

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



**Τεχνικές βελτιστοποίησης χρόνου συμπίεσης βίντεο στο
πρότυπο HEVC**

Δημήτριος Σκουμπουρδής

Πτυχιακή Εργασία

Επιβλέπων Καθηγητής: Γεώργιος Σταμούλης

Συνεπιβλέπουσα Καθηγήτρια: Μαρία Κοζύρη

Λαμία, 2017

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Τεχνικές βελτιστοποίησης χρόνου συμπίεσης βίντεο στο
πρότυπο HEVC**

Δημήτριος Σκουμπουρδής

Πτυχιακή Εργασία

Επιβλέπων Καθηγητής: Γεώργιος Σταμούλης

Συνεπιβλέπουσα Καθηγήτρια: Μαρία Κοζύρη

Λαμία, 2017

«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 αντιγραφική. 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

Ο-Η Δηλ.

(Υπογραφή)

(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 αυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

Τεχνικές βελτιστοποίησης χρόνου συμπίεσης βίντεο στο πρότυπο HEVC

Δημήτριος Σκουμπουρδής

Τριμελής Επιτροπή:

ΠΕΡΙΛΗΨΗ

Συμπίεση δεδομένων είναι η διαδικασία επεξεργασίας με σκοπό τη μείωση του όγκου, με ελάχιστη ή καμία απώλεια. Στην περίπτωση της συμπίεσης βίντεο, η επεξεργασία αυτή εφαρμόζεται πάνω σε ακολουθίες εικόνων καθιστώντας έτσι δυνατή τη μετάδοση και τον περαιτέρω χειρισμό τους. Για το σκοπό αυτό, έχουν αναπτυχθεί ανά τα χρόνια διάφορα πρότυπα συμπίεσης. Το HEVC έχει εμφανιστεί ως το καινούργιο πρότυπο συμπίεσης βίντεο, που υπόσχεται βελτιστοποίηση λόγου συμπίεσης (για την ίδια ποιότητα) μέχρι και 50% σε σύγκριση με το H.264/AVC. Για να πετύχει αυτήν την απόδοση, το HEVC απαιτεί αυξημένο υπολογιστικό φόρτο σε σχέση με τον προκάτοχό του. Για αυτόν το λόγο χρησιμοποιείται η παραλληλοποίηση, συνήθως σε coarse grained επίπεδο, π.χ. σε κάθε slice ή tile.

Η εργασία αυτή έχει θέμα τις διάφορες τεχνικές που μπορούν να επιστρατευθούν για την βελτιστοποίηση χρόνου συμπίεσης βίντεο, συγκεκριμένα στο πρότυπο HEVC. Η μελέτη περιλαμβάνει τις περιπτώσεις χρήσης AVX2 εντολών. Παρουσιάζουμε τις υλοποιήσεις των τεχνικών αυτών αναλύοντας τα υπέρ και τα κατά και τις διαφορές ανάμεσά τους. Στρέφουμε την προσοχή μας προς την περαιτέρω επιτάχυνση της διαδικασίας κωδικοποίησης του HEVC με την συγχώνευση της coarse grained και της fine grained παραλληλοποίησης με τη μορφή εντολών AVX2 που εφαρμόζουν SIMD παραλληλοποίηση στον υπολογισμό του SAD (Άθροισμα απολύτων διαφορών) και του SSE (Άθροισμα τετραγώνων σφαλμάτων).

Τέλος, η εργασία περιλαμβάνει τα αποτελέσματα των πειραμάτων που διεξήχθησαν για την επιβεβαίωση της σωστής υλοποίησης και λειτουργίας των τεχνικών μαζί με τα συμπεράσματα αυτών. Η πειραματική εκτίμηση με τη χρήση κοινών ακολουθιών βίντεο απεικονίζει πως μπορεί να επιτευχθεί μια επιπλέον μείωση (στο χρόνο κωδικοποίησης) περίπου της τάξης του 11% κατά μέσο όρο, σε σύγκριση με standalone coarse grained παραλληλοποίηση, οδηγώντας σε πολλές περιπτώσεις σε superlinear επιτάχυνση. Μέρος της πτυχιακής εργασίας δημοσιεύτηκε στο 21st Panhellenic Conference on Informatics (PCI 2017) κάτω από τον τίτλο: “On Improving the Speedup of Slice and Tile Level Parallelism in HEVC Using AVX2”.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΚΟΝΕΣ	9
ΠΙΝΑΚΕΣ.....	10
1 ^ο ΚΕΦΑΛΑΙΟ.....	11
ΣΥΜΠΙΕΣΗ ΒΙΝΤΕΟ ΚΑΙ Η ΕΞΕΛΙΞΗ ΣΤΟ ΧΡΟΝΟ	11
1.1. ΕΙΣΑΓΩΓΗ.....	11
1.2. ΕΞΕΛΙΞΗ ΣΤΟ ΧΡΟΝΟ.....	12
1.3. ΣΧΕΤΙΚΗ ΔΟΥΛΕΙΑ	13
1.4. ΣΥΓΚΡΙΣΗ ΚΩΔΙΚΟΠΟΙΗΤΩΝ	15
1.4.1. VP8	15
1.4.2. VP9	16
1.4.3. H.264/AVC	19
1.5. ΣΤΟΧΟΙ ΚΑΙ ΟΡΓΑΝΩΣΗ ΠΤΥΧΙΑΚΗΣ.....	22
2 ^ο ΚΕΦΑΛΑΙΟ.....	23
ΕΠΙΣΚΟΠΗΣΗ HEVC	23
2.1. Εισαγωγή	23
A. Video Coding Layer	23
B. Αρχιτεκτονική σύνταξης υψηλού επιπέδου.....	26
2.2. Σύνταξη υψηλού επιπέδου.....	28
2.3. HEVC Video τεχνικές κωδικοποίησης.....	30
2.4. HEVC Profiles και Level Definitions.....	42
3 ^ο ΚΕΦΑΛΑΙΟ.....	43
ΤΕΧΝΙΚΕΣ ΠΑΡΑΛΛΗΛΟΠΟΙΗΣΗΣ	43
3.1. Slices.....	43
3.2. Tiles	43
3.3. SIMD	44
3.4. Υλοποίηση.....	45
4 ^ο ΚΕΦΑΛΑΙΟ.....	53
ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	53
4.1. Μεθοδολογία	53
4.2. Αποτελέσματα	55

4.3. Γραφικές παραστάσεις και πίνακες αποτελεσμάτων.....	56
5 ^ο ΚΕΦΑΛΑΙΟ.....	67
ΕΠΙΛΟΓΟΣ.....	67
ΒΙΒΛΙΟΓΡΑΦΙΑ	68

ΕΙΚΟΝΕΣ

Εικόνα 1. Παράδειγμα διαχωρισμού στην ακολουθία Kimono.....	14
Εικόνα 2. Συνάρτηση υπολογισμού SAD με είσοδο 32 pixels.....	47
Εικόνα 3. Συνάρτηση υπολογισμού SAD με είσοδο 12 pixels.....	49
Εικόνα 4. Συνάρτηση υπολογισμού SSE με είσοδο 32 pixels.....	51
Εικόνα 5. RD Γραφική παράσταση με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία PeopleOnStreet.	60
Εικόνα 6. RD Γραφική παράσταση) με τη χρήση Slices/Tiles με AVX2 για την ακολουθία PeopleOnStreet.	60
Εικόνα 7. RD Γραφική παράσταση με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία BasketballDrive.....	61
Εικόνα 8. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία BasketballDrive.....	61
Εικόνα 9. RD Γραφική παράσταση με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία BQTerrace.....	62
Εικόνα 10. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία BQTerrace.....	62
Εικόνα 11. RD Γραφική παράσταση) με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία Kimono.	63
Εικόνα 12. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία Kimono.	63
Εικόνα 13. RD Γραφική παράσταση) με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία ParkScene.....	64
Εικόνα 14. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία ParkScene.....	64
Εικόνα 15. RD Γραφική παράσταση με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία Traffic.	65
Εικόνα 16. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία Traffic.	65
Εικόνα 17. RD Γραφική παράσταση με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία Cactus.....	66
Εικόνα 18. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία Cactus.....	66

ΠΙΝΑΚΕΣ

Πίνακας 1. Παράδειγμα μέρους αρχείου ρυθμίσεων για την ακολουθία BasketballDrive.	54
Πίνακας 2. Χαρακτηριστικά Ακολουθιών Βίντεο	56
Πίνακας 3. Μείωση χρόνου συμπίεσης με τη χρήση AVX2	57
Πίνακας 4. Επιτάχυνση Slice και Tile παραλληλοποίησης σε συνδυασμό με AVX2 βελτιστοποιήσεις.....	58
Πίνακας 5. Διαφορά στο PSNR (Y) από τη χρήση AVX2.....	59

1^ο ΚΕΦΑΛΑΙΟ

ΣΥΜΠΙΕΣΗ ΒΙΝΤΕΟ ΚΑΙ Η ΕΞΕΛΙΞΗ ΣΤΟ ΧΡΟΝΟ

1.1. ΕΙΣΑΓΩΓΗ

Η συμπίεση βίντεο είναι ένα θέμα που απασχολεί την επιστημονική κοινότητα εδώ και περίπου έξι δεκαετίες [1], με ουσιαστική πρόοδο να υπάρχει τις τελευταίες τρεις. Η κύρια ανάγκη που οδήγησε στην ανάπτυξη προτύπων συμπίεσης ήταν η διευκόλυνση στην κατανάλωση και στο γενικό χειρισμό αρχείων βίντεο. Για παράδειγμα, ένα ασυμπίεστο βίντεο ανάλυσης 1920x1080 εικονοστοιχείων ανά καρέ, με 60 καρέ ανά δευτερόλεπτο και διάρκεια 10 λεπτών έχει μέγεθος περίπου 9GB¹. Αυτό δηλώνει από μόνο του πως δεν θα ήταν δυνατή η γρήγορη μετάδοση μέσω συμβατικών γραμμών και η μαζική αποθήκευση ενός τόσο διαδεδομένου τύπου αρχείου, ειδικά στην περίπτωση μεγαλύτερων αναλύσεων όπως 4K και 8K.

