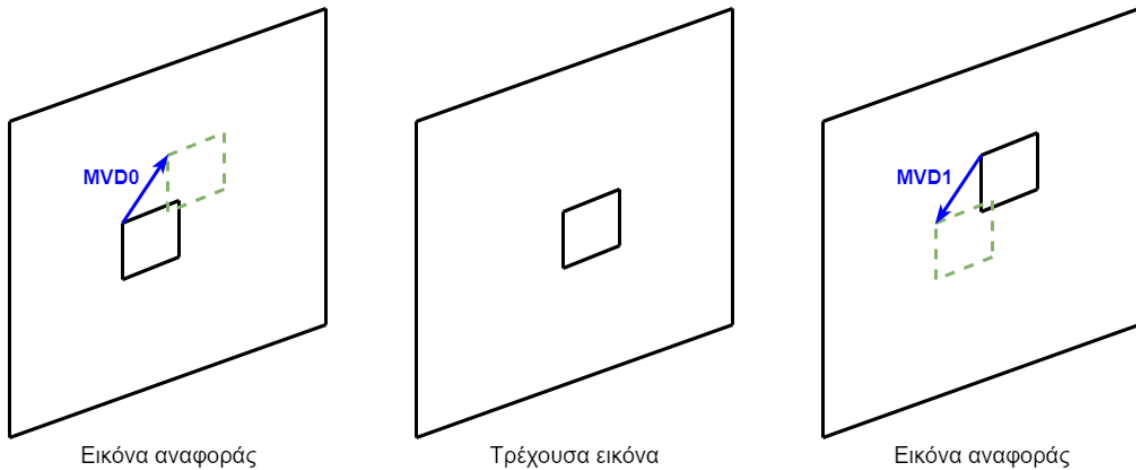
Συμμετρική κωδικοποίηση MVD

Στο VVC εκτός από την κανονική λειτουργία πρόβλεψης MVD σηματοδότησης μίας κατεύθυνσης και αμφίδρομης κατεύθυνσης, εφαρμόζεται και η λειτουργία συμμετρικού MVD για την αμφίδρομη πρόβλεψη σηματοδότησης MVD. Στην λειτουργία συμμετρικού MVD οι πληροφορίες κίνησης, συμπεριλαμβανομένων των δεικτών της εικόνας αναφοράς των λιστα-0 και λιστα-1, αλλά και του MVD της λιστα-1, δεν σηματοδοτούνται αλλά δημιουργούνται (από σύνθεση ήδη υπάρχουσας πληροφορίας).

Η διαδικασία αποκωδικοποίησης της λειτουργίας συμμετρικού MVD είναι η ακόλουθη:

- Στο επίπεδο του slice, οι μεταβλητές BiDirPredFlag, RefIdxSymL0 και RefIdxSymL1 καθορίζονται ως εξής:
 - Αν η μεταβλητή `mvd_l1_zero_flag` ισούται με 1, τότε η μεταβλητή BiDirPredFlag ορίζεται 0
 - Διαφορετικά αν η κοντινότερη εικόνα αναφοράς στην λιστα-0 και η κοντινότερη εικόνα αναφοράς στην λιστα-1 σχηματίζουν ένα ζευγάρι εικόνων αναφοράς με κατεύθυνση προς τα εμπρός και προς τα πίσω ή ένα ζευγάρι εικόνων αναφοράς με κατεύθυνση προς τα πίσω και προς τα εμπρός, τότε η μεταβλητή BiDirPredFlag ορίζεται 1 και οι εικόνες αναφοράς λιστα-0 και λιστα-1 θεωρούνται βραχυπρόθεσμες εικόνες αναφοράς
 - Διαφορετικά BiDirPredFlag ορίζεται 0
- Στο επίπεδο του CU μια σημαία συμμετρίας υποδεικνύει αν χρησιμοποιείται συμμετρική μέθοδος ή όχι και σηματοδοτείται ρητά (η λειτουργία σημαίας συμμετρίας παίρνει την τιμή true) αν το CU είναι κωδικοποιημένο με αμφίδρομη πρόβλεψη και αν το BiDirPredFlag ισούται με 1

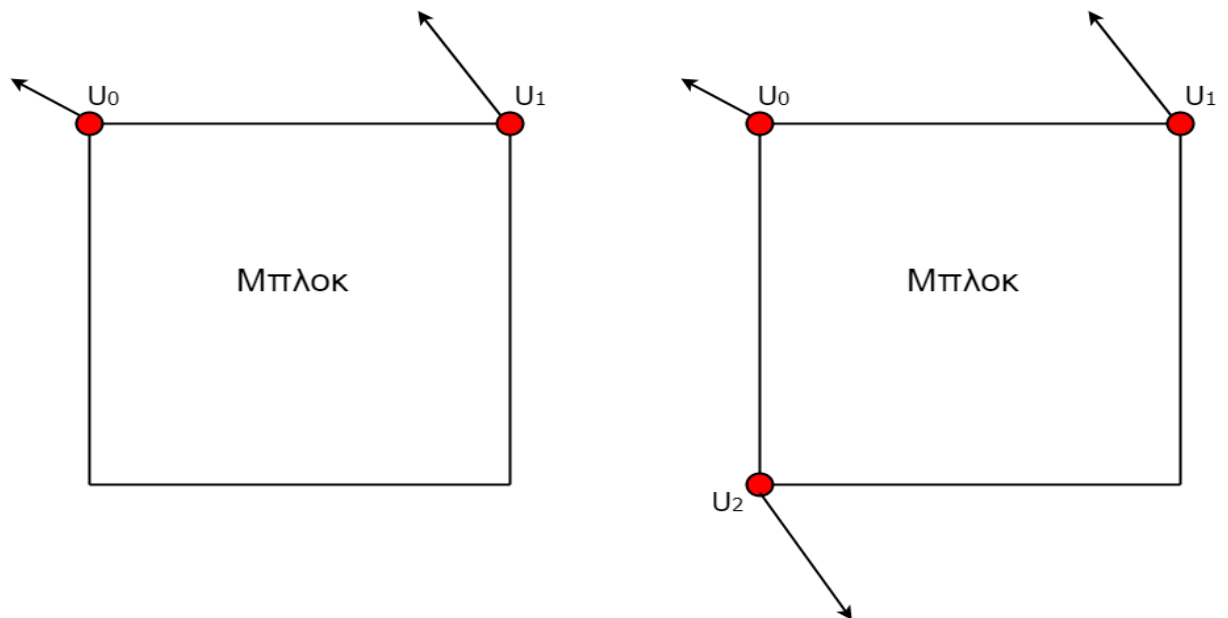
Όταν η σημαία συμμετρίας έχει την τιμή true, μόνο τα `mvp_l0_flag`, `mvp_l1_flag` και MVD0 σηματοδοτούνται ρητά. Οι δείκτες αναφοράς για τις εικόνες αναφοράς λιστα-0 και λιστα-1 θέτονται ίσοι με το ζεύγος εικόνων αναφοράς αντιστοίχα. Το MVD1 ορίζεται ίσο με το (-MVD0).



Εικόνα 11 – Απεικόνιση της λειτουργίας συμμετρικού MVD

Πρόβλεψη αντισταθμισμένης συνδετικής κίνησης – Affine motion compensated prediction

Ενώ στον πραγματικό κόσμο υπάρχουν πολλά είδη κινήσεων, όπως η μεγέθυνση/σμίκρυνση, η περιστροφή και άλλες ακανόνιστες κινήσεις, στο HEVC εφαρμόζεται μόνο το μοντέλο μετακίνησης (translation motion model) για την πρόβλεψη και την αντιστάθμιση κίνησης. Στο VVC η πρόβλεψη αντισταθμισμένης κίνησης γίνεται σε επίπεδο μπλοκ και χρησιμοποιεί το μοντέλο συνδετικού μετασχηματισμού (affine transform). Όπως φαίνεται και στην Εικόνα 11 το πεδίο συνδετικής κίνησης (affine motion field) του μπλοκ περιγράφεται από πληροφορίες κίνησης διανυσμάτων δύο σημείων ελέγχου (τεσσάρων παραμέτρων) ή τριών σημείων ελέγχου (έξι παραμέτρων).

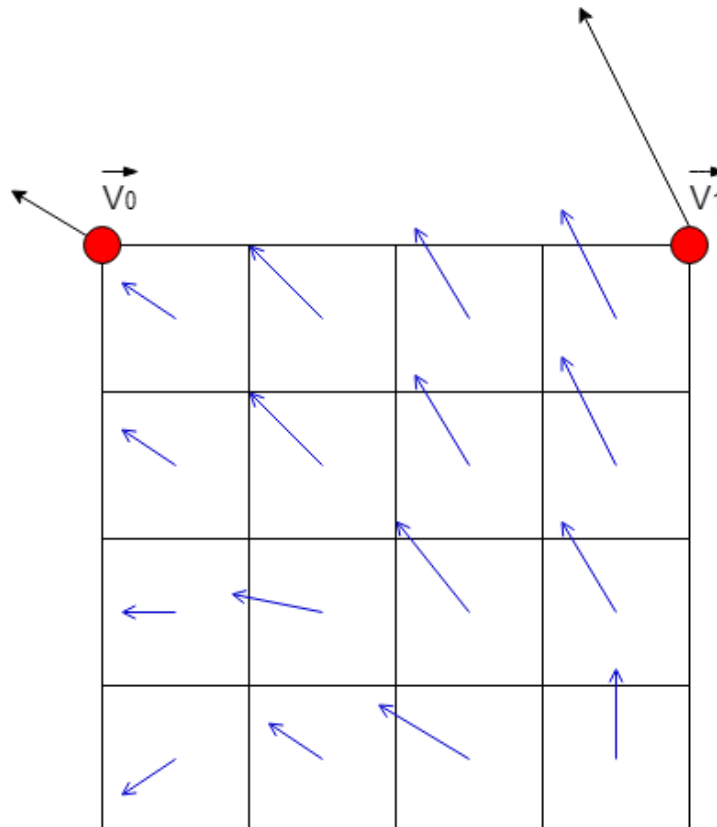


(α) Αφινικό μοντέλο τεσσάρων παραμέτρων

(β) Αφινικό μοντέλο έξι παραμέτρων

Εικόνα 12 – Μοντέλο συνδετικής κίνησης βασισμένο στα σημεία ελέγχου

Προκειμένου να απλοποιηθεί η πρόβλεψη της αντιστάθμισης κίνησης, εφαρμόστηκε η πρόβλεψη συνδεδειγμένου μετασχηματισμού με βάση το μπλοκ. Για να παραχθεί διάνυσμα κίνησης για κάθε υπό-μπλοκ luma 4x4, το διάνυσμα κίνησης του κεντρικού δείγματος κάθε υπο-μπλοκ υπολογίζεται, όπως φαίνεται και στην Εικόνα 13. Στη συνέχεια εφαρμόζονται τα φίλτρα παρεμβολής αντιστάθμισης κίνησης για να παραχθεί η πρόβλεψη του κάθε υπό-μπλοκ με βάση το παραγόμενο διάνυσμα κίνησης. Το μέγεθος του υπό-μπλοκ των ουσιαστικών chroma είναι επίσης 4x4. Το MV ενός υπό-μπλοκ chroma 4x4 υπολογίζεται ως ο μέσος όρος των MVs των τεσσάρων αντίστοιχων υπό-μπλοκ luma 4x4.



Εικόνα 13 – Συνδεδειγμένο MVF (affine MVF) ανά υπό-μπλοκ

Χρονική πρόβλεψη διανύσματος κίνησης με βάση υπό-μπλοκ (SbTMVP)

Η χρονική πρόβλεψη διανύσματος κίνησης με βάση υπό-μπλοκ (SbTMVP) είναι ακόμη μία από τις νέες μεθόδους που υποστηρίζονται στο VVC. Παρόμοια με την χρονική πρόβλεψη διανύσματος κίνησης (TMVP) του HEVC, η SbTMVP χρησιμοποιεί το πεδίο κίνησης στην collocated εικόνα για να βελτιώσει τη λειτουργία πρόβλεψης και συγχώνευσης διανυσμάτων κίνησης για τα CUs στην τρέχουσα εικόνα. Οι κύριες πτυχές στις οποίες παρατηρούμε διαφορές μεταξύ του SbTMVP και του TMVP είναι οι εξής:

- Το TMVP προβλέπει κίνηση σε επίπεδο CU, αλλά το SbTMVP προβλέπει κίνηση σε επίπεδο sub-CU
- Ενώ το TMVP ανακτά τα χρονικά διανύσματα κίνησης από το collocated μπλοκ στην collocated εικόνα (collocated μπλοκ λέγεται το μπλοκ που βρίσκεται κεντρικά ή κάτω δεξιά σε σχέση με το τρέχον CU), το SbTMVP εφαρμόζει μια μετατόπιση κίνησης προτού ανακτήσει τις πληροφορίες χρονικής κίνησης από την collocated

εικόνα. Αυτή η μετατόπιση κίνησης λαμβάνεται από το διάστημα κίνησης που βρίσκεται σε ένα από τα χωρικά γειτονικά μπλοκ του τρέχοντος CU.