Με τις 4K κάμερες να κερδίζουν δημοτικότητα σε έξυπνες κινητές συσκευές και σε σημαντικούς πάροχους μέσων κοινωνικής δικτύωσης να προσφέρουν online υπηρεσίες streaming, η δικτυακή επιβάρυνση που επιβάλλεται από την μεταδώσει video μπορεί να γίνει απαγορευτικά μεγάλη. Αναγνωρίζοντας νωρίς τις νέες προκλήσεις που προκαλούνται από τις 4K και τις επερχόμενες 8K ακολουθίες video, η ομάδα MPEG κυκλοφόρησε το 2012 το High Efficiency Video Coding (HEVC) [2] πρότυπο για να αντικαταστήσει το ολοένα και παλαιότερο H.264/AVC [3]. Συγκριτικά με τον προκάτοχο του, το HEVC προσφέρει βελτιωμένη απόδοση (σε όρους συμπίεσης) κατά περίπου 50% [4], με αυξημένο υπολογιστικό φόρτο.

Για να αντιμετωπιστούν οι μεγάλες ανάγκες υπολογιστικής ισχύος, το HEVC προσφέρει coarse-grained παραλληλισμό σε 3 κύρια επίπεδα: Slices (που έχει πάρει από το H.264/AVC), Tiles και Wave-Front. Παρόλα αυτά ένα σημαντικό κομμάτι δουλειάς υπάρχει στην υλοποίηση και στην αξιολόγηση κάθε επιπέδου παραλληλισμού π.χ. [4], [5], [6], η πρόκληση μείωσης του χρόνου κωδικοποίησης του HEVC με τη χρήση παραλληλισμού είναι ακόμα ανοιχτή. Αυτό ισχύει και για τις δυο περιπτώσεις κωδικοποίησης, από raw αρχείο (π.χ. από κάμερα) και στο transcoding, [7], [54] όπου ήδη κωδικοποιημένες ακολουθίες χρησιμοποιούνται ως είσοδος. Συγκεκριμένα, η δεύτερη περίπτωση(transcoding) έχει αρκετές εφαρμογές που σχετίζονται με το cloud, π.χ. live casting [8], [9] και ζωντανή μετάδοση video σε μέσα κοινωνικής δικτύωσης [10].

Η κύρια αρχή των κωδικοποιητών βίντεο είναι η αφαίρεση της πλεονάζουσας πληροφορίας με ελάχιστη ή και καθόλου διαφορά στην τελική ποιότητα του αποκωδικοποιημένου προϊόντος, χρησιμοποιώντας όσο το δυνατό λιγότερη υπολογιστική ισχύ. Η πληροφορία μπορεί να χαρακτηριστεί ως πλεονάζουσα στο χώρο ή στο χρόνο. Στην περίπτωση της συσχέτισης στο

¹ Εάν θεωρήσουμε το πεδίο χρώματος ως YUV420 και έχουμε κατά συνέπεια 12 bits ανά εικονοστοιχείο.

χρόνο μιλάμε για τις ομοιότητες που μπορεί να παρουσιάζει μια ομάδα εικονοστοιχείων σε μια συγκεκριμένη θέση με μια άλλη από ένα προηγούμενο η επόμενο καρέ στην ίδια η σε κοντινή θέση και στην περίπτωση της συσχέτισης στο χώρο μιλάμε για ένα μοναδικό εικονοστοιχείο η μια ομάδα αυτών που η τιμή τους μπορεί να βρεθεί έπειτα από επεξεργασία των εικονοστοιχείων στις γειτονικές τους θέσεις. Μια τρίτη κατηγορία πλεονάζουσας πληροφορίας είναι η ψυχο-οπτική. Όπως αναφέρεται στην [11] το σύστημα ανθρώπινης όρασης αντιλαμβάνεται τον εξωτερικό κόσμο με έναν πολύπλοκο τρόπο. Η ανταπόκρισή του σε οπτικό ερέθισμα είναι μια μη γραμμική συνάρτηση της δύναμης κάποιων φυσικών χαρακτηριστικών του ερεθίσματος όπως η ένταση και το χρώμα. Στην ανθρώπινη όραση, η πληροφορία δε λαμβάνεται ίσα, δηλαδή κάποια πληροφορία μπορεί να είναι πιο σημαντική από άλλη. Αυτό σημαίνει πως εάν χρησιμοποιήσουμε λιγότερα δεδομένα για να αναπαραστήσουμε τις λιγότερο οπτικά σημαντικές πληροφορίες η αντίληψη δε θα αλλάξει. Με αυτόν τον τρόπο βλέπουμε πως κάποια πληροφορία είναι ψυχο-οπτικά πλεονάζουσα και η αφαίρεση αυτής οδηγεί σε συμπίεση των δεδομένων.

1.2. ΕΞΕΛΙΞΗ ΣΤΟ ΧΡΟΝΟ

[1] Βίντεο μπορούμε να συμπίεσουμε με πολλούς διαφορετικούς ιδιόκτητους ή τυποποιημένους αλγορίθμους. Οι πιο σημαντικές οικογένειες αλγορίθμων συμπίεσης δημοσιεύονται από αναγνωρισμένους οργανισμούς τυποποίησης, όπως τους International Organization for Standardization (ISO), International Telecommunication Union (ITU) ή Motion Picture Expert Group (MPEG). Η ιστορία της συμπίεσης βίντεο ξεκινάει από το 1950. Μια από τις πρώτες προσπάθειες που είχαν γίνει ήταν η σχεδίαση, κατασκευή και δοκιμή ενός videophone το 1960 αλλά η ανάγκη του για σύνδεση με μεγάλο εύρος ζώνης καθιστούσε τη χρήση του μη αποτελεσματική. Το 1970 έγινε συνειδητό πως η οπτική αναγνώριση των επικοινωνούντων πλευρών ενδέχεται να βελτιώσει ουσιαστικά την ποιότητα των πολυμερών συζητήσεων, συνεπώς η εισαγωγή υπηρεσιών επικοινωνίας βίντεο άρχισε να λαμβάνεται υπόψη. Το ενδιαφέρον των χρηστών αυξανόταν παράλληλα με την βελτίωση στην ποιότητα εικόνας.

Οι ενέργειες τυποποίησης κωδικοποιητών βίντεο ξεκίνησαν στις αρχές του 1980. Αυτές οι ενέργειες άρχισαν από τους International Telegraph and Telephone Consultative Committee (CCITT) [12], που είναι πια γνωστοί ως International Telecommunications Union — Telecommunication Standardisation Sector (ITU-T) [13]. Αυτούς τους οργανισμούς τυποποίησης ακολούθησε αργότερα η δημιουργία των Consultative Committee for International Radio (CCIR), αυτή τη στιγμή ITU-R [14], ISO, και International Electrotechnical Commission (IEC).

1.3. ΣΧΕΤΙΚΗ ΔΟΥΛΕΙΑ

Οι μέθοδοι παραλληλοποίησης κωδικοποίησης βίντεο μελετώνται από την εισαγωγή του MPEG-2 τη δεκαετία του 1990. Παραδείγματα αρχικών εργασιών περιλαμβάνει το [15] όπου SIMD εντολές χρησιμοποιήθηκαν για να επιταχύνουν τον κωδικοποιητή.