Προσαρμοσμένη ανάλυση διανυσμάτων κίνησης (AMVR)

Στο HEVC οι διαφορές μεταξύ του διανύσματος κίνησης και του προβλεπόμενου διανύσματος κίνησης ενός CU ονομάζονται MVDs και σηματοδοτούνται σε μονάδες του ενός τετάρτου του δείγματος luma, όταν η μεταβλητή `use_integer_mv_flag` ισούται με το 0 στην επικεφαλίδα του slice. Μια ακόμη νέα προσθήκη του VVC είναι ένα σχήμα προσαρμοστικής ανάλυσης διανυσμάτων κίνησης σε επίπεδο CU που ονομάζεται AMVR. Το AMVR επιτρέπει στο MVD της CU να κωδικοποιείται με διαφορετική ακρίβεια. Ανάλογα με την AMVP λειτουργία (κανονική ή συνήθως συνδετική (affine)) για το τρέχον CU, οι MVDs του τρέχοντος CU μπορούν να επιλεγθούν προσαρμοστικά ως εξής:

- Κανονική AMVP λειτουργία: ένα τέταρτο του δείγματος luma, μισό δείγμα luma, ακέραιο δείγμα luma ή δείγμα τεσσάρων luma.
- Συνδετική (affine) AMVP λειτουργία: ένα τέταρτο του δείγματος luma, ακέραιο δείγμα luma ή ένα δέκατο έκτο του δείγματος luma.

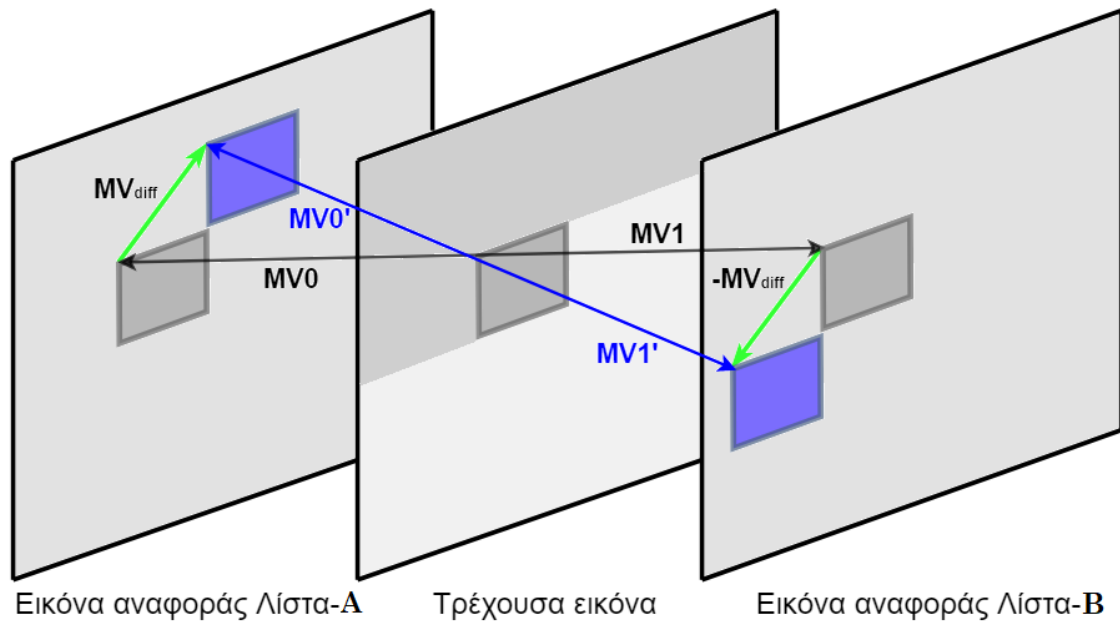
Η ένδειξη ανάλυσης MVD σε επίπεδο CU σηματοδοτείται υπό τον όρο ότι το τρέχον CU έχει τουλάχιστον ένα μη μηδενικό MVD στοιχείο. Εάν όλα τα MVD στοιχεία είναι μηδενικά, προκύπτει η MVD ανάλυση του ενός τετάρτου του δείγματος luma.

Αποθήκευση πεδίου κίνησης

Στο VVC η υψηλότερη ακρίβεια των ρητά σηματοδοτημένων διανυσμάτων κίνησης είναι το δείγμα luma του ενός τετάρτου. Σε μερικές λειτουργίες inter prediction όπως η συνδετική (affine) λειτουργία, τα διανύσματα κίνησης προέρχονται από ακρίβεια δείγματος luma του ενός δέκατου έκτου και εκτελείται πρόβλεψη αντισταθμισμένης κίνησης με ακρίβεια δείγματος της τάξεως ένα προς δεκαέξι. Από τη άποψη της εσωτερικής αποθήκευσης του πεδίου κίνησης, όλα τα διανύσματα κίνησης αποθηκεύονται με ακρίβεια ένα προς δεκαέξι δείγματος luma. Στο VVC για την προσωρινή αποθήκευση του πεδίου κίνησης που χρησιμοποιείται από τα TMVP και ATVMP, εκτελείται συμπίεση του πεδίου κίνησης σε βαθμό λεπτομέρειας 8x8, σε αντίθεση με την συμπίεση του πεδίου κίνησης του HEVC που εκτελείται σε βαθμό λεπτομέρειας 16x16.

Βελτίωση του διανύσματος κίνησης από πλευράς αποκωδικοποιητή (DMVR)

Στο VVC προκειμένου να αυξηθεί η ακρίβεια των MVs της λειτουργίας συγχώνευσης, εφαρμόστηκε μια λειτουργία βελτίωσης του διανύσματος κίνησης από πλευράς αποκωδικοποιητή βάσει διμερής αντιστοιχίας. Στη λειτουργία αμφίδρομης πρόβλεψης ένα βελτιωμένο MV αναζητείται γύρω από τα αρχικά MVs στις εικόνες αναφοράς λίστα-A και λίστα-B. Μέσω της μεθόδου BM υπολογίζεται η παραμόρφωση που υπάρχει μεταξύ των δύο υποψήφιων μπλοκ στις εικόνες αναφοράς λίστα-A και λίστα-B. Όπως απεικονίζεται και στην Εικόνα 14, υπολογίζεται το SAD (μέτρηση ομοιότητας μπλοκ εικόνας) μεταξύ των μπλε μπλοκ με βάση κάθε υποψήφιο MV γύρω από το αρχικό MV και ο υποψήφιος MV με το λιγότερο SAD γίνεται το βελτιωμένο MV και χρησιμοποιείται για να παραχθεί το σήμα αμφίδρομης πρόβλεψης.



Εικόνα 14 - Βελτίωση του διανύσματος κίνησης από πλευράς αποκωδικοποιητή

Στο VVC το DMVR μπορεί να εφαρμοστεί για τα CUs τα οποία έχουν κωδικοποιηθεί με τα ακόλουθα χαρακτηριστικά και λειτουργίες:

- Το WP (Weighted Prediction) δεν είναι ενεργοποιημένο για το τρέχων μπλοκ
- Το CU έχει περισσότερα από 64 δείγματα luma
- Λειτουργία συγχώνευσης με αμφίδρομη πρόβλεψη MV σε επίπεδο CU
- Μία εικόνα αναφοράς είναι στο παρελθόν και μια άλλη εικόνα αναφοράς είναι στο μέλλον σε σχέση με την τρέχουσα εικόνα
- Η λειτουργία CIIP (Combined Inter and Intra Prediction) δεν χρησιμοποιείται για το τρέχων μπλοκ
- Οι αποστάσεις μεταξύ της τρέχουσας εικόνας και δύο εικόνων αναφοράς είναι ίδιες
- Το ύψος και το πλάτος του CU είναι μεγαλύτερα ή ίσα των οκτώ δειγμάτων luma
- Ο δείκτης βάρους BCD δείχνει ίσο βάρος
- Και οι δύο εικόνες αναφοράς είναι βραχυπρόθεσμες εικόνες αναφοράς

Το βελτιωμένο MV που προέρχεται από την διαδικασία DMVR χρησιμοποιείται στην πρόβλεψη χρονικών διανυσμάτων κίνησης για την κωδικοποίηση μελλοντικών εικόνων και στην δημιουργία των δειγμάτων inter-prediction, ενώ το αρχικό MV χρησιμοποιείται στην πρόβλεψη χωρικών διανυσμάτων κίνησης για την κωδικοποίηση μελλοντικών CU και στην διαδικασία αποδέσμευσης.

Μετασχηματισμός

Στο VVC υπάρχει η δυνατότητα μετασχηματισμού μπλοκ μεγέθους μέχρι και 64x64, πράγμα το οποίο χρησιμεύει κυρίως σε βίντεο υψηλότερης ανάλυσης όπως για παράδειγμα τα βίντεο 1080p και 4K. Οι συντελεστές μετασχηματισμού υψηλής συχνότητας μηδενίζονται για τα μπλοκ μετασχηματισμού με μέγεθος (ύψους ή πλάτους, ή ύψους και πλάτους) ίσο του 64, έτσι ώστε να διατηρούνται μόνο οι συντελεστές χαμηλότερης συχνότητας. Για παράδειγμα, ένα μπλοκ μετασχηματισμού μεγέθους $M \times N$, όπου M το

πλάτος και N το ύψος του μπλοκ, όταν το N είναι ίσο με 64, μόνο οι 32 πάνω σειρές των συντελεστών μετασχηματισμού κρατούνται. Ομοίως, όταν το M είναι ίσο με 64, μόνο οι 32 αριστερές στήλες των συντελεστών μετασχηματισμού κρατούνται. Όταν χρησιμοποιείται η λειτουργία παράλειψης μετασχηματισμού για ένα μπλοκ μεγάλου μεγέθους, ολόκληρο το μπλοκ χρησιμοποιείται χωρίς να μηδενίζονται τιμές. Ένα ακόμα νέο χαρακτηριστικό είναι ότι υποστηρίζεται ρυθμιζόμενο μέγιστο μέγεθος μετασχηματισμού στο SPS, έτσι ώστε ο κωδικοποιητής να έχει την ευελιξία να επιλέξει το μέγεθος του μετασχηματισμού να φτάνει μέχρι το μήκος 16, 32 ή 64 αναλόγως τις ανάγκες της κάθε εκτέλεσης. Επιπλέον εκτός από το DCT-II που έχει χρησιμοποιηθεί στο HEVC, χρησιμοποιείται και ένα σχήμα πολλαπλής επιλογής μετασχηματισμού (MTS) για την κωδικοποίηση των διαφορών των inter και των intra μπλοκ, το οποίο σχήμα χρησιμοποιεί πολλαπλούς επιλεγμένους μετασχηματισμούς από τους DST-VII και DCT-VIII

Κβαντισμός

Στο VVC, το μέγιστο QP επεκτάθηκε από 51 σε 63 και ανάλογα άλλαξε και η σηματοδότηση της αρχικής QP. Η αρχική τιμή του SliceQpY τροποποιείται στο επίπεδο του τμήματος slice όταν κωδικοποιείται μια μη μηδενική τιμή του slice_qp_delta. Για την παράλειψη μπλοκ μετασχηματισμού η ελάχιστη επιτρεπόμενη παράμετρος κβαντισμού (QP) ορίζεται 4 επειδή το μέγεθος του βήματος κβαντισμού γίνεται 1 όταν το QP ισούται με 4.

Στο HEVC αλλά και στο H.264, χρησιμοποιείται ένας σταθερός πίνακας αναζήτησης για την μετατροπή της παραμέτρου κβαντισμού luma QP_Y σε παραμέτρους κβαντισμού chroma QP_C. Στο VVC χρησιμοποιείται μια πιο ευέλικτη αντιστοιχία luma-to-chroma (από παραμέτρους luma σε παραμέτρους chroma) QP. Αντί να υπάρχει ένας σταθερός πίνακας, η σχέση αντιστοιχίας luma-to-chroma QP σηματοδοτείται στο SPS χρησιμοποιώντας ένα ευέλικτο γραμμικό κατά τμήματα μοντέλο, με τον μοναδικό περιορισμό να έγκειται στο ότι στο γραμμικό μοντέλο η κλίση κάθε τμήματος δεν μπορεί να είναι αρνητική. Η σχέση αντιστοιχίας QP μπορεί να σηματοδοτηθεί ξεχωριστά για τις κωδικοποιήσεις Cb,Cr και την κοινή κωδικοποίηση Cb/Cr ή σηματοδοτείται από κοινού για τους τρεις τύπους υπολειπόμενης κωδικοποίησης.

Όπως και στο HEVC έτσι και στο νέο πρότυπο επιτρέπεται η προσαρμογή της παραμέτρου QP σε επίπεδο CU. Οι τιμές Delta QP για τα στοιχεία luma και chroma μπορούν να σηματοδοτηθούν ξεχωριστά. Για τα chroma στοιχεία, οι επιτρεπόμενες τιμές μετατόπισης chroma QP σηματοδοτούνται με τη μορφή λιστών μετατόπισης στο PPS. Οι λίστες ορίζονται ξεχωριστά για τις κωδικοποιήσεις Cb,Cr και την κοινή κωδικοποίηση Cb/Cr και για κάθε μία από αυτές επιτρέπονται έως και έξι τιμές μετατόπισης. Στο επίπεδο CU, ένας δείκτης σηματοδοτείται για να υποδείξει ποια από τις τιμές μετατόπισης στην λίστα μετατόπισης χρησιμοποιείται για να ρυθμίσει το chroma QP για αυτό το CU.

Φιλτράρισμα

Στο VVC υπάρχουν συνολικά τρία in-loop filters στα οποία βλέπουμε ορισμένες ομοιότητες συγκριτικά με το HEVC. Τα φίλτρα αυτά, με την σειρά που εκτελούνται από το VVC είναι το deblocking filter, το SAO και το adaptive loop filter (ALF). Το SAO παραμένει το ίδιο με αυτό που χρησιμοποιείται και στο HEVC. Επιπλέον υπάρχει ένα νέο φίλτρο που έχει προστεθεί στο VVC το οποίο ονομάζεται αντιστοιχία στοιχείων luma με κλιμάκωση chroma (luma mapping with chroma scaling – LMCS). Το νέο αυτό φίλτρο το οποίο ήταν παλαιότερα γνωστό ως adaptive in-loop reshapener, πραγματοποιείται στην εκτέλεση πριν το deblocking.

Στο VVC κατά την εφαρμογή του adaptive loop filter, για τα στοιχεία luma επιλέγεται ένα από τα 25 φίλτρα για κάθε μπλοκ 4 x 4, με κριτήριο την δραστηριότητα και την κατεύθυνση των τοπικών βαθμίδων. Η διαδικασία που εκτελείται κατά το deblocking filter είναι παρόμοια με αυτή στο HEVC με την διαφορά ότι έχουν προστεθεί ορισμένες μετατροπές για καλύτερα αποτελέσματα. Το φίλτρο LMCS που προστέθηκε στο VVC ως ένα νέο μπλοκ επεξεργασίας πριν από τα loop filters, έχει δύο κύρια στοιχεία. Το πρώτο είναι η αντιστοίχιση in-loop των στοιχείων luma, με κριτήριο τα προσαρμοστικά μερικώς γραμμικά μοντέλα. Το δεύτερο είναι η εφαρμογή της υπολειπόμενης κλίμακας chroma που εξαρτάται από τα στοιχεία luma, για τα στοιχεία chroma. Όπως και τα περισσότερα εργαλεία του VVC, το LMCS μπορεί να ενεργοποιηθεί ή να απενεργοποιηθεί χρησιμοποιώντας μια σημαία σε επίπεδο SPS.

Κωδικοποίηση Εντροπίας

Στο VVC το CABAC περιέχει τρεις κύριες αλλαγές συγκριτικά με το HEVC. Η πρώτη αλλαγή αφορά τον πυρήνα της μηχανής CABAC. Η δεύτερη είναι η δομή ξεχωριστής υπολειπόμενης κωδικοποίησης για τα μπλοκ μετασχηματισμού και τα μπλοκ παράλειψης μετασχηματισμού. Την τρίτη αλλαγή αποτελεί η μοντελοποίηση πλαισίου για συντελεστές μετασχηματισμού.

Η διαδικασία αρχικοποίησης του CABAC στο VVC εξαρτάται από τον αριθμό QP και επικαλείται στην αρχή του κάθε slice. Στο HEVC χρησιμοποιούνται μη αλληλεπικαλυπτόμενες ομάδες συντελεστών για την κωδικοποίηση των συντελεστών μετασχηματισμού του κάθε μπλοκ κωδικοποίησης και κάθε CG (ομάδα συντελεστών) περιέχει τους συντελεστές ενός μπλοκ μεγέθους 4 x 4 το οποίο αντλείται από το μπλοκ κωδικοποίησης. Ωστόσο στο VVC τα μεγέθη των ομάδων συντελεστών εξαρτώνται μόνο από το μέγεθος του TB, πράγμα που διακόπτει την εξάρτηση από τον τύπο του καναλιού. Οι CG ενός μπλοκ κωδικοποίησης καθώς και οι συντελεστές μετασχηματισμού που υπάρχουν εντός των CG, κωδικοποιούνται σύμφωνα με προκαθορισμένες εντολές σάρωσης. Ενώ στο HEVC η υπολειμματική κωδικοποίηση έχει σχεδιαστεί κυρίως για τα στατιστικά αποτελέσματα και για τα χαρακτηριστικά του σήματος των επιπέδων συντελεστών μετασχηματισμού, στο VVC χρησιμοποιούνται δύο διαφορετικές δομές υπολειμματικής κωδικοποίησης για τους συντελεστές μετασχηματισμού και τους συντελεστές παράλειψης μετασχηματισμού αντίστοιχα. Το VVC υποστηρίζει την λειτουργία παράλειψης μετασχηματισμού. Παρόμοια λειτουργία υποστηρίζεται και στο HEVC. Στην λειτουργία αυτή τα στατιστικά χαρακτηριστικά του σήματος διαφέρουν από εκείνα των συντελεστών μετασχηματισμού.

ΚΕΦΑΛΑΙΟ 4 Split Parallelism

Όπως είδαμε και στο Κεφάλαιο 3 της εργασίας ο κωδικοποιητής του VVC σπάει το κάθε καρέ σε CTU μπλοκ τα οποία φτάνουν το μέγεθος 128 x 128 pixel, αλλά στην συνέχεια μπορούν να σπάσουν σε ακόμα μικρότερα μπλοκ. Το κάθε μπλοκ κωδικοποιείται ξεχωριστά και οι μέθοδοι κωδικοποίησής τους μπορούν να είναι τελείως διαφορετικοί μεταξύ τους. Ο κωδικοποιητής επιλέγει αν στο CTU χρειάζεται περεταίρω διαίρεση καθώς και το βέλτιστο είδος διαίρεσης με βάση το περιεχόμενο του μπλοκ. Στις ενιαίες περιοχές της εικόνας τα μεγαλύτερα μπλοκ είναι πιο αποδοτικά, ενώ σε περιοχές που έχουν περισσότερη λεπτομέρεια τα μικρότερα μπλοκ είναι η καλύτερη επιλογή.

Η διαίρεση στο VVC γίνεται με την χρήση δύο δομών ιεραρχικών δέντρων. Η πρώτη είναι η δομή τετραδικού δέντρου στην οποία τα μπλοκ χωρίζονται σε τέσσερα τετράγωνα υπό-μπλοκ με το καθένα να έχει το μισό πλάτος και ύψος του αρχικού. Αν ένα μπλοκ δεν διαιρείται περεταίρω σε αυτή την δομή τότε εφαρμόζεται η δεύτερη δομή ιεραρχικού δέντρου. Στη δομή δέντρου πολλαπλού τύπου υπάρχουν διάφορες επιλογές για την διαίρεση του μπλοκ οι οποίες σηματοδοτούνται με την χρήση τριών σημαιών. Η πρώτη σημαία είναι η `mtt_split_cu_flag` και σηματοδοτεί αν θα γίνει επιπλέον διαίρεση στο μπλοκ. Η δεύτερη σημαία ονομάζεται `mtt_split_cu_vertical_flag` και υποδεικνύει αν ο διαχωρισμός θα είναι οριζόντιος ή κάθετος. Η `mtt_split_cu_binary_flag` υποδεικνύει αν ο διαχωρισμός θα είναι διπλός ή τριπλός. Τα μπλοκ που δεν διαιρούνται σε άλλα υπό-μπλοκ είναι τα CUs και σε αυτά γίνεται η κωδικοποίηση.

Η διαδικασία διαίρεσης του καρέ σε CTUs και η επιπλέον διάσπασή τους μέχρι τα μπλοκ να μην επιδέχονται άλλη διαίρεση, πραγματοποιείται από τον κωδικοποιητή του VVC με την χρήση των συναρτήσεων `Partitioner` που βρίσκονται στο αρχείο `UnitPartitioner.cpp` του αλγορίθμου.

Στην συνέχεια το κάθε μπλοκ που έχει δημιουργηθεί περνάει από την διαδικασία πρόβλεψης. Η διαδικασία πρόβλεψης χωρίζεται σε δύο κατηγορίες, την `intra prediction` και την `inter prediction`. Κατά την `intra prediction` όπως είδαμε και σε προηγούμενο κεφάλαιο το μπλοκ προβλέπεται χρησιμοποιώντας άλλα μπλοκ του ίδιου καρέ που έχουν ήδη ανασηματιστεί. Υπάρχουν αρκετά είδη `intra prediction` στο VVC μερικά εκ των οποίων αναλύσαμε σε προηγούμενο κεφάλαιο. Ο αλγόριθμος ανάλογα με τις ιδιομορφίες του τρέχοντος μπλοκ, συγκρίνει αυτά τα είδη πρόβλεψης και επιλέγει το αποδοτικότερο για να εφαρμοστεί στην προκειμένη περίπτωση. Στην `inter prediction` γίνεται πρόβλεψη με την χρήση μπλοκ από άλλα καρέ είτε προηγούμενων και επόμενων αν πρόκειται για B-frame, είτε μόνο προηγούμενων αν πρόκειται για P-frame. Εδώ και πάλι έχουμε αρκετά είδη πρόβλεψης μερικά από τα οποία αναλύσαμε και προηγουμένως. Όταν ο αλγόριθμος επιλέξει τον κατάλληλο τρόπο πρόβλεψης αλλά και το μπλοκ άλλου καρέ το οποίο θα χρησιμοποιήσει, παράγει το `Residual` μπλοκ. Το μπλοκ αυτό είναι το τρέχων μπλοκ που κωδικοποιείται, από το οποίο έχουν αφαιρεθεί τα κομμάτια που έχουν προβλεφθεί με την χρήση άλλων καρέ. Φυσικά προκειμένου να γίνει μετά η αποκωδικοποίηση κρατούνται οι αριθμοί των καρέ καθώς και οι συντεταγμένες των `prediction` μπλοκ.

Πολλές από της συναρτήσεις του κωδικοποιητή συνεργάζονται για την διαδικασία της πρόβλεψης. Οι κυριότερες αυτών των συναρτήσεων είναι οι `InterSearch`, `IntraSearch`, `Distortion`, `TrQuant`, `CABACWriter(residual_coding)`, `EncCu(xEncodeInterResidual)`, `InterPrediction` και `IntraPrediction`.

Το `Residual` μπλοκ στην συνέχεια παρόλο που είναι ήδη μικρότερου μεγέθους σε σχέση με το αρχικό μπλοκ, θα περάσει και από ένα ακόμα επίπεδο επεξεργασίας που ονομάζεται μετασηματισμός και κβαντισμός. Κατά το μετασηματισμό ο αλγόριθμος θα επιλέξει τον

καταλληλότερο για την περίπτωση τύπο μετασχηματισμού, όπως είναι ο DST και ο DCT. Το αποτέλεσμα του μετασχηματισμού θα διαιρεθεί με τον πίνακα κβαντισμού που δημιουργείται από την τιμή που θα δώσουμε στην μεταβλητή QP. Ο πίνακας κβαντισμού επηρεάζει σε μεγάλο βαθμό την ποιότητα της κωδικοποίησης καθώς και τον χρόνο που χρειάζεται για να ολοκληρωθεί ο αλγόριθμος. Οι κυριότερες συναρτήσεις που εκτελούν αυτό το κομμάτι της κωδικοποίησης είναι οι TransformUnit, fastForwardDCT2_B2 (και οι άλλες fastForward από το TrQuant_EMT.cpp), CABACWriter, Quantizer, DepQuant, Quant, QpParam.

Στην συνέχεια το αποτέλεσμα του κβαντισμού περνάει από την διαδικασία του in loop filter το οποίο χρησιμοποιείται για να καλύπτει πιθανά σφάλματα από την διαδικασία του κβαντισμού αλλά και στα όρια των pixel προκειμένου να έχουμε μια ομαλή μετάβαση από pixel σε pixel και ένα ομοιόμορφο αποτέλεσμα. Η διαδικασία αυτή επαναλαμβάνεται για όλα τα μπλοκ του καρέ μέχρι να έχουμε την καλύτερη δυνατή ποιότητα στην εικόνα μας. Το κομμάτι αυτό πραγματοποιείται μέσω των συναρτήσεων AdaptiveLoopFilter, AlfCovariance, EncAdaptiveLoopFilter.