Πιο πρόσφατες μελέτες σε αυτό το θέμα περιλαμβάνουν τα [16], [17], [18] προς σύντομη αναφορά. Στο [18] ένα σχήμα παραλληλισμού σε επίπεδο slice είχε προταθεί για το H.264/AVC. Το σχήμα βασιζόταν σε επαναπροσδιορισμό του μεγέθους των slices σε σχέση με το βεβαρυμμένο past μέσο όρο χρόνου κωδικοποίησής των macroblocks. Στο HEVC αξιολογήθηκε η παραλληλοποίηση επιπέδου slice με στατικό μέγεθος των Slices ενώ στο [16] αξιολογήθηκε ένα προσαρμοστικό σχήμα διαχωρισμού Slices που αξιοποιεί το πλεονέκτημα της GOP (group of pictures) δομής για να προσφέρει έναν καλύτερο εκτιμητή CTU για καλύτερους χρόνους κωδικοποίησης

Όσο έχει να κάνει με τον παραλληλισμό σε επίπεδο tile, στο [5] οι συγγραφείς αξιολόγησαν τα θέματα επιτάχυνσης και ποιότητας χρησιμοποιώντας ομοιόμορφα στατικά Tiles. Στο [19] οι συγγραφείς αξιολόγησαν πιθανά κέρδη ενέργειας χρησιμοποιώντας μικρότερο αριθμό πυρήνων CPU από το διαθέσιμο και χωρίζοντας την εικόνα σε αυξημένο αριθμό Tiles (ελαχιστοποιώντας έτσι τις πιθανές ανισορροπίες φόρτου). Μια προσέγγιση ανάθεσης πυρήνα tile ένα προς ένα ακολουθήθηκε από τα [20], [21], [22] και [23]. Στο [20] οι συγγραφείς προτείνουν τον επαναπροσδιορισμό των μεγεθών των Tiles βασισμένοι σε εκτιμήσεις του χρόνου κωδικοποίησης των CTUs που αποκτούνται μέσω μια βεβαρυμμένης συνάρτησης που συνδυάζει τρόπο κωδικοποίησης και βάθος CTU. Στο [22] μια παρόμοια μέθοδος διαχωρισμού χρησιμοποιήθηκε και αξιολογήθηκε μαζί με εκτιμητές κόστους CTU που συνδύαζαν την GOP δομή μαζί με πραγματικούς καταγεγραμμένους χρόνους. Ο ίδιος εκτιμητής χρησιμοποιήθηκε στο [23] όπου μια πιο περίπλοκη μορφή διαχωρισμού παρουσιάστηκε που εμπεριέχει μια επαναλαμβανόμενη εφαρμογή ενός βέλτιστου αλγορίθμου διαχωρισμού μιας διάστασης. Σε αντίθεση με τα [20], [22], και [23], που ο σκοπός ήταν η μεγιστοποίηση της επιτάχυνσης εξισορροπώντας τον φόρτο στους πυρήνες της CPU που έχουν ανατεθεί στην κωδικοποίηση των Tiles, στο [21] οι συγγραφείς ερεύνησαν προσαρμοστικό επαναπροσδιορισμό των μεγεθών των tiles με σκοπό την ελαχιστοποίηση της απώλειας ποιότητας που προκαλείται από την χρήση των tiles.

Λίγες εργασίες υπάρχουν στην εφαρμογή διανυσματικών εντολών στην κωδικοποίηση του HEVC π.χ. [24], [25]. Αυτές οι εργασίες συνήθως μελετούν την επίπτωση των SIMD εντολών σε αυτόνομο τρόπο [24] ή σε περιορισμένα σημεία του κωδικοποιητή π.χ. intra κωδικοποίηση στο [25]. Εδώ δίνουμε βάση στην αξιοποίηση του κέρδους εφαρμόζοντας AVX2 εντολές πάνω σε ομοιόμορφο και στατικό Slice και Tile επίπεδο παραλληλοποίησης. Ένα παράδειγμα ομοιόμορφου Slice και Tile διαχωρισμού παρουσιάζεται στην Εικόνα 1. Εάν και πιο προχωρημένες προσεγγίσεις που περιέχουν προσαρμοστικό επαναπροσδιορισμό μεγέθους

παρουσιάζουν μεγαλύτερη βελτίωση στο χρόνο κωδικοποίησης, ως τώρα, δεν έχουν ενσωματωθεί στο reference software. Παρόλα αυτά, μπορούμε ευλόγως να περιμένουμε παρόμοια με αυτής της εργασίας κέρδη στην απόδοση και στους προσαρμοστικούς αλγορίθμους.



Εικόνα 1. Παράδειγμα διαχωρισμού στην ακολουθία Kimono.

(α) Διαχωρισμός σε 4 Slices



(β) Διαχωρισμός σε 4 Tiles

1.4. ΣΥΓΚΡΙΣΗ ΚΩΔΙΚΟΠΟΙΗΤΩΝ

Η σύνταξη του υποκεφαλαίου 1.4.1 βασίστηκε σε απόδοση στα Ελληνικά της [26] .

1.4.1. VP8

Τον Ιανουάριο του 2010, η Google ανακοίνωσε ένα καινούργιο project ανοιχτού προς το κοινό οπτικοακουστικού υλικού “WebM” που έχει ως σκοπό την ανάπτυξη υψηλού επιπέδου τύπο οπτικοακουστικού υλικού ο οποίος θα είναι δωρεάν για όλους. ο πυρήνας του, ένας καινούργιος ανοιχτού κώδικα κωδικοποιητής video, το VP8. Ο τύπος VP8 αρχικά αναπτυσσόταν από μια μικρή ερευνητική ομάδα στην On2 Technologies, Inc. ως διάδοχος της VPx οικογενείας από video codecs. Συγκριτικά με τους υπόλοιπους τύπους κωδικοποίησης video, το VP8 έχει πολλά ξεχωριστά τεχνικά χαρακτηριστικά που το βοηθάνε να επιτύχει μεγάλη απόδοση συμπίεσης με μικρό υπολογιστικό κόστος αποκωδικοποίησης ταυτόχρονα.

Από την αρχή της ανάπτυξης του VP8, οι προγραμματιστές είχαν ως σκοπό τις βασισμένες στο Internet video εφαρμογές. Αυτή η εστίαση έχει οδηγήσει σε αρκετά χαρακτηριστικά του συνολικού σχεδιασμού του.

Μικρές απαιτήσεις εύρους ζώνης:

Ένα από τα βασικά χαρακτηριστικά είναι πως το VP8 σχεδιάστηκε για να λειτουργεί σε συγκεκριμένο εύρος ποιότητας από ~30dB (σε PSNR μετρικές) έως ~45dB.

Ετερογενές υλικό πελάτη.

Υπάρχει ένα μεγάλο εύρος υλικού πελατών συνδεδεμένο στο διαδίκτυο, από χαμηλής ισχύος κινητές και ενσωματωμένες συσκευές έως προχωρημένοι προσωπικοί υπολογιστές με πολλούς επεξεργαστικούς πυρήνες. Πρέπει λοιπόν να είναι δυνατή η κατασκευή πολλαπλών υλοποιήσεων για ένα μεγάλο εύρος από συσκευές.

Τύπος Web Video:

Το VP8 σχεδιάστηκε να χειρίζεται τον τύπο εικόνας που χρησιμοποιείται περισσότερο τα web video: 420 δειγματοληψία χρώματος, 8 bit ανά κανάλι βάθους χρώματος, progressive scan (όχι interlaced), και διαστάσεις εικόνας με μέγιστο τα 16383x16383 pixels. Αυτή η ώθηση για απόδοση συμπίεσης και απλότητα αποκωδικοποιητή οδήγησε σε ιδιαίτερα τεχνικά χαρακτηριστικά στο VP8 [27], σχετικά με άλλους τύπους συμπίεσης video, όπως το MPEG-2 [28], H.263 [29] και H.264/AVC [30].

Υβριδικός μετασχηματισμός και προσαρμοστική κβαντοποίηση.

Το VP8 χρησιμοποιεί 4x4 block-βασισμένο διακριτό μετασχηματισμό συνημιτόνου για όλα τα luma και chroma residual σήματα. Ανάλογα με τον τρόπο πρόβλεψης, οι DC συντελεστές από ένα 16x16 macroblock μπορεί να υποβληθούν σε έναν 4x4 Walsh-Hadamard μετασχηματισμό.

Ευέλικτες εικόνες αναφοράς.

Το VP8 χρησιμοποιεί 3 reference εικόνες για inter πρόβλεψη, αλλά το σχήμα είναι κάπως διαφορετικό από αυτό που έχουμε δει σε διαφορετικούς τύπους κωδικοποιητών σε ότι έχει να κάνει με το motion compensation με πολλαπλές reference εικόνες. Ο σχεδιασμός του περιορίζει την ανάγκη του buffer σε 3 buffers reference εικόνων, και συνεχίζει να επιτυγχάνει αποτελεσματική αποσυσχέτιση στο motion compensation.

Αποδοτική intra και inter πρόβλεψη.

Το VP8 κάνει εκτεταμένη χρήση inter και intra πρόβλεψης. Η intra πρόβλεψη του, έχει μια καινούργια λειτουργία "TM_PRED" ως μια από τις πολλές απλές και αποδοτικές μεθόδους πρόβλεψης. Στην intra πρόβλεψη έχει έναν ευέλικτο τρόπο "SPLITMV" ικανό να κωδικοποιήσει τυχαία block patterns μέσα σε ένα macroblock.

Υψηλών επιδόσεων sub-pixel παρεμβολή.

Το motion compensation του VP8 χρησιμοποιεί ενός τετάρτου ακριβείας διανυσμάτων κίνησης για τα luma pixels και ενός ογδούτου ακριβείας διανυσμάτων κίνησης για τα chroma pixels. Η sub-pixel παρεμβολή του VP8 είναι μια διαδικασία ενός σταδίου με ένα σύνολο από υψηλών επιδόσεων 6 εισόδων φίλτρα.

Προσαρμοστικά in-loop deblocking φίλτρα.

Το VP8 έχει υψηλά προσαρμοστικά in-loop deblocking φίλτρα. Ο τύπος και η ισχύς του φίλτρου μπορεί να ρυθμιστεί για διαφορετικούς τρόπους πρόβλεψης και reference εικόνων.

Προσαρμοστική κωδικοποίηση εντροπίας επιπέδου εικόνας.

Το VP8 χρησιμοποιεί δυαδική αριθμητική κωδικοποίηση εκτεταμένα για σχεδόν όλες τις τιμές των δεδομένων εκτός από λίγα header bits. Τα πλαίσια εντροπίας είναι προσαρμοστικά στο επίπεδο εικόνας, ισορροπώντας έτσι την αποδοτικότητα συμπίεσης και τον υπολογιστικό φόρτο.

Διαχωρισμός δεδομένων φιλικός προς παράλληλη επεξεργασία.

Το VP8 μπορεί να βάλει μαζί τους κωδικοποιημένους συντελεστές μετασχηματισμού σε πολλά τμήματα για να μπορεί να υποστηρίξει παράλληλη επεξεργασία στους αποκωδικοποιητές. Αυτός ο σχεδιασμός βελτιώνει την απόδοση του κωδικοποιητή σε επεξεργαστές πολλαπλών πυρήνων, έχοντας σχεδόν μηδαμινή επίπτωση στην απόδοση συμπίεσης και καθόλου επίδραση στην απόδοση κωδικοποίησης σε μονοπύρηνους επεξεργαστές.

Η σύνταξη του υποκεφαλαίου 1.4.2 βασίστηκε σε απόδοση στα Ελληνικά της [31].

1.4.2. VP9

Όπως αναφέρθηκε παραπάνω, η Google ξεκίνησε το WebM project το 2010, ως μια συλλογική προσπάθεια ανάπτυξης ενός δωρεάν τύπου οπτικοακουστικού υλικού. Το WebM είναι ένας δωρεάν τύπος που δεν καθορίζει μόνο τη δομή του οπτικοακουστικού αρχείου αλλά και τους τύπους video και ήχου. Συγκεκριμένα, τα WebM αρχεία αποτελούνται από το συμπίεσμένο με τον κωδικοποιητή VP9 video [27], και τον συμπίεσμένο με τον δωρεάν Vorbis κωδικοποιητή ήχου [32] και μαζί αυτά τα 2 αποθηκεύονται σε μια μορφή αρχείου βασισμένο στον περιέκτη Matroska [33].

Όμως, για να αντιμετωπιστεί η αυξανόμενη ζήτηση σε video υψηλής ποιότητας για κατανάλωση στο διαδίκτυο, ήταν προφανές πως το VP8 δε θα μπορούσε να ανταπεξέλθει. Στο τέλος του 2011 η Google άρχισε να δουλεύει προς την ανάπτυξη ενός νέας γενιάς video codec, το VP9, με σκοπό τη δημιουργία ενός τύπου που παράγει ένα πολύ πιο συμπιεσμένο bit-stream από τον προκάτοχο του, το VP8, ειδικά σε υψηλής ευκρίνειας περιεχόμενο μόνο με μικρή αύξηση στην πολυπλοκότητα αποκωδικοποίησης. Όλη η τεχνική ανάπτυξη σε σχέση με το VP9 διεξήχθη ανοιχτά στο κοινό πειραματικό branch του libvpx repository του WebM project από τα μέσα του 2012. Όπως και το WebM project, το VP9 είναι και αυτό λογισμικό ανοιχτού κώδικα στο οποίο μπορεί ο καθένας να συμμετάσχει και να συνεισφέρει.

Εργαλεία κωδικοποίησης VP9

Ένα μεγάλο κομμάτι προόδου από το VP8 προς το VP9 ήταν φυσική μετάβαση γενιάς. Η τεχνική πρόκληση ήταν η διαβεβαίωση πως όλα τα εργαλεία κωδικοποίησης δουλεύουν μεταξύ τους με υψηλά αποδοτικό τρόπο. Παρακάτω παρατίθενται τα κύρια εργαλεία που εμπεριέχονται στο VP9.

Μέγεθος Block Προβλέψεων

Ένα μεγάλο κομμάτι βελτιώσεων στην απόδοση κωδικοποίησης στο VP9 οφείλεται στην ενσωμάτωση μεγαλύτερων block προβλέψεις. Το VP9 εισάγει τα super-blocks (SB) μεγέθους μέχρι και 64x64 και επιτρέπει τον διαχωρισμό αυτών χρησιμοποιώντας αναδρομική διάσπαση αυτών, μέχρι και το μέγεθος 4x4 ως εξής. Σε κάθε τετράγωνο επίπεδο block μέσα στη δομή πρόβλεψης υπάρχουν 4 επιλογές. 3 από αυτές αναπαριστούν άκρα μετά τα οποία δεν επιτρέπεται περαιτέρω διάσπαση και περιέχουν: χειρισμό του block γονιού ως ένα ενιαίο τετράγωνο block, ή 2 κάθετα γειτονικά ορθογώνια blocks ή 2 οριζόντια γειτονικά ορθογώνια blocks. Η τέταρτη επιλογή χωρίζει το block γονιό σε 4 τετράγωνα blocks μεγέθους μισού από του block γονιού. Έπειτα η διαδικασία διάσπασης επαναλαμβάνεται με αναδρομικό τρόπο μέχρι τέλους, εάν χρειάζεται, δηλαδή ως το μέγεθος 4x4. Άρα υπάρχουν στο σύνολο 13 διαφορετικά ακριανά μεγέθη block για τον σκοπό του προσδιορισμού των σημάτων πρόβλεψης: 64x64, 64x32, 32x64, 32x32, 32x16, 16x32, 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, και 4x4.

Μέθοδοι πρόβλεψης

INTRA Τρόποι

Το VP9 υποστηρίζει ένα σύνολο 10 τρόπων Intra προβλέψεων για μεγέθη block από 4x4 μέχρι και 32x32: DC_PRED (DC πρόβλεψη), TM_PRED (True-motion πρόβλεψη), H_PRED (Οριζόντια πρόβλεψη), V_PRED (Κάθετη πρόβλεψη), και 6 πλάγιας κατεύθυνσης: D27, D153, D135, D117, D63, D45 αντίστοιχα κατά προσέγγιση με τις γωνίες 27, 153, 135, 117, 63, και 45 μοιρών (μετρώντας αριστερόστροφα ενάντια στον οριζόντιο άξονα. Οι οριζόντιες, κάθετες και πλάγιας κατεύθυνσης τρόποι πρόβλεψης περιλαμβάνουν την αντιγραφή (ή προσέγγιση) των τιμών των εικονοστοιχείων από τα περιβάλλοντα blocks στο τρέχων blocks σύμφωνα με τη γωνία που ορίζεται από τον τρόπο πρόβλεψης.

Για τα blocks που χρησιμοποιούν Intra πρόβλεψη, η διαδικασία κωδικοποίησης (και αποκωδικοποίησης) λειτουργεί σε μονάδες τις οποίες έχει χρησιμοποιήσει ο μετασχηματισμός.

Για παράδειγμα, όταν ένα 16x16 block κωδικοποιείται με Intra τρόπο χρησιμοποιώντας 4x4 μετασχηματισμούς, κάθε 4x4 block σε raster scan σειρά μέσα στο 16x16 block περνάει από πρόβλεψη, κωδικοποίηση και ανακατασκευή πριν από το επόμενο 4x4 block. Άρα τα ακόλουθα 4x4 blocks χρησιμοποιούν ανακατασκευασμένα pixels από τα προηγούμενα Intra κωδικοποιημένα 4x4 blocks.

Inter Τρόποι

Το VP9 υποστηρίζει ένα σύνολο 4 inter τρόπων πρόβλεψης για μέγεθος block από 4x4 μέχρι και 64x64 pixels: NEARESTMV, NEARMV, ZEROMV και NEWMV. Τα προηγούμενα κωδικοποιημένα διανύσματα κίνησης στη γειτονιά της τρέχουσας μονάδας πρόβλεψης ορίζουν 2 υποψήφια vectors - NEARESTMV και NEARMV - τα οποία μπορούν να χρησιμοποιηθούν στους 2 πρώτους παραπάνω τρόπους. Ο τρόπος ZEROMV συνεπάγεται τη χρήση του (0,0) διανύσματος ενώ ο τρόπος NEWMV υποδεικνύει πως ένα καινούργιο διάνυσμα κίνησης θα κωδικοποιηθεί στο stream. Στην τελευταία περίπτωση το πιο κοντινό διάνυσμα κίνησης χρησιμοποιείται και ως παράγοντας πρόβλεψης. Τα διανύσματα κίνησης για τα chroma blocks μέχρι και μεγέθους 8x8 υπολογίζονται ως ο μέσος όρος αυτών που βρίσκονται στα αντίστοιχα luma blocks. Κάθε inter εικόνα μπορεί να κάνει reference σε 3 reference buffers επιλεγμένους από μια συλλογή μέχρι και 8 αποθηκευμένων reference εικόνων όπως ορίζει ένα συντακτικό στοιχείο στο header της εικόνας. Κάθε αποκωδικοποιημένη εικόνα μπορεί να ενημερώσει κάθε υποσύνολο από αυτές τις 8 εικόνες όπως ορίζει ένα άλλο συντακτικό στοιχείο στο header της εικόνας. Η επιλογή του ποιες reference εικόνες θα χρησιμοποιηθούν η ενημερωθούν εξαρτάται από τη στρατηγική που χρησιμοποιεί ο κωδικοποιητής.