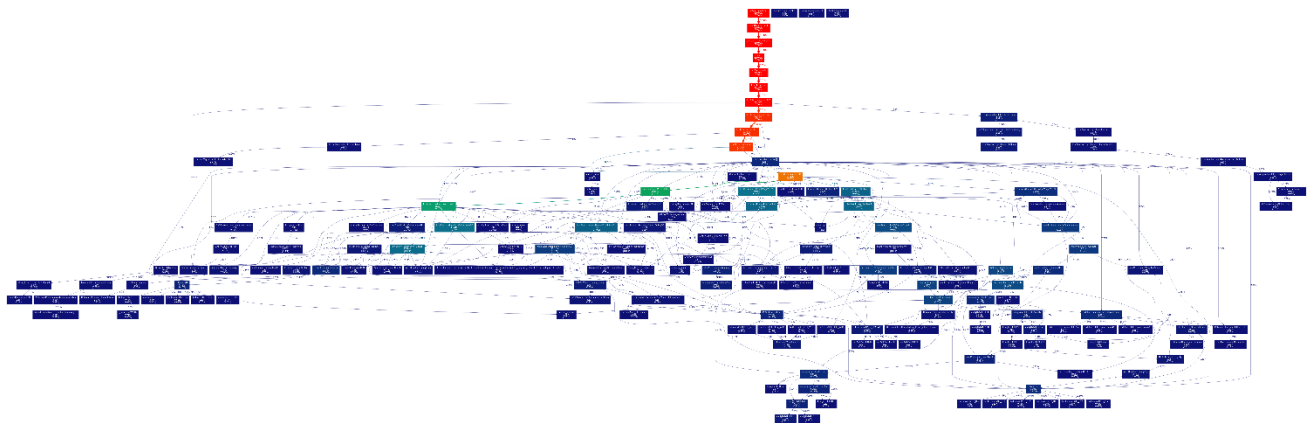
Για να φτάσουμε στο τελικό αποτέλεσμα του κωδικοποιητή του VVC, το αποτέλεσμα όλων των προηγούμενων φάσεων θα περάσει από κωδικοποίηση εντροπίας. Με αυτόν τον τρόπο το αποτέλεσμα συμπιέζεται χωρίς καθόλου επιπλέον απώλειες δεδομένων.

Προκειμένου να γίνει η κωδικοποίηση του βίντεο θα πρέπει να πραγματοποιηθεί η παραπάνω διαδικασία ανεξάρτητα από το αν η εκτέλεση του αλγορίθμου είναι σειριακή ή παράλληλη. Στην παράλληλη εκτέλεση η διαφορά είναι ότι προκειμένου να αυξηθεί η ταχύτητα ολοκλήρωσης του αλγορίθμου, δημιουργούνται διαφορετικά νήματα στα οποία μοιράζονται κατάλληλα τα κομμάτια της κωδικοποίησης. Τον αριθμό των νημάτων (NumSplitThreads) που δημιουργούνται για την παράλληλη εκτέλεση τον ορίζουμε εμείς από το config αρχείο του κωδικοποιητή του VVC, που εμπεριέχει και άλλες ρυθμίσεις για την εκτέλεση του κώδικα όπως τον αριθμό QP και το όνομα του αρχείου που θέλουμε να κωδικοποιήσουμε (InputFile). Αλλαγή στις ρυθμίσεις για την εκτέλεση του κώδικα μπορεί να γίνει και από το αρχείο TypeDef.h από το οποίο μπορούμε να επιλέξουμε και τον αριθμό των παράλληλων εργασιών που θα πραγματοποιηθούν. Οι παράλληλες εργασίες θα μοιραστούν στα νήματα που έχουν δημιουργηθεί ανάλογα με τον αριθμό τους. Αν για παράδειγμα έχουμε τον αριθμό 4 για τα νήματα και τον αριθμό 8 για τις παράλληλες εργασίες τότε το κάθε νήμα θα αναλάβει από 2 εργασίες, ενώ αν τα νήματα είναι 4 και οι εργασίες 6, τα 2 νήματα θα πάρουν από 2 εργασίες και τα άλλα 2 από 1. Ο αριθμός των εργασιών δεν επιτρέπεται να είναι μικρότερος από τον αριθμό των νημάτων, καθώς θα είχαμε νήματα στα οποία δεν θα υπήρχαν εργασίες για να τους ανατεθούν. Επιπλέον από το TypeDef.h γίνεται η ενεργοποίηση της παράλληλης εκτέλεσης ή όχι. Στους ορισμούς του TypeDef.h για να ενεργοποιηθεί ή να απενεργοποιηθεί η παράλληλη λειτουργία, αλλάζουμε το #define ENABLE_SPLIT_PARALLELISM σε 1 ή σε 0 αντίστοιχα και για τον ορισμό του αριθμού των παράλληλων εργασιών βάζουμε τον αριθμό που επιθυμούμε στο #define PARL_SPLIT_MAX_NUM_JOBS.

Αφού ρυθμίσουμε τον κωδικοποιητή όπως θέλουμε και αφού κάνουμε build το πρότζεκτ μπορούμε να παρατηρήσουμε τις αλλαγές που υπάρχουν στον κώδικα όταν τον έχουμε ρυθμίσει για παράλληλη εκτέλεση σε σχέση με την σειριακή. Η πιο εμφανής διαφορά στην ροή του κώδικα είναι στα σημεία του αλγορίθμου που υπάρχει η εντολή #if ENABLE_SPLIT_PARALLELISM. Σε εκείνα τα σημεία μέχρι και να κλείσει η εντολή if, στην περίπτωση που είναι ενεργοποιημένη η παράλληλη εκτέλεση ο κώδικας συνεχίζει την ροή του κανονικά. Στην περίπτωση που η παράλληλη εκτέλεση είναι απενεργοποιημένη, τα κομμάτια αυτά δεν περνούν από compile όταν γίνεται το “χτίσιμο” του πρότζεκτ προκειμένου να γίνει εξοικονόμηση πόρων. Έτσι τα σημεία αυτά παραλείπονται κατά την εκτέλεση του κώδικα. Το ίδιο αντίστοιχα γίνεται και σε σημεία του κώδικα που πρέπει να

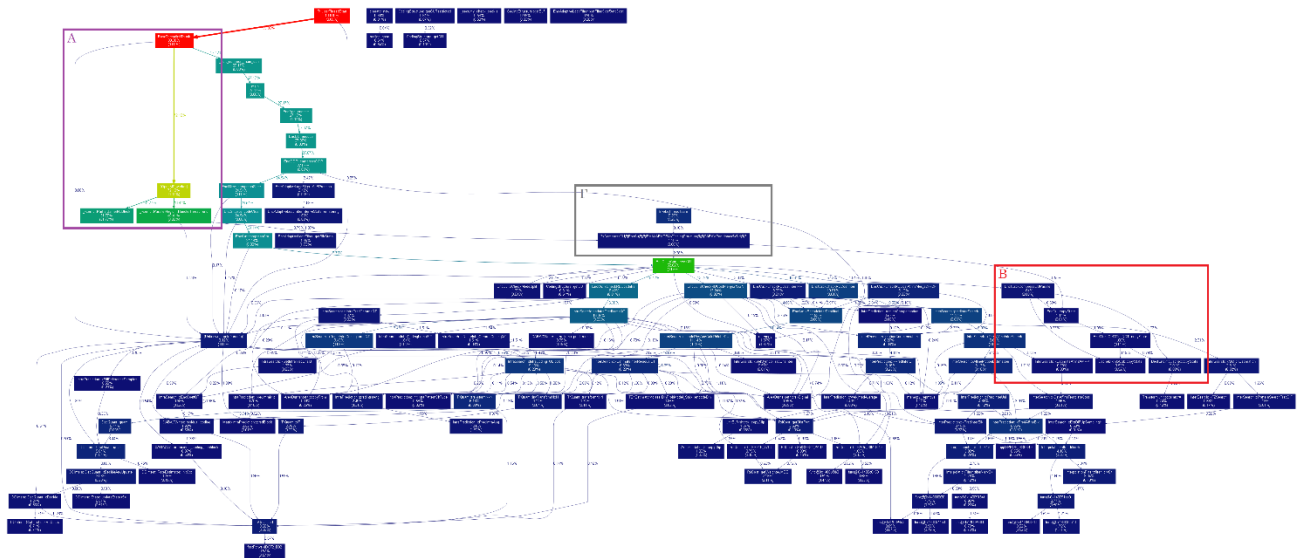
εκτελεστούν όταν η παράλληλη εκτέλεση είναι απενεργοποιημένη και παραλείπονται όταν είναι ενεργοποιημένη. Ένα τέτοιο παράδειγμα είναι στο TypeDef.h που το κομμάτι ορισμού του αριθμού των παράλληλων εργασιών βρίσκεται κάτω από ένα #if ENABLE_SPLIT_PARALLELISM με αποτέλεσμα να μη εκτελείται όταν η παράλληλη εκτέλεση είναι απενεργοποιημένη. Αυτό συμβαίνει γιατί όταν τρέχει σειριακά ο κώδικας δεν δημιουργούνται παράλληλες εργασίες γιατί δεν υπάρχουν νήματα για να τις εκτελέσουν. Άλλο ένα παράδειγμα αλλαγών της ροής του κώδικα μεταξύ σειριακής και παράλληλης εκτέλεσης είναι στην συνάρτηση EncLib::create, μέσα στην οποία αν υπάρχει παραλληλισμός μπαίνει σε μια δομή επανάληψης for και εκτελεί ορισμένες φορές την συνάρτηση createEnc, ενώ σειριακά παραλείπει αυτό το κομμάτι και εκτελεί το κομμάτι της σειριακής εκτέλεσης που είναι πάλι η συνάρτηση createEnc αλλά για μόνο μια φορά. Στο ίδιο παράδειγμα ο αριθμός των επαναλήψεων της for εξαρτάται από την τιμή που έχουμε δώσει στην PARL_SPLIT_MAX_NUM_JOBS. Οι επαναλήψεις είναι ο αριθμός των παράλληλων εργασιών που έχουμε ορίσει αυξημένος κατά 1, που είναι για το κύριο νήμα της εκτέλεσης. Τα νήματα στην παράλληλη εκτέλεση δεν παραμένουν μόνιμα ενεργά. Σε κάθε αλλαγή καρτέ στην κωδικοποίηση όλα τα νήματα εκτός από το κύριο νήμα κλείνουν και ξανά ανοίγουν αφού περαστεί σε αρχείο το κωδικοποιημένο αποτέλεσμα του καρτέ.

Προκειμένου να κατανοηθεί καλύτερα η ροή του κώδικα με και χωρίς παραλληλοποίηση χρησιμοποιήσαμε το εργαλείο Intel® VTune™ Amplifier¹ [29]. Κάνοντας εκτέλεση του αλγορίθμου, το Intel® VTune™ Amplifier συλλέγει στοιχεία που χρησιμεύουν στην ανάλυση της ροής του κώδικα και στο τέλος της εκτέλεσης εμφανίζει οργανωμένα τα αποτελέσματα. Στην συνέχεια για να μπορέσουμε να φτιάξουμε μια γραφική αναπαράσταση του αλγορίθμου με τη μορφή γράφου, χρησιμοποιήσαμε τα προγράμματα gprof2dot™ [31] για να μετατρέψει το αποτέλεσμα της ανάλυσης του Intel® VTune™ Amplifier σε αρχείο της μορφής dot και το Graphviz™ [30] για να πάρει αυτό το αρχείο dot και να δημιουργήσει την γραφική αναπαράσταση.