Μερικές κωδικοποιημένες εικόνες γνωστές ως alternative reference frames (ARF) μπορούν να χρησιμοποιηθούν μόνο ως reference και να μην παρουσιαστούν ποτέ. Αυτές οι εικόνες ορίζονται ως αόρατες χρησιμοποιώντας ένα flag στο header της εικόνας. Μια πιθανή χρήση μιας ARF είναι να κωδικοποιηθεί μια μέλλουσα εικόνα εκτός σειράς για να βελτιωθεί η αποδοτικότητα της συμπίεσης ενσωματώνοντας μελλοντική πρόβλεψη, αλλά δεν περιορίζεται σε αυτό. Αυτό το framework είναι αρκετά ευέλικτο για να μπορεί να υιοθετηθεί για προχωρημένες δομές πρόβλεψης όπως την χρονική ιεραρχική πρόβλεψη.

Subpel παρεμβολή

Τα φίλτρα που χρησιμοποιούνται για την subpel παρεμβολή για κλασματική κίνηση είναι αποφασιστικής σημασίας για την απόδοση ενός video codec. Στο VP9, η μέγιστη ακρίβεια των διανυσμάτων κίνησης είναι 1/8-pel, με τη δυνατότητα αλλαγής μεταξύ 1/4-pel και 1/8-pel με τη χρήση ενός flag επιπέδου εικόνας. Όμως εάν η ακρίβεια 1/8-pel είναι ενεργοποιημένη σε μια εικόνα χρησιμοποιείται μόνο για μικρή κίνηση, όπως υποδεικνύεται από την τάξη μεγέθους του διανύσματος κίνησης αναφοράς. Για μεγαλύτερη κίνηση - που υποδεικνύεται από μεγαλύτερο reference - υπάρχει συνήθως motion blur, το οποίο καθιστά περιττή την ανάγκη για κίνηση μεγαλύτερης ακριβείας. Προσέχουμε πως για την πρόβλεψη των chroma blocks με 4:2:0 δειγματοληψία χρώματος, η ακρίβεια του διανύσματος κίνησης αυξάνεται κατά 1 bit ώστε η μεγαλύτερη ακρίβεια που απαιτείται να είναι 1/16-pel. Για την πραγματική παρεμβολή για subpel motion, το VP9 υποστηρίζει μια οικογένεια από 3 8-εισόδων φίλτρα - καλούμενα EIGHTTAP,

EIGHTTAP_SHARP και EIGHTTAP_SMOOTH – μαζί με ένα πιο απλό Bilinear φίλτρο , το κάθε ένα με 1/16-pel ακρίβεια. Το τελευταίο χρησιμοποιείται σε σενάρια γρήγορης κωδικοποίησης/αποκωδικοποίησης. Το φίλτρο που θα χρησιμοποιηθεί μπορεί να επιλεγεί στο επίπεδο εικόνας, αλλά υπάρχει και η επιλογή υπόδειξης ενός από τα 3 8-εισόδων φίλτρα για χρήση σε κάθε super-block. Κάθε φίλτρο αναπαρίσταται σε 7-bit ακρίβεια.

Τύποι μετασχηματισμών

Το VP9 χρησιμοποιεί 3 είδη μετασχηματισμών: Διακριτό Μετασχηματισμό Συνημιτόνου (DCT), Ασυμμετρικό Μετασχηματισμό Ημιτόνου (ADST) και Walsh-Hadamard Μετασχηματισμό (WHT). Συγκεκριμένα, για τον τύπο DCT, μπορούν να χρησιμοποιηθούν τετράγωνοι 2D-DCT μεγέθους 4x4, 8x8, 16x16 και 32x32 μετασχηματισμοί. Όλα τα Inter κωδικοποιημένα blocks χρησιμοποιούν έναν από αυτούς τους 4 μετασχηματισμούς. Για τα Intra κωδικοποιημένα blocks το VP9 υποστηρίζει επίσης και 4-pt, 8pt και 16-pt Ασυμμετρικούς Μετασχηματισμούς Ημιτόνου (ADSTs) [34], οι οποίοι μπορούν να συνδυαστούν με 1-D DCT ίδιου μεγέθους - ανάλογα με το μέγεθος στον επιλεγμένο Intra τρόπο πρόβλεψης - για να σχηματιστεί ένας 2-D υβριδικός μετασχηματισμός. Οι πιθανοί συνδυασμοί μετασχηματισμών που χρησιμοποιούνται στις οριζόντιες και κάθετες κατευθύνσεις στην Intra πρόβλεψη είναι (ADST, ADST), (ADST, DCT), (DCT, ADST) και (DCT,DCT), για 4x4, 8x8 και 16x16 μεγέθη block. Ένας τελευταίος μετασχηματισμός που υποστηρίζεται από το VP9 είναι ο 4x4 Walsh-Hadamard Μετασχηματισμός ο οποίος χρησιμοποιείται μόνο στο χαμηλότερο επίπεδο κβαντοποίησης (δείκτης $q = 0$) για να παρέχει μη απολεστική κωδικοποίηση. Προσέχουμε πως στο σύνολο υπάρχουν 14 πιθανοί 2-D μετασχηματισμοί που υποστηρίζονται από το VP9, ο κάθε ένας σε τετράγωνο μέγεθος.

Κωδικοποίηση εντροπίας και προσαρμοστικότητα

Το bit-stream του VP9 είναι αριθμητικά κωδικοποιημένο. Έστω ένα σύμβολο από ένα n -αδικό αλφάβητο, φτιάχνεται ένα δυαδικό δέντρο με $n-1$ μη-φύλλα κόμβους, και ένας δυαδικός αριθμητικώς κωδικοποιητής τρέχει σε κάθε τέτοιο κόμβο για να κωδικοποιήσει ένα σύμβολο. Οι πιθανότητες σε κάθε κόμβο χρησιμοποιούν 8-bit ακρίβεια και είναι στο $[1, 255]$. Αναφερόμαστε στο σύνολο των $n-1$ πιθανοτήτων για κωδικοποίηση του συμβόλου ως το context κωδικοποίησης εντροπίας. Τα περισσότερα από τα σύμβολα που μεταφέρονται στο VP9 bit-stream χρησιμοποιούν αυτή τη μεθοδολογία.

Η σύνταξη του υποκεφαλαίου 1.4.1 βασίστηκε σε απόδοση στα Ελληνικά της [3].

1.4.3. H.264/AVC

Στις αρχές του 1998, οι Video Coding Experts Group (VCEG) ITU-T SG16 Q.6 εκδώσαν μια πρόσκληση υποβολής προτάσεων σε ένα project με όνομα H.26L, με σκοπό να διπλασιάσουν την αποδοτικότητα κωδικοποίησης σε σχέση με όλα τα υπόλοιπα πρότυπα κωδικοποίησης που υπήρχαν μέχρι στιγμής για ένα ευρύ φάσμα εφαρμογών. Το πρώτο προσχέδιο για το καινούργιο

πρότυπο υιοθετήθηκε τον Οκτώβρη του 1999. Το Δεκέμβρη του 2001, οι VCEG και οι Moving Picture Experts Group (MPEG) ISO/IEC JTC 1/SC 29/WG 11 σχημάτισαν τους Joint Video Team (JVT), με σκοπό την ολοκλήρωση του προσχεδίου για το νέο πρότυπο κωδικοποίησης για επίσημη υποβολή ως H.264/AVC [35] τον Μάρτιο του 2003.

Το καινούργιο πρότυπο ήταν σχεδιασμένο για τεχνικές λύσεις που περιείχαν τουλάχιστον τα παρακάτω τμήματα εφαρμογών: διαδραστική ή σειριακή αποθήκευση σε οπτικά και μαγνητικά μέσα, οπτικοακουστικό υλικό κατά απαίτηση, broadcast μέσω δορυφόρου ή DSL, υπηρεσίες ανταλλαγής μηνυμάτων κλπ.

Το H.264/AVC αναπτύχθηκε λαμβάνοντας υπόψιν την επέκταση και τη δημιουργία καινούργιων εφαρμογών οι οποίες θα χρησιμοποιούσαν τα υπάρχοντα και τα μελλοντικά δίκτυα. Αυτό σημαίνει πως θα έπρεπε να είναι ικανό να χειριστεί μια ποικιλία εφαρμογών και δικτύων.

Για αυτό το λόγο το σχέδιο του H.264/AVC καλύπτει ένα VCL (Video Coding Layer) το οποίο είναι σχεδιασμένο να απεικονίσει αποδοτικά το οπτικό υλικό, και ένα NAL (Network Abstraction Layer) το οποίο μορφοποιεί την αναπαράσταση του VCL και παρέχει header πληροφορίες με έναν τρόπο κατάλληλο για την μετάδοση του από μια ποικιλία από επίπεδα μεταφοράς ή μέσα αποθήκευσης. Όπως και σε προηγούμενες μεθόδους κωδικοποίησης video, μερικά χαρακτηριστικά του σχεδιασμού που καθιστούν ικανή τη βελτιωμένη αποδοτικότητα στην κωδικοποίηση συμπεριλαμβάνουν τις ακόλουθες βελτιστοποιήσεις της ικανότητας πρόβλεψης των τιμών του περιεχομένου μιας εικόνας που κωδικοποιείται.