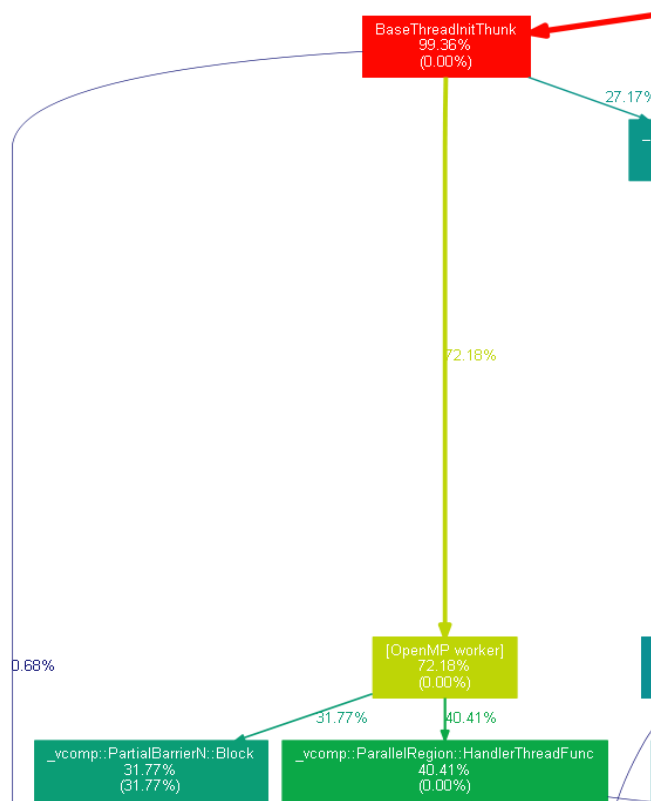
Εικόνα 15 -Γράφος του Intel® VTune™ Amplifier σειριακής εκτέλεσης

¹ VTune is a trademark of Intel Corporation or its subsidiaries



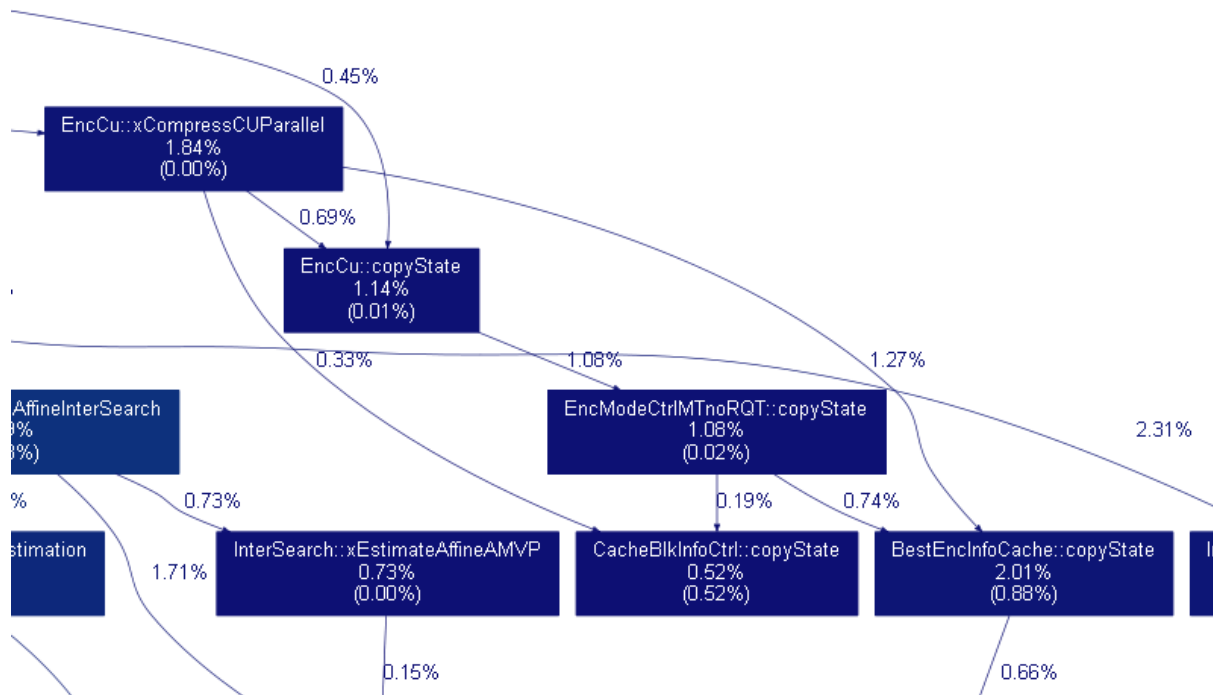
Εικόνα 16 -Γράφος του Intel® VTune™ Amplifier παράλληλης εκτέλεσης

Στους δύο αυτούς γράφους των Εικόνων 15 και 16 φαίνονται οι κυριότερες συναρτήσεις που συναντώνται στην ροή του κώδικα μα όχι όλες. Έτσι μπορεί να φαίνεται ότι κάποιες συναρτήσεις που υπάρχουν στον ένα γράφο δεν υπάρχουν στον άλλο, αλλά για να το επιβεβαιώσουμε πρέπει να ελέγξουμε τα αποτελέσματα στο Intel® VTune™ Amplifier¹. Συγκρίνοντας τους γράφους των δύο εκτελέσεων αλλά και τα αποτελέσματα του Intel® VTune™ Amplifier παρατηρήθηκαν τρεις κύριες διαφορές στην ροή του κώδικα (Εικόνα 16).

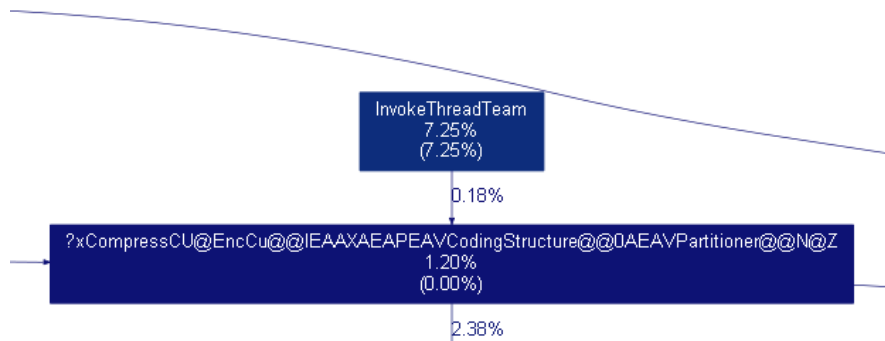


Εικόνα 17 -Περιοχή A του γράφου της Εικόνας 16

¹ VTune is a trademark of Intel Corporation or its subsidiaries.



Εικόνα 18 -Περιοχή Β του γράφου της Εικόνας 16



Εικόνα 19 -Περιοχή Γ του γράφου της Εικόνας 16

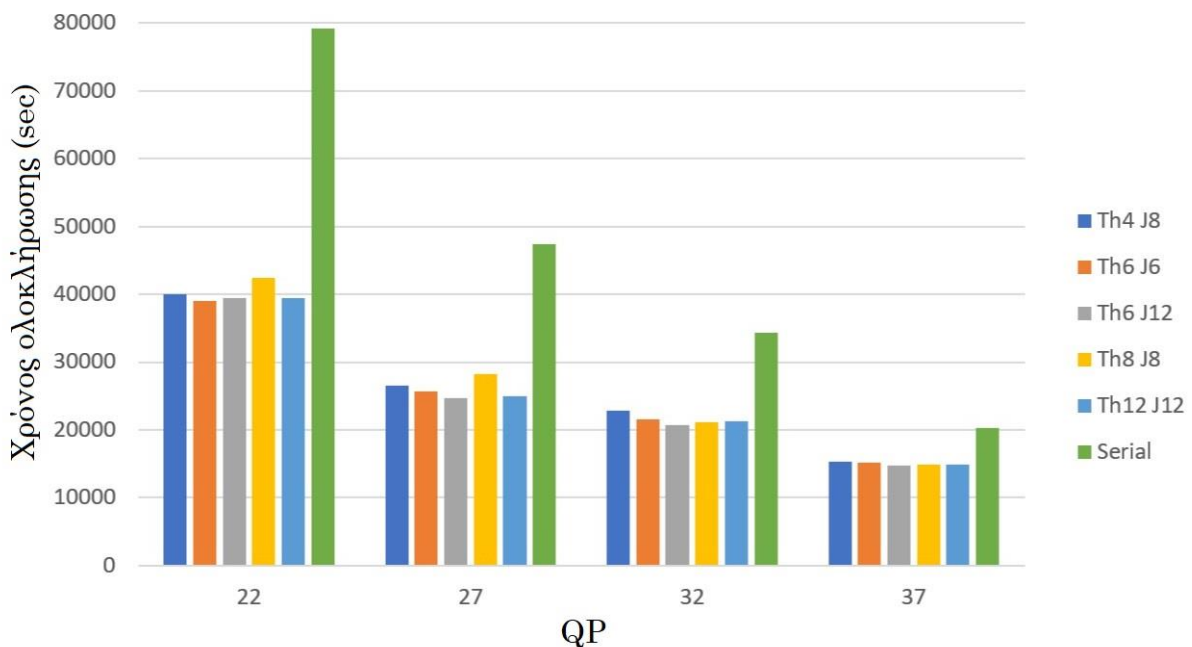
Στις Εικόνες 17, 18 και 19 βλέπουμε τις συναρτήσεις στην ροή του αλγορίθμου που υπάρχουν μόνο στην παράλληλη εκτέλεση του κώδικα. Οι συναρτήσεις της Εικόνας 17 χρησιμοποιούνται στον χειρισμό των νημάτων που έχουν παραχθεί. Διαχειρίζονται τις παύσεις των νημάτων στα σημεία που χρειάζονται καθώς και τα κρίσιμα σημεία του κώδικα στα οποία πρέπει να μπαίνει μία συνάρτηση την φορά, ενώ οι υπόλοιπες μπλοκάρονται μέχρι να έρθει η σειρά τους και να τους επιτραπεί η είσοδος. Από την Εικόνα 17 φαίνεται πώς μεγάλο μέρος του χρόνου της κωδικοποίησης καταναλώνεται αποκλειστικά σε barriers, προκειμένου να γίνεται σωστός συντονισμός των νημάτων. Αυτό ίσως εξηγεί σε μεγάλο βαθμό ορισμένα από τα αποτελέσματα των πειραμάτων που παρατηρούμε στο επόμενο κεφάλαιο. Οι συναρτήσεις της περιοχής Γ του γράφου παράλληλης εκτέλεσης (Εικόνα 19) διαχειρίζονται τις κλήσεις των νημάτων στα σημεία που χρειάζονται. Στην Εικόνα 18 οι συναρτήσεις copyState χρησιμοποιούνται για τη σωστή διαχείριση των δεδομένων που υφίστανται επεξεργασία στον κωδικοποιητή που εκτελείται παραλληλοποιημένα.

ΚΕΦΑΛΑΙΟ 5 Πειραματικά Αποτελέσματα

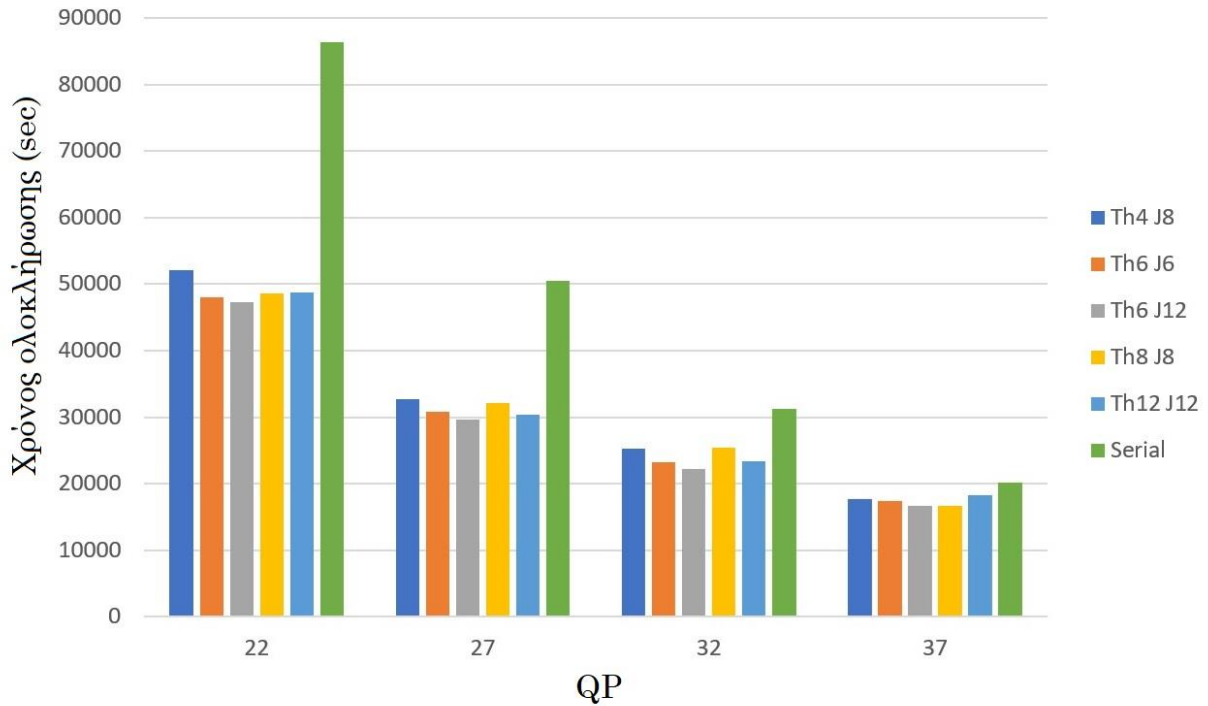
Πραγματοποιήσαμε πειράματα σε έναν Linux Server ο οποίος είναι εξοπλισμένος με δύο επεξεργαστές 12-core Intel Xeon E5-2650 @ 2.20GHz, χρησιμοποιώντας το λογισμικό αναφοράς VTM 6.0 [27]. Στα πειράματα χρησιμοποιήσαμε 2 βίντεο, το Kimono με ανάλυση 1920 x 1080 και 24 καρέ ανά δευτερόλεπτο και το Traffic με ανάλυση 2560 x 1600 και 30 καρέ ανά δευτερόλεπτο. Προκειμένου να εξοικονομήσουμε χρόνο στα πειράματα χρησιμοποιήσαμε για κωδικοποίηση 100 καρέ από κάθε βίντεο αντί να τα κωδικοποιήσουμε ολόκληρα. Στο Kimono χρησιμοποιήσαμε τα καρέ από 50 μέχρι και 149 προκειμένου να συμπεριληφθεί στην κωδικοποίηση η αλλαγή σκηνής που συμβαίνει στο καρέ 140. Στο Traffic χρησιμοποιήσαμε τα πρώτα 100 καρέ. Σε όλες τις περιπτώσεις χρησιμοποιήθηκαν οι προκαθορισμένες ρυθμίσεις του encoder του VVC, όπως αυτές ορίζονται στα [3] και [6], με την μόνη διαφορά ότι αλλάζαμε τις τιμές των `PARL_SPLIT_MAX_NUM_JOBS`, `NumSplitThreads` και `QP` για να συγκρίνουμε τις διαφορές απόδοσης σε κάθε ρύθμιση.