Μεταβλητό μέγεθος από blocks και motion compensation με μικρά μεγέθη blocks.

Αυτό το πρότυπο υποστηρίζει μεγαλύτερη ευελιξία στην επιλογή motion compensation μεγέθη block και σχήματα από άλλα προηγούμενα πρότυπα, με ένα ελάχιστο luma motion compensation μέγεθος block όσο μικρό όσο 4x4.

Motion compensation ακριβείας 1/4.

Τα περισσότερα προηγούμενα πρότυπα παρέχουν ακρίβεια διανύσματος κίνησης μέχρι το πολύ 1/2. Ο σχεδιασμός αυτός βελτιώνει αυτήν την ακρίβεια προσθέτοντας 1/4 ακρίβεια, όπως βρίσκεται και στο advanced profile του MPEG-4 Visual (part 2) προτύπου, αλλά ταυτόχρονα μειώνει και την πολυπλοκότητα της διαδικασίας παρεμβολής σε σχέση με τον προηγούμενο σχεδιασμό.

Διανύσματα κίνησης πέρα από όρια εικόνων.

Ενώ τα διανύσματα κίνησης στο MPEG-2 και στους προκατόχους του έπρεπε να δείχνουν μόνο σε περιοχές μέσα στην προηγουμένως κωδικοποιημένη εικόνα αναφοράς, η τεχνική προεκβολής, που πρώτη εμφανίστηκε ως μια προαιρετική επιλογή στο H.263, συμπεριλαμβάνεται στο H.264/AVC.

Motion compensation πολλαπλών εικόνων αναφοράς.

Οι κωδικοποιημένες με πρόβλεψη εικόνες (ονόματι “P” εικόνες) στο MPEG-2 και στους προκατόχους του χρησιμοποιούν μόνο μια προηγούμενη εικόνα για να προβλέψουν τις τιμές σε μια ερχόμενη εικόνα. Ο καινούργιος σχεδιασμός επεκτείνει τη μέθοδο επιλογής εικόνας αναφοράς που βρίσκεται στο H.263++ για να κάνει δυνατή την αποδοτική κωδικοποίηση επιτρέποντας στον κωδικοποιητή να επιλέξει, για λόγους motion compensation, μεταξύ ενός μεγαλύτερου συνόλου αποκωδικοποιημένων εικόνων που είναι αποθηκευμένες στον αποκωδικοποιητή.

Κατευθυντική χωρική πρόβλεψη για Intra κωδικοποίηση.

Μια νέα τεχνική για την προεκβολή των ακμών και των προηγουμένως αποκωδικοποιημένων τμημάτων της τρέχουσας εικόνας εφαρμόζεται σε περιοχές που οι εικόνες έχουν κωδικοποιηθεί ως intra. Αυτό βελτιώνει την ποιότητα του σήματος που έχει προβλεφθεί και επιτρέπει την πρόβλεψη από γειτονικές περιοχές που δεν έχουν κωδικοποιηθεί με intra τρόπο.

In-the-loop deblocking φίλτρο.

Η κωδικοποίηση βασισμένη σε block παράγει artifacts, γνωστά ως blocking artifacts. Αυτά μπορούν να προέλθουν από τα στάδια πρόβλεψης και residual διαφοράς κωδικοποίησης κατά τη φάση της αποκωδικοποίησης. Η εφαρμογή αυτών των προσαρμοστικών deblocking φίλτρων είναι μια αρκετά γνωστή μέθοδος βελτίωσης της ποιότητας εικόνας, και όταν είναι σχεδιασμένα καλά, αυτά μπορούν να βελτιώσουν την υποκειμενική και αντικειμενική ποιότητα της εικόνας.

Μετασχηματισμός block μικρού μεγέθους.

Όλα τα προηγούμενα μεγάλα πρότυπα κωδικοποίησης χρησιμοποιούσαν block μετασχηματισμού μεγέθους 8x8, ενώ ο σχεδιασμός του H.264/AVC είναι κυρίως βασισμένος στον 4x4 μετασχηματισμό. Αυτό επιτρέπει στον κωδικοποιητή να αναπαραστήσει τα σήματα με έναν πιο τοπικά προσαρμοστικό τρόπο, ο οποίος μειώνει τα artifacts.

Ιεραρχικός block μετασχηματισμός.

Ενώ στις περισσότερες περιπτώσεις, η χρήση μικρού 4x4 block μετασχηματισμού είναι αντιληπτά ωφέλιμη, υπάρχουν μερικά σήματα τα οποία περιέχουν αρκετή συσχέτιση ώστε να απαιτούν μερικές μεθόδους χρήσης αναπαραστάσεις με μεγαλύτερες βασικές συναρτήσεις. Το H.264/AVC πρότυπο καθιστά αυτό ικανό με 2 τρόπους: 1) χρησιμοποιώντας ιεραρχικό μετασχηματισμό για την επέκταση του μεγέθους του block size για την χαμηλής συχνότητας chroma πληροφορία σε 8x8 πίνακα και 2) επιτρέποντας τον κωδικοποιητή να επιλέξει έναν ειδικό τρόπο κωδικοποίησης για intra, επεκτείνοντας έτσι το μήκος του luma μετασχηματισμού για την πληροφορία χαμηλής συχνότητας σε 16x16 μέγεθος με τρόπο παρόμοιο με αυτόν που εφαρμόζεται στο chroma.

Ακριβής αντίστροφος μετασχηματισμός.

Στα προηγούμενα πρότυπα κωδικοποίησης, ο μετασχηματισμός που χρησιμοποιούνταν για το βίντεο ήταν γενικά προσδιορισμένος μέχρι ένα όριο, λόγω της μη πρακτικότητας της απόκτησης

της ακριβής αντιστοιχίας του αντιστρόφου μετασχηματισμού. Ως αποτέλεσμα, κάθε σχεδιασμός κωδικοποιητή παρήγαγε ελαφρώς διαφορετικό αποκωδικοποιημένο video, προκαλώντας έτσι μια διαφοροποίηση ανάμεσα στην αναπαράσταση κωδικοποιητή και αποκωδικοποιητή του video και μειώνοντας έτσι την πραγματική ποιότητα του. Συνεχίζοντας από το προαιρετικό χαρακτηριστικό του H.263++, το H.264/AVC είναι το πρώτο πρότυπο που πετύχαινε συγκεκριμένο αποκωδικοποιημένο video από όλους τους αποκωδικοποιητές.

Αριθμητική κωδικοποίηση εντροπίας.

Μια προχωρημένη μέθοδος κωδικοποίησης εντροπίας γνωστή και ως αριθμητική κωδικοποίηση συμπεριλαμβάνεται στο H.264/AVC. Ενώ η αριθμητική κωδικοποίηση ήταν μια προαιρετική επιλογή στο H.263, μια πιο αποτελεσματική χρήση αυτής της τεχνικής βρίσκεται στο H.264/AVC για να δημιουργήσει μια ισχυρή μέθοδο κωδικοποίησης εντροπίας γνωστή ως CABAC (context-adaptive binary arithmetic coding).

1.5. ΣΤΟΧΟΙ ΚΑΙ ΟΡΓΑΝΩΣΗ ΠΤΥΧΙΑΚΗΣ

Σε αυτήν την εργασία αξιολογούμε τον συνδυασμό των δυο coarse-grained προσεγγίσεων στο HEVC, δηλαδή, παραλληλοποίησης επιπέδου Slice και Tile, μαζί με SIMD (Single Instruction Multiple Data) υλοποίηση του SAD (Sum of Absolute Difference) και SSE (Sum of Squared Error) μέρους του κωδικοποιητή. Η αξιολόγηση αυτή είναι βασισμένη στο reference software HM16.7 [36], με Slice και Tile παραλληλοποίησης υλοποιημένη με OpenMP [37] και SIMD υλοποιημένο με τη μέθοδο AVX2 εντολών [38] σε ένα chip της Intel (i7 7700k). Η συνεισφορά μας περιλαμβάνει τα ακόλουθα.

- Προσφέρουμε λεπτομέρειες υλοποίησης στην εφαρμογή του συνδυασμού coarse-fine grained παραλληλοποίησης στο reference software.
- Αξιολογούμε το αποτέλεσμα με τη χρήση κοινών δοκιμαστικών ακολουθιών video [39].

Τα αποτελέσματα υποδεικνύουν ότι και οι δυο υλοποιήσεις παραλληλισμού, δηλαδή AVX2 εντολές μαζί με Slices και AVX2 μαζί με Tiles είναι πιο γρήγορες κατά περίπου 11% σε σύγκριση με τις αυτόνομες υλοποιήσεις. Σε πολλές περιπτώσεις ο συνδυασμός των Slices και Tiles μαζί με AVX2 εντολές οδήγησε σε superlinear επιτάχυνση. Επιπλέον, η πτώση στην ποιότητα video (εάν υπήρχε) ήταν μηδαμινή. Να σημειωθεί ότι μέρος της πτυχιακής εργασίας δημοσιεύτηκε στο [40].