Μετρήσαμε την απόδοση του αλγόριθμου όσον αφορά την ταχύτητα ολοκλήρωσής του ανάλογα με το αν η εκτέλεσή του ήταν σειριακή ή παράλληλη. Στην παράλληλη εκτέλεση οι περιπτώσεις που πήραμε στα πειράματα είναι 5. Να έχουμε 4 threads με 8 jobs (σε τόσα κομμάτια χωρίζονται οι εργασίες που πρέπει να γίνουν στον αλγόριθμο και μοιράζονται ομοιόμορφα στα threads ανάλογα με τον αριθμό τους), 6 threads με 6 jobs, 6 threads με 12 jobs, 8 threads με 8 jobs και 12 threads με 12 jobs.

Στους Πίνακες 1 και 2 συνοψίζονται τα πειραματικά αποτελέσματα για τα βίντεο Kimono και Traffic αντίστοιχα, ενώ στις Εικόνες 20 και 21 παρουσιάζεται η γραφική τους απεικόνιση.



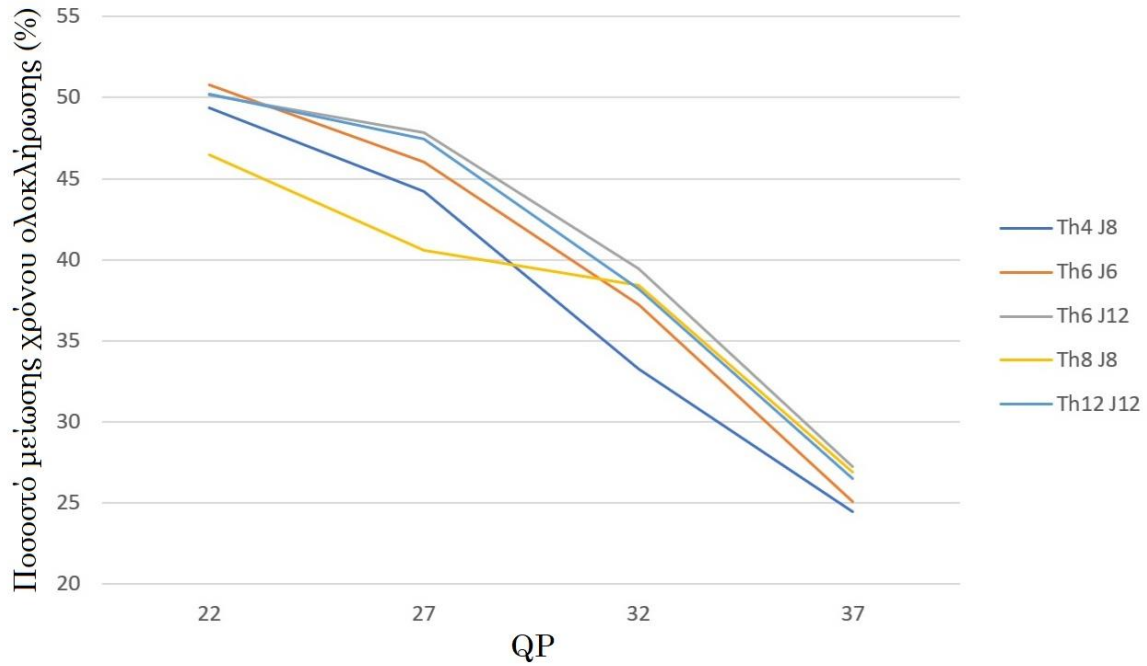
Εικόνα 20 -Χρόνος εκτέλεσης για το Kimono



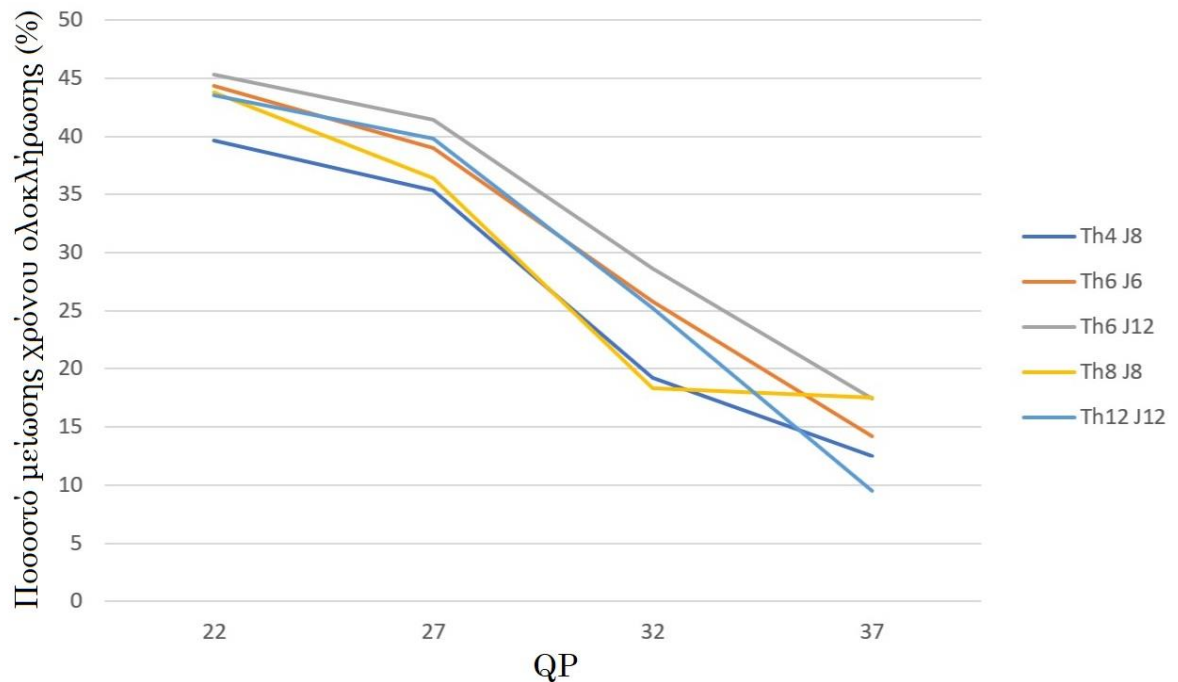
Εικόνα 21 - Χρόνος εκτέλεσης για το Traffic

Αυτό που μπορεί να παρατηρήσει κανείς από τα πειραματικά αποτελέσματα που παρουσιάζονται στους Πίνακες 1 και 2, καθώς και στις Εικόνες 20 και 21 είναι ότι οι περιπτώσεις της παράλληλης εκτέλεσης φαίνονται να κυμαίνονται περίπου στα ίδια επίπεδα ταχύτητας, ανεξαρτήτως του πλήθους των νημάτων που χρησιμοποιούνται. Κάτι τέτοιο μπορεί να σημαίνει ότι το μέγιστο επίπεδο παραλληλοποίησης που επιτυγχάνεται με τη μέθοδο του Split Parallelism είναι πολύ μικρό. Δεν μπορούμε βέβαια να αποκλείσουμε και την περίπτωση μη αποδοτικής υλοποίησης της παραλληλοποίησης στο λογισμικό αναφοράς.

Στην συνέχεια θα δούμε το ποσοστό μείωσης του χρόνου ολοκλήρωσης των περιπτώσεων παράλληλης εκτέλεσης σε σχέση με τον χρόνο ολοκλήρωσης με σειριακή εκτέλεση προκειμένου να γίνουν πιο εμφανής οι καλύτερες επιδόσεις.



Εικόνα 22 -Ποσοστό μείωσης χρόνου συγκριτικά με το Serial στο Kimono



Εικόνα 23 -Ποσοστό μείωσης χρόνου συγκριτικά με το Serial στο Traffic

Ένας σημαντικός παράγοντας ο οποίος θα πρέπει να αξιολογηθεί προκειμένου να έχουμε μια σαφέστερη άποψη κατά πόσο η παραλληλοποίηση με το Split Parallelism θα μπορούσε να βοηθήσει στην επιτάχυνση της διαδικασίας κωδικοποίησης, είναι η απόδοση της συμπίεσης. Όπως είναι γνωστό η απόδοση της συμπίεσης αποδίδεται από τις καμπύλες Ρυθμού – Παραμόρφωσης (R-D Curves). Στη συνέχεια λοιπόν θα παρουσιάσουμε τα

πειραματικά δεδομένα που αφορούν την απόδοση της συμπίεσης με και χωρίς παραλληλοποίηση. Στους Πίνακες 1 και 2 εμφανίζονται τα συγκεντρωτικά αυτά αποτελέσματα για τα βίντεο Kimono και Traffic αντίστοιχα.

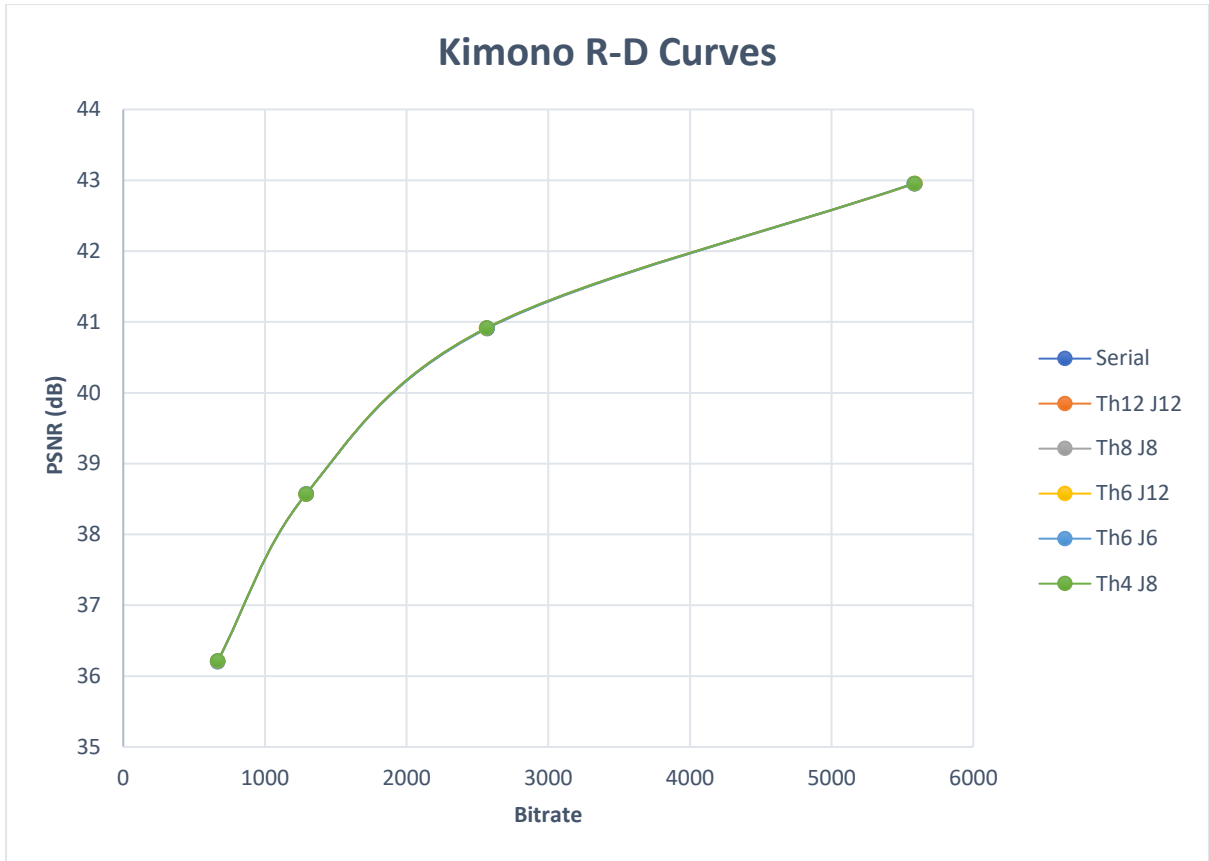
Πίνακας 1 Απόδοση Συμπίεσης για το Kimono