Τέλος, η υπόλοιπη εργασία είναι οργανωμένη ως εξής: Στο Κεφάλαιο 2 γίνεται μια επισκόπηση του κωδικοποιητή HEVC, μαζί με πλήρης ανάλυση τεχνικών κωδικοποίησης, στο Κεφάλαιο 3 περιγράφονται οι τεχνικές παραλληλοποίησης και η υλοποίηση του συνδυασμού AVX2 εντολών με Slices/Tiles, στο Κεφάλαιο 4 γίνεται αναφορά και αξιολόγηση των πειραμάτων και στο Κεφάλαιο 5 υπάρχει ο επίλογος με τέλος, την μελλοντική δουλειά που έπεται στον κωδικοποιητή.

2^ο ΚΕΦΑΛΑΙΟ

ΕΠΙΣΚΟΠΗΣΗ HEVC

Η σύνταξη του κεφαλαίου 2 βασίστηκε σε απόδοση στα Ελληνικά της [2].

2.1. Εισαγωγή

Το High Efficiency Video Coding (HEVC) είναι το πιο πρόσφατο πρότζεκτ των ITU-T Video Coding Experts Group (VCEG) και των ISO/IEC Moving Picture Experts Group (MPEG) δουλεύοντας μαζί ως Joint Collaborative Team on Video Coding (JCT-VC) [41]. Τα πρότυπα κωδικοποίησης video έχουν εξελιχθεί κυρίως από την ανάπτυξη των προτύπων ITU-T and ISO/IEC. Οι ITU-T παρήγαγαν το H.261 [42] και H.263 [29], οι ISO/IEC το MPEG-1 [43] και MPEG-4 Visual [44] και μαζί το H.262/MPEG-2 Video [45] και H.264/MPEG-4 Advanced Video Coding (AVC) [46]. Τα πρότυπα που αναπτύχθηκαν από κοινού είχαν μεγάλο αντίκτυπο και υιοθετήθηκαν από μια μεγάλη ποικιλία προϊόντων συσχετισμένα με την καθημερινή ζωή. Η κίνηση δεδομένων που προκαλείται από εφαρμογές video με στόχο προσωπικούς υπολογιστές και tablets επιβάλλουν μια μεγάλη πρόκληση στα σημερινά δίκτυα. Επίσης, μια μεγαλύτερη πια επιθυμία για καλύτερη ποιότητα και μεγαλύτερες αναλύσεις υπάρχει σε mobile εφαρμογές. Η ανάγκη λοιπόν για μεγαλύτερη απόδοση συμπίεσης από το H.264/MPEG-4 AVC σε HD και πάνω μορφές video οδήγησε στη δημιουργία του HEVC.

A. Video Coding Layer

Το video coding layer του HEVC επιστρατεύει την ίδια υβριδική προσέγγιση (intra/interpicture πρόβλεψη και κωδικοποίηση 2-D μετασχηματισμού) που χρησιμοποιείται σε όλα τα πρότυπα συμπίεσης από το H.261.

Ένας αλγόριθμος κωδικοποίησης που φτιάχνει ένα bitstream σύμφωνα με το πρότυπο του HEVC συνήθως λειτουργεί ως εξής. Κάθε εικόνα χωρίζεται σε block περιοχές, με τις πληροφορίες διαχωρισμού και διαστάσεων να στέλνονται στον αποκωδικοποιητή. Η πρώτη εικόνα κάθε ακολουθίας (και η πρώτη εικόνα σε κάθε καινούργιο random access σημείο) κωδικοποιείται χρησιμοποιώντας intra πρόβλεψη (που χρησιμοποιεί πρόβλεψη από κοντινά χωρικά δεδομένα μέσα στην ίδια εικόνα, αλλά δεν εξαρτάται από άλλες εικόνες). Για τις υπόλοιπες εικόνες μιας ακολουθίας ή μεταξύ random access points, χρησιμοποιείται συνήθως inter χρονικής πρόβλεψης κωδικοποίηση. Η διαδικασία κωδικοποίησης inter πρόβλεψης αποτελείται από την επιλογή των δεδομένων κίνησης που συνιστούν τη εικόνα αναφοράς και το διάνυσμα κίνησης (MV) που θα εφαρμοστεί για την πρόβλεψη των δειγμάτων κάθε block. Ο κωδικοποιητής και ο αποκωδικοποιητής παράγουν όμοια σήματα inter πρόβλεψης εφαρμόζοντας motion compensation χρησιμοποιώντας το MV και mode decision δεδομένα, που μεταφέρονται ως παραπάνω πληροφορία.

Το residual σήμα της intra η inter πρόβλεψης, το οποίο είναι η διαφορά ανάμεσα στο αρχικό block και στην πρόβλεψη του, μετασχηματίζεται από έναν γραμμικό χωρικό μετασχηματισμό.

Έπειτα, οι συντελεστές του μετασχηματισμού πολλαπλασιάζονται, κβαντοποιούνται, κωδικοποιούνται με χρήση κωδικοποίησης εντροπίας και μεταφέρονται μαζί με την πληροφορία της πρόβλεψης.

Ο κωδικοποιητής αναπαράγει τη διαδικασία του αποκωδικοποιητή με σκοπό να παράγουν και τα 2 όμοιες προβλέψεις για τα ακόλουθα δεδομένα. Επομένως, οι κβαντισμένοι συντελεστές φτιάχνονται χρησιμοποιώντας αντίστροφο κβαντισμό και έπειτα αντίστροφο μετασχηματισμό για την αναπαραγωγή της αποκωδικοποιημένης προσέγγισης του residual σήματος. Το residual έπειτα προστίθεται στην πρόβλεψη και το αποτέλεσμα αυτής τις προσθέσεις περνάει από ένα ή δυο loop φίλτρα για να εξομαλυνθούν τα artifacts που προκλήθηκαν από την block επεξεργασία και τον κβαντισμό. Η τελική αναπαράσταση της εικόνας (που είναι ένα διπλότυπο της εξόδου του κωδικοποιητή) αποθηκεύεται σε έναν buffer αποκωδικοποιημένων εικόνων για να χρησιμοποιηθεί στην πρόβλεψη των ακολούθων εικόνων. Γενικά, η διαδικασία κωδικοποίησης και αποκωδικοποίησης των εικόνων συχνά διαφέρει από την σειρά με την οποία φτάνουν από την πηγή, καθιστώντας έτσι αναγκαία τη διαφοροποίηση ανάμεσα στη σειρά αποκωδικοποίησης και τη σειρά εξόδου για έναν αποκωδικοποιητή.

Συνήθως, το οπτικό υλικό προς κωδικοποίηση από το HEVC υποτίθεται πως είναι progressive scan imagery (είτε λόγο της πηγής είτε ως αποτέλεσμα deinterlacing πριν την κωδικοποίηση). Δεν υπάρχουν τεχνικές κωδικοποίησης video στο HEVC που χρησιμοποιούν interlaced scanning, καθώς δε χρησιμοποιείται πια και γίνεται ουσιαστικά λιγότερο κοινό για μετάδοση.

Όμως, μια σύνταξη μεταδεδομένων έχει προσφερθεί στο HEVC για να επιτρέψει στον κωδικοποιητή να υποδείξει πως έχει σταλεί interlaced-scanned video, κωδικοποιώντας κάθε περιοχή (τις ζυγές ή της μονές γραμμές κάθε εικόνας) του interlaced video ως διαφορετική εικόνα η κωδικοποιώντας κάθε interlaced εικόνα ως HEVC κωδικοποιημένη εικόνα. Αυτό προσφέρει μια αποδοτική μέθοδο κωδικοποίησης interlaced video χωρίς να επιβαρύνονται οι αποκωδικοποιητές με την ανάγκη υποστήριξης ειδικής διαδικασίας αποκωδικοποίησης για αυτό.

Τα ακόλουθα χαρακτηριστικά περιέχονται στην υβριδικό κωδικοποίηση video:

1. **Coding tree units και coding tree block δομή.** Ο πυρήνας του coding layer στα προηγούμενα πρότυπα ήταν το macroblock, που περιείχε ένα 16x16 block από luma δείγματα, συνήθως 4:2:0 δειγματοληψία χρώματος, 2 αντίστοιχα 8x8 chroma δείγματα, όπου στο HEVC η ανάλογη μορφή είναι το coding tree unit (CTU), όπου το μέγεθος του καθορίζεται από τον κωδικοποιητή και μπορεί να είναι μεγαλύτερο από ένα παραδοσιακό macroblock. Το CTU αποτελείται από ένα luma CTB και τα αντίστοιχα chroma CTBs και συντακτικά στοιχεία. Το μέγεθος LxL ενός luma CTB μπορεί να επιλεγεί ως L= 16, 32 ή 64 δείγματα, με μεγαλύτερα μεγέθη συνήθως να καθιστούν τη συμπίεση πιο αποδοτική. Το HEVC υποστηρίζει το διαχωρισμό των CTBs σε μικρότερα blocks χρησιμοποιώντας μια δομή δέντρου και quad-tree like signaling.
2. **Coding Units (CUs) και coding blocks (CBs):** Η quad-tree σύνταξη του CTU προσδιορίζει το μέγεθος και τις θέσεις των luma και chroma CBs. Η ρίζα του quadtree είναι συσχετισμένη

με το CTU. Οπότε, το μέγεθος του luma CTB είναι το μεγαλύτερο υποστηριζόμενο μέγεθος για ένα luma CB. Ο διαχωρισμός ενός CTU σε luma και chroma CBs σηματοδοτούνται από κοινού. Ένα luma CB και συνήθως 2 chroma CBs, μαζί με συσχετισμένη σύνταξη, φτιάχνουν ένα coding unit (CU). Ένα CTB μπορεί να περιέχει μόνο ένα CU ή μπορεί να χωριστεί σε πολλά CUs, και κάθε CU να έχει ένα συσχετισμένο διαχωρισμό σε prediction units (PUs) και ένα δέντρο από transform units (TUs).