KIMONO						
	QP	Bitrate	YUV-PSNR	Time (s)	ΔBitrate (%)	ΔPSNR
Serial	22	5588.6016	42.9543	79,146		
	27	2568.3379	40.9069	47,434		
	32	1291.5226	38.5705	34,324		
	37	665.7869	36.203	20,356		
Th4J8	22	5584.3642	42.9551	40,082	0.076	-0.001
	27	2564.9453	40.915	26,466	0.132	-0.008
	32	1290.5088	38.5719	22,898	0.078	-0.001
	37	665.5526	36.2149	15,377	0,035	-0.012
Th6J6	22	5581.4707	42.9553	38,955	0.128	-0.001
	27	2566.3066	40.9137	25,615	0.079	-0.007
	32	1291.1251	38.5732	21,535	0.031	-0.003
	37	665.4912	36.2115	15,245	0.044	-0.008
Th6J12	22	5587.4688	42.9562	39,436	0.020	-0.002
	27	2564.7053	40.9155	24,731	0.141	-0.009
	32	1290.0384	38.5716	20,780	0.115	-0.001
	37	665.4912	36.2115	14,811	0.044	-0.008
Th8J8	22	5583.7344	42.9556	42,362	0.087	-0.001
	27	2565.817	40.9152	28,189	0.098	-0.008
	32	1291.2384	38.5715	21,134	0.022	-0.001
	37	665.4182	36.2105	14,881	0.055	-0.008
Th12J12	22	5584.3104	42.9557	39,381	0.077	-0.001
	27	2563.4995	40.9154	24,924	0.188	-0.008
	32	1290.8467	38.5719	21,209	0.052	-0.001
	37	665.4912	36.2115	14,958	0.044	-0.008

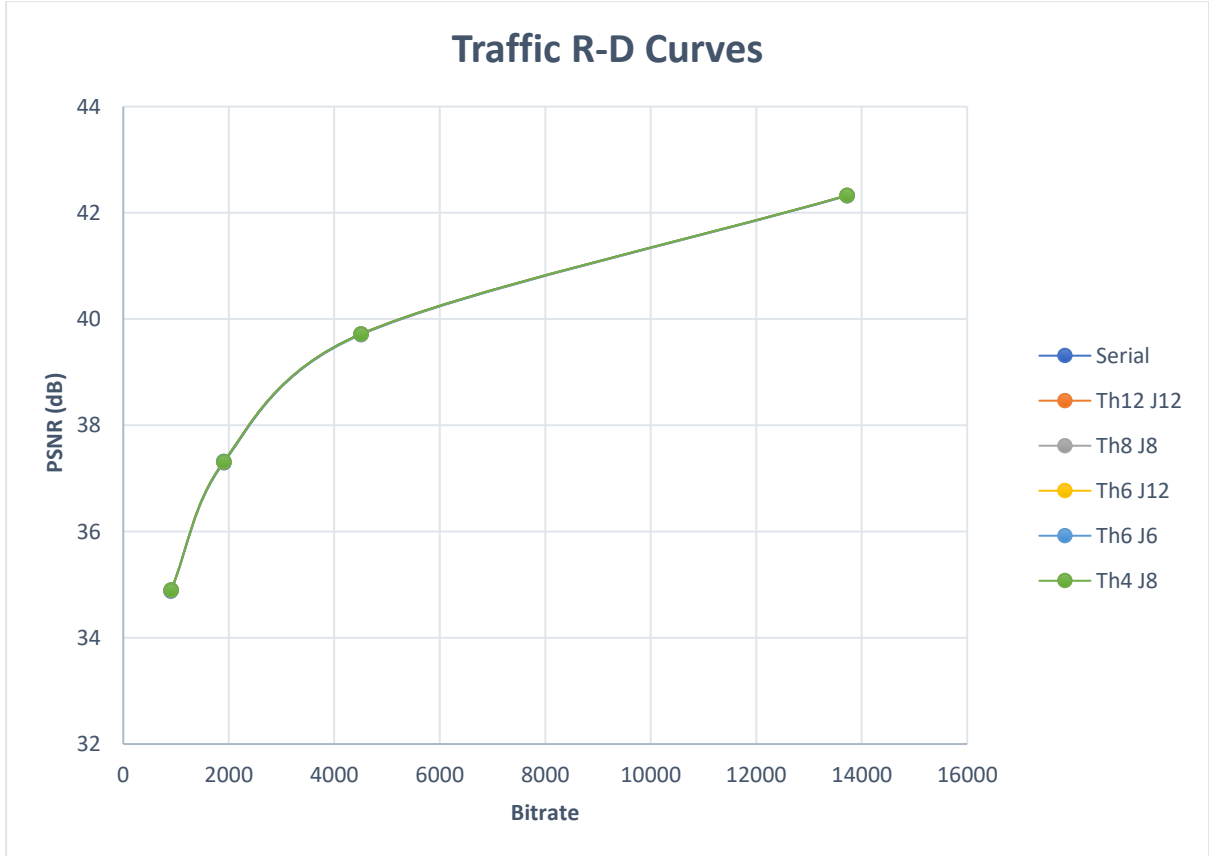
Πίνακας 2 Απόδοση Συμπίεσης για το Traffic

TRAFFIC						
	QP	Bitrate	YUV-PSNR	Time (s)	ΔBitrate (%)	ΔPSNR
Serial	22	13723.973	42.3192	86,412		
	27	4507.6008	39.7064	50,556		
	32	1908.0024	37.299	31,224		
	37	904.7544	34.8799	20,204		
Th4J8	22	13720.831	42.3273	52,145	0.023	-0.008
	27	4507.6872	39.7202	32,692	-0.002	-0.014
	32	1907.0616	37.3153	25,223	0.049	-0.016
	37	906.3072	34.9001	17,685	-0.172	-0.020
Th6J6	22	13724.34	42.3275	48,096	-0.003	-0.008
	27	4507.3224	39.7203	30,849	0.006	-0.014
	32	1906.1976	37.3188	23,176	0.095	-0.020
	37	907.5936	34.8983	17,339	-0.314	-0.018
Th6J12	22	13725.838	42.3286	47,279	-0.014	-0.009
	27	4505.7552	39.7193	29,608	0.041	-0.013
	32	1904.3304	37.3171	22,287	0.192	-0.018
	37	907.5936	34.8983	16,680	-0.314	-0.018
Th8J8	22	13722.238	42.327	48,569	0.013	-0.008
	27	4507.5504	39.7195	32,157	0.001	-0.013
	32	1905.6984	37.3204	25,494	0.121	-0.021
	37	907.632	34.8998	16,670	-0.318	-0.020
Th12J12	22	13723.313	42.3272	48,807	0.005	-0.008
	27	4505.9256	39.7174	30,420	0.037	-0.011
	32	1904.3304	37.3171	23,356	0.192	-0.018
	37	906.804	34.8997	18,289	-0.227	-0.020

Αυτό που μπορεί να παρατηρήσει κανείς από τα παραπάνω στοιχεία είναι ότι η απόδοση της παράλληλης συμπίεσης είναι σχεδόν ταυτόσημη με αυτήν της σειριακής. Οι απώλειες σε ποιότητα (ΔPSNR) όσο και η αύξηση του ρυθμού από bits (ΔBitrate) είναι αμελητέες. Αυτό έχει ως αποτέλεσμα οι καμπύλες Ρυθμού-Παραμόρφωσης της σειριακής κωδικοποίησης και των παράλληλων κωδικοποιήσεων να συμπίπτουν όπως φαίνεται στις Εικόνες 24 και 25.



Εικόνα 24 R-D Curves for Kimono sequence

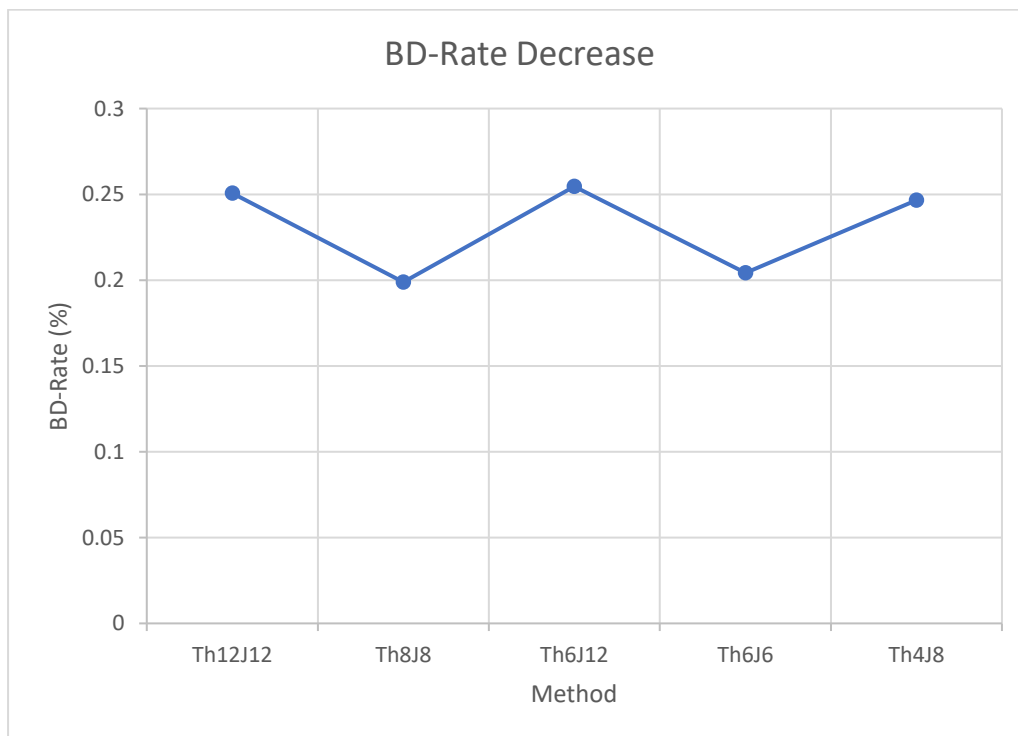


Εικόνα 25 R-D Curves for Traffic sequence

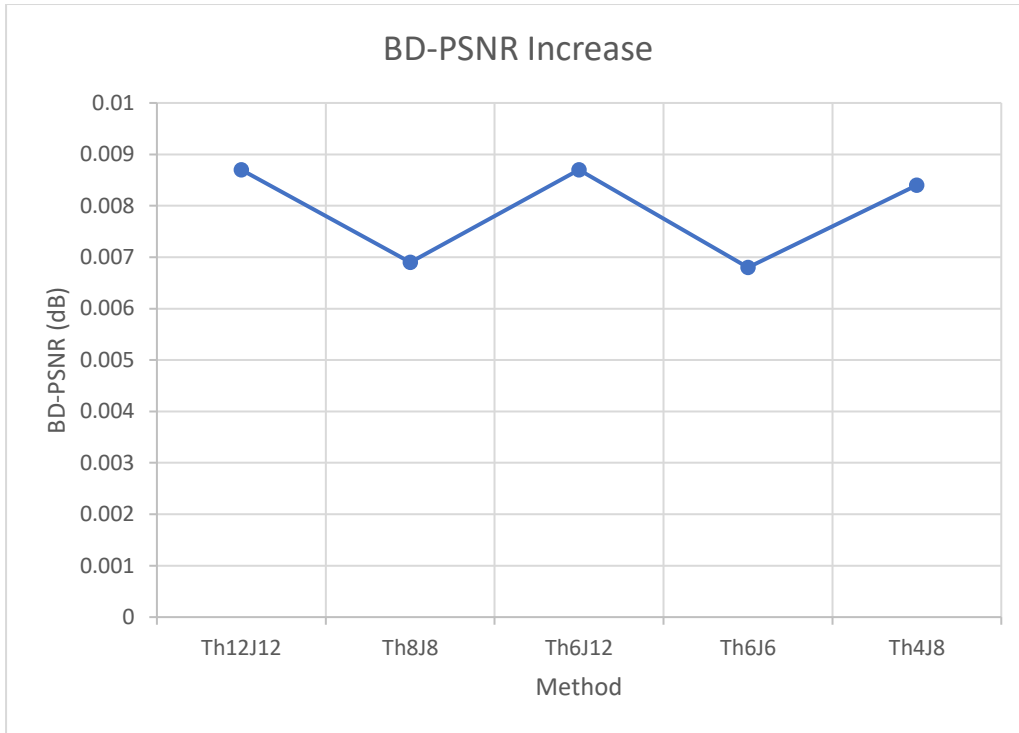
Καθώς η μονάδα μέτρησης της ποιότητας είναι σε λογαριθμική κλίμακα, ενδέχεται οι παραπάνω καμπύλες να συμπίπτουν, αλλά πάραυτα να υπάρχουν μη αμελητέες διαφορές όταν εξετάζουμε τα δεδομένα μας σε μη λογαριθμική κλίμακα. Για να βεβαιωθούμε ότι δεν συμβαίνει κάτι τέτοιο προχωρήσαμε στον υπολογισμό των διαφορών Bjontegaard [28], και τα αποτελέσματα παρουσιάζονται παρακάτω. Στον Πίνακα 3 εμφανίζονται η ποσοστιαία αύξηση του bitrate και η αντίστοιχη μείωση του PSNR για το Kimono και το Traffic, ενώ στις Εικόνες 26 και 27 παρουσιάζονται τα παραπάνω δεδομένα για το Kimono. Τα αντίστοιχα για το Traffic παρουσιάζονται στις Εικόνες 28 και 29.

Πίνακας 3 Διαφορές Bjontegaard

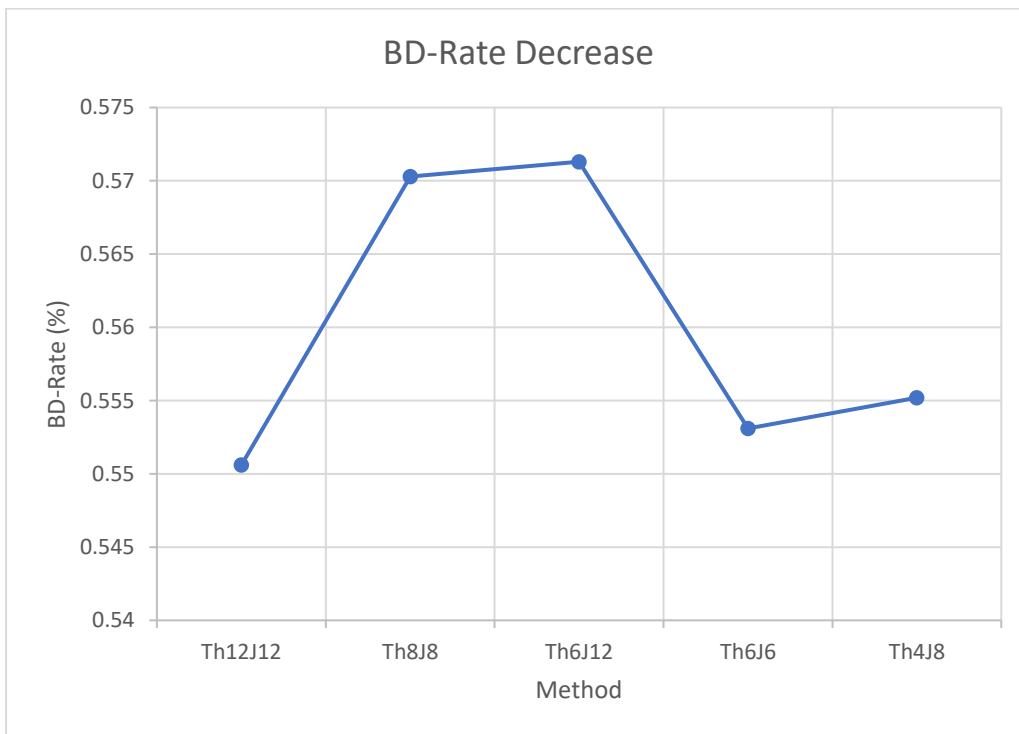
<Threads>_Jobs	KIMONO		TRAFFIC	
	BD-Rate (%)	BD-PSNR (dB)	BD-Rate (%)	BD-PSNR (dB)
Th12J12	0,2508	0,0087	0,5506	0,0143
Th8J8	0,1989	0,0069	0,5703	0,0147
Th6J12	0,2546	0,0087	0,5713	0,0148
Th6J6	0,2042	0,0068	0,5531	0,0145
Th4J8	0,2466	0,0084	0,5552	0,0147



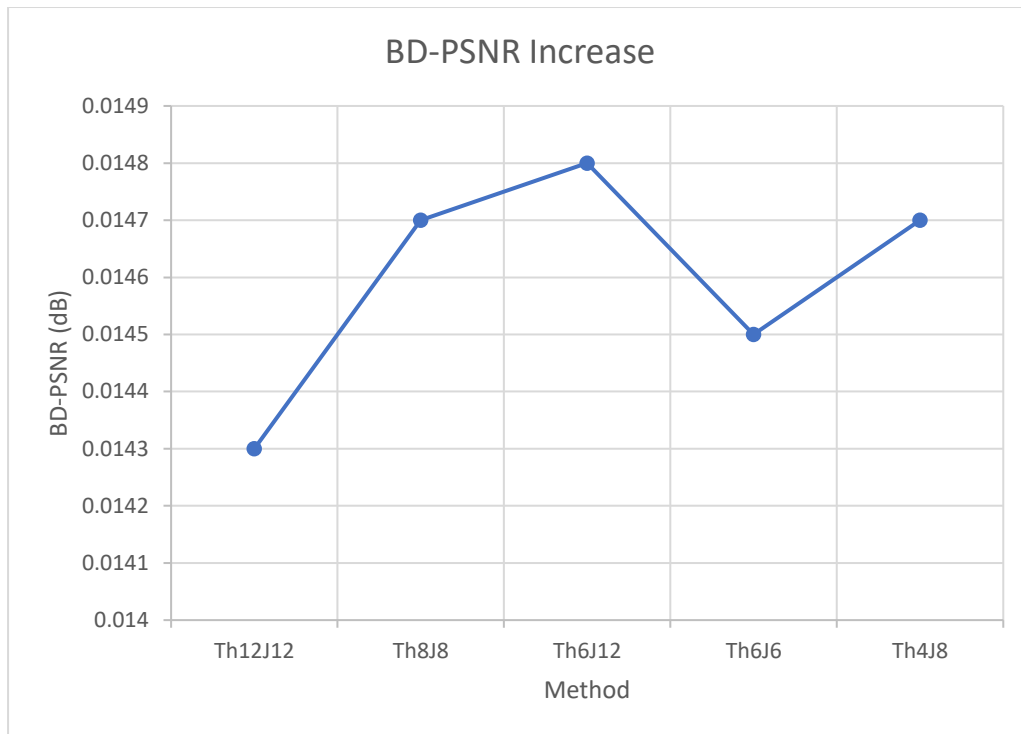
Εικόνα 26 BD-Rate for Kimono sequence



Εικόνα 27 BD-PSNR for Kimono sequence



Εικόνα 28 BD-Rate for Traffic sequence



Εικόνα 29 BD-PSNR for Traffic sequence

Από τα παραπάνω αποτελέσματα βλέπουμε ότι η παραλληλοποίηση σε επίπεδο CTU, όπως αυτή εκτιμήθηκε με τη μέθοδο Split Parallelism δεν παρουσιάζει καμία επίπτωση στην απόδοση συμπίεσης, ενώ μπορεί να επιτύχει -έστω και περιορισμένη- επιτάχυνση στην όλη διαδικασία. Στο Κεφάλαιο 6 που ακολουθεί θα υπάρξει μια πιο λεπτομερής περιγραφή των συμπερασμάτων που προκύπτουν από την παρούσα εργασία.

ΚΕΦΑΛΑΙΟ 6 Συμπεράσματα

Το VVC αποτελεί το αποδοτικότερο και πιο εξελιγμένο πρότυπο συμπίεσης βίντεο παρόλο που η ανάπτυξή του δεν έχει ολοκληρωθεί. Έχει κάνει μεγάλα βήματα προόδου σε σχέση με τους προκατόχους του και έχει εισάγει νέες τεχνολογίες στην κωδικοποίηση βίντεο. Πρόκειται για μια στέρεη βάση για την ανάπτυξη της επόμενης γενιάς εφαρμογών και πολυμέσων. Αυτό όμως έρχεται με το κόστος της μεγάλης αύξησης της πολυπλοκότητάς του. Για να μπορέσει το νέο αυτό πρότυπο να εδραιωθεί και να χρησιμοποιηθεί ευρέως είναι επιτακτική η ανάγκη να βρεθούν τρόποι μείωσης της πολυπλοκότητας αυτής χωρίς να υπάρχει σημαντικό κόστος στην απόδοση του νέου προτύπου. Η παραλληλοποίηση, σε πολλά επίπεδα, μπορεί να είναι μια τέτοια λύση.

Στην εργασία αυτή μελετήσαμε την απόδοση της παραλληλοποίησης σε επίπεδο CTU, όπως αυτή έχει υλοποιηθεί από το λογισμικό αναφοράς VTM. Στα πειράματα που πραγματοποιήσαμε στον server του πανεπιστημίου είδαμε πως υπάρχει σημαντική μείωση στον χρόνο που χρειάζεται για να ολοκληρωθεί η κωδικοποίηση ενός βίντεο όταν το εκτελούμε με ενεργοποιημένη την παραλληλοποίηση σε σχέση με όταν το τρέχουμε σειριακά. Οι διαφορές στην απόδοση όμως όταν ανεβάζουμε τον αριθμό των νημάτων που χρησιμοποιούνται από τον αλγόριθμο και τον αριθμό των παράλληλων εργασιών είναι ελάχιστες. Τα πειράματα έδειξαν ότι η εκτέλεση με τα 6 νήματα και τις 12 παράλληλες εργασίες ήταν η ταχύτερη με μικρές διαφορές και στα δύο βίντεο που χρησιμοποιήσαμε. Τα αποτελέσματα αυτά μπορούν να ερμηνευτούν με δυο τρόπους:

- a) *Η παραλληλοποίηση σε επίπεδο CTU είναι περιορισμένη.* Αυτό εν μέρη είναι ορθό, καθώς το βάθος των CUs είναι περιορισμένο. Ταυτόχρονα όμως το πλήθος τους είναι αρκετά μεγάλο. Θα περίμενε λοιπόν κανείς η μείωση του χρόνου εκτέλεσης να αυξάνεται καθώς ο αριθμός των νημάτων αυξάνει.
- b) *Η υλοποίηση της παραλληλοποίησης στο λογισμικό αναφοράς δεν είναι αποδοτική.* Κάτι τέτοιο είναι πολύ πιθανό, γεγονός που παρέχει τη δυνατότητα βελτίωσής του.

Το αποτέλεσμα όμως στο οποίο αξίζει να εστιάσουμε είναι το γεγονός ότι η παραλληλοποίηση σε επίπεδο CTU δεν επιφέρει καμία επίπτωση στην απόδοση του κωδικοποιητή. Αυτό σημαίνει ότι εάν καταφέρουμε να αυξήσουμε το ποσοστό μείωσης του χρόνου βελτιώνοντας το λογισμικό έτσι ώστε να επιτυγχάνει τον μέγιστο ρυθμό παραλληλοποίησης, τα συνολικά αποτελέσματα θα είναι πραγματικά σημαντικά.

Ακόμα και αν ο μέγιστος ρυθμός παραλληλοποίησης δεν είναι αρκετά μεγάλος έτσι ώστε να αξιοποιεί αποδοτικά τους πόρους των σύγχρονων υπολογιστικών συστημάτων, η παραλληλοποίηση σε επίπεδο CTU μπορεί να χρησιμοποιηθεί ταυτόχρονα με άλλες μεθόδους παραλληλοποίησης (π.χ. tile parallelization), επιτυγχάνοντας με αυτόν τον τρόπο σημαντική μείωση στον χρόνο κωδικοποίησης, χωρίς περαιτέρω μείωση της απόδοσης του κωδικοποιητή.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] B. Bross, J. Chen, S. Liu, and Y.-K. Wang “Versatile Video Coding (Draft 7),” document JVET-P2001, 16th JVET meeting: Geneva, CH, 1–11 Oct. 2019.
- [2] J. Chen, Y. Ye, and S.-H. Kim “Algorithm description for Versatile Video Coding and Test Model 8 (VTM 8),” document JVET-Q2002, 17th JVET meeting: Brussels, 7–17 Jan. 2020
- [3] F. Bossen, J. Boyce, X. Li, and V. Seregin, K. Sühring, “JVET common test conditions and software reference configurations for SDR video,” document JVET-N1010, 14th JVET meeting: Geneva, MA, 19–27 Mar. 2019.
- [4] High Efficiency Video Coding (HEVC), Rec. ITU-T H.265 and ISO/IEC 23008-2, Jan. 2013 (and later editions).
- [5] 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.
- [6] F. Bossen, “Common test conditions and software reference configurations,” in Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 5th meeting, Jan. 2011, 2011.
- [7] 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.
- [8] M. Koziri, P. K. Papadopoulos, N. Tziritas, N. Giachoudis, T. Loukopoulos, S. U Khan, and G. I. Stamoulis, “Heuristics for tile parallelism in hevc,” in *Signal Processing Conference (EUSIPCO)*, 2017 25th European. IEEE, 2017, pp. 1514–1518.
- [9] M. Shafique, M. U. K. Khan, and J. Henkel, “Power efficient and workload balanced tiling for parallelized high efficiency video coding,” in *Image Processing (ICIP)*, 2014 IEEE International Conference on. IEEE, 2014, pp. 1253–1257.
- [10] 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, pp. 26 – 26 – 10, 2017.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.

- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] x265 HEVC encoder, available at: <http://x265.org>.
- [27] VTM reference software, available at: <http://jvet.hhi.fraunhofer.de>
- [28] G. Bjontegaard, "Calculation of average PSNR differences between RD-Curves," Doc. VCEG-M33, Apr. 2001
- [29] Intel® VTune™ Amplifier, available at: <https://software.intel.com/en-us/vtune>
- [30] Graphviz™, available at: <https://www.graphviz.org/>
- [31] gprof2dot™, available at: <https://github.com/jrfonseca/gprof2dot>