3. **Prediction units και prediction blocks (PBs):** Η απόφαση του αν κωδικοποιηθεί ένα μέρος της εικόνας χρησιμοποιώντας inter ή intra κωδικοποίηση γίνεται στο επίπεδο του CU. Μια δομή διαχωρισμού του PU έχει τη ρίζα της στο CU επίπεδο. Ανάλογα με την απόφαση του βασικού τύπου πρόβλεψης, τα luma και chroma CBs μπορούν να χωριστούν περαιτέρω στο μέγεθος, και να προβλεφθούν από luma και chroma prediction blocks (PBs). Το HEVC υποστηρίζει μεταβλητό μέγεθος PB από 64x64 μέχρι και 4x4.
4. **TUs και transform blocks:** Το προβλεφθέν residual κωδικοποιείται χρησιμοποιώντας block μετασχηματισμούς. Μια δομή δέντρου TU έχει τη ρίζα της στο CU επίπεδο. Το luma CB residual μπορεί να είναι όμοιο με το luma block μετασχηματισμού (TB) ή μπορεί να χωριστεί περαιτέρω σε μικρότερα luma TBs. Το ίδιο ισχύει και στα chroma TBs. Συναρτήσεις ακεραίας βάσης παρόμοιες με αυτές του διακριτού μετασχηματισμού συνημιτόνου (DCT) ορίζονται για τα τετράγωνα TB μεγέθους 4x4, 8x8, 16x16, και 32x32. Για τον 4x4 μετασχηματισμό των προβλεφθέντων luma intra residual σημάτων, ορίζεται ένας ακέραιος μετασχηματισμός που προέρχεται από μια μορφή του διακριτού μετασχηματισμού ημιτόνου.
5. **Motion vector signaling:** Advanced motion vector prediction (AMVP) χρησιμοποιείται, περιέχοντας προέλευση από πολλούς πιθανούς υποψηφίους βασισμένο στα δεδομένα από γειτονικά PBs και από τη εικόνα αναφοράς. Ένα merge mode για την κωδικοποίηση MV μπορεί να χρησιμοποιηθεί, επιτρέποντας την κληρονόμηση των MVs από χρονικά ή χωρικά γειτονικά PBs. Επιπλέον, σε σύγκριση με το H.264/MPEG-4 AVC ορίζεται βελτιωμένο skipped και direct motion inference.
6. **Motion Compensation:** Ακρίβεια ενός τετάρτου χρησιμοποιείται για τα MVs και φίλτρα 7 ή 8 εισόδων για παρεμβολή κλασματικών θέσεων (σε σύγκριση με τα φίλτρα 6 εισόδων και κλασματικές θέσεις 1/2 ακολουθούμενα από γραμμική παρεμβολή 1/4 θέσεις στο H.264/MPEG-4 AVC). Παρομοίως με το H.264/MPEG-4 AVC, πολλαπλές εικόνες αναφοράς μπορούν να χρησιμοποιηθούν. Για κάθε PB μπορούν να μεταδοθούν ένα ή δυο διανύσματα κίνησης, με αποτέλεσμα unipredictive ή bipredictive κωδικοποίηση αντίστοιχα. Όπως στο H.264/MPEG-4 AVC, μια scaling και offset διαδικασία μπορεί να εφαρμοστεί στο προβλεφθέν σήμα με έναν τρόπο γνωστό ως βεβαρυμμένη πρόβλεψη.
7. **Intra prediction:** Τα ακριανά αποκωδικοποιημένα δείγματα γειτονικών blocks χρησιμοποιούνται ως δεδομένα αναφοράς για χωρικοί πρόβλεψη σε περιοχές όπου η inter πρόβλεψη δεν εφαρμόζεται. Η intra πρόβλεψη υποστηρίζει 33 κατευθυντικά modes (σε σύγκριση με τα 8 modes στο H.264/MPEG-4 AVC), μαζί με planar και dc modes πρόβλεψης. Τα επιλεγμένα intra modes κωδικοποιούνται παίρνοντας τα πιο πιθανά modes βασισμένα στα προηγούμενα αποκωδικοποιημένα γειτονικά PBs.
8. **Έλεγχος κβαντισμού:** Όπως στο H.264/MPEG-4 AVC, uniform reconstruction quantization (URQ) χρησιμοποιείται στο HEVC με με quantization scaling πίνακες να υποστηρίζονται για διαφορά μεγέθη block μετασχηματισμού.

9. **Κωδικοποίηση εντροπίας:** Context adaptive binary arithmetic context (CABAC) χρησιμοποιείται για κωδικοποίηση εντροπίας. Αυτό είναι παρόμοιο με το CABAC σχήμα στο H.264/MPEG-4 AVC αλλά έχει υποστεί πολλές βελτιώσεις για να αυξηθεί η ταχύτητα (ειδικά για παράλληλες αρχιτεκτονικές) και η απόδοση συμπίεσης και για να μειωθεί η χρήση μνήμης του.
10. **In-Loop deblocking φίλτράρισμα:** Το deblocking φίλτρο που χρησιμοποιείται είναι παρόμοιο με αυτό του H.264/MPEG-4 AVC και λειτουργεί μέσα στο βρόγχο του της inter πρόβλεψης. Όμως ο σχεδιασμός έχει απλοποιηθεί σε ότι έχει να κάνει με τη λήψη αποφάσεων και την διαδικασία φιλτραρίσματος για να είναι πιο φιλικό σε παράλληλη επεξεργασία.
11. **Sample adaptive offset (SAO):** Ένα μη γραμμικό amplitude mapping εισάγεται στο βρόγχο inter πρόβλεψης μετά το deblocking φίλτρο. Ο σκοπός του είναι να ξαναφτιάξει το πραγματικό πλάτος των σημάτων χρησιμοποιώντας ένα look up table που περιγράφεται από μερικές επιπλέον παραμέτρους που μπορούν οριστούν από ανάλυση ιστογραμμάτων στη πλευρά του κωδικοποιητή.

B. Αρχιτεκτονική σύνταξης υψηλού επιπέδου.

Ένας αριθμός από καινούργια σχέδια βελτιώνουν την ευελιξία για τη λειτουργία πάνω σε μια ποικιλία εφαρμογών και δικτυακών περιβαλλόντων και βελτιώνουν την αντοχή σε απώλεια δεδομένων. Όμως, η αρχιτεκτονική σύνταξης υψηλού επιπέδου που χρησιμοποιήθηκε στο H.264/MPEG-4 AVC πρότυπο έχει γενικά μείνει, συμπεριλαμβανομένου και των ακολούθων χαρακτηριστικών.

1. **Parameter set structure:** Τα σύνολα παραμέτρων περιέχουν πληροφορία η οποία μπορεί να μοιραστεί για την αποκωδικοποίηση πολλών περιοχών του αποκωδικοποιημένου video. Η δομή συνόλου παραμέτρων προσφέρει έναν ισχυρό μηχανισμό για την μετάδοση δεδομένων που είναι απαραίτητα για τη διαδικασία αποκωδικοποίησης. Οι έννοιες των συνόλων παραμέτρων εικόνας και ακολουθίας από το H.264/MPEG-4 AVC συγχωνεύονται σε μια καινούργια δομή συνόλου παραμέτρων video (VPS).
2. **NAL unit syntax structure:** κάθε δομή σύνταξης τοποθετείται σε ένα λογικό πακέτο δεδομένων που καλείται network abstraction layer (NAL) unit. Χρησιμοποιώντας το περιεχόμενο του 2-byte NAL unit header, είναι δυνατό να αναγνωριστεί ο σκοπός του συσχετισμένου πακέτου δεδομένων.
3. **Slices.** Το slice είναι μια δομή δεδομένων που μπορεί να αποκωδικοποιηθεί ανεξάρτητα από τα άλλα slices στην ίδια εικόνα, σε ότι έχει να κάνει με την κωδικοποίηση εντροπίας, την πρόβλεψη σήματος και την ανακατασκευή του residual σήματος. Ένα slice μπορεί να είναι ολόκληρη η εικόνα ή μέρος αυτής. Ένας από τους κύριους σκοπούς των slices είναι ο επανασυγχρονισμός κατά την περίπτωση απώλειας δεδομένων. Στην περίπτωση πακετοποιημένης μετάδοσης, ο μέγιστος αριθμός payload bits μέσα σε ένα slice είναι συνήθως περιορισμένος, και ο αριθμός των CTUs σε ένα slice συχνά ποικίλλει για να ελαχιστοποιήσει τον φόρτο πακετοποίησης ενώ ταυτόχρονα να κρατήσει το μέγεθος του κάθε πακέτου μέσα στο όριο του.

4. **Supplemental enhancement information (SEI) και video usability information (VUI) metadata:** Η σύνταξη συμπεριλαμβάνει υποστήριξη για πολλούς τύπους μεταδεδομένων γνωστού ως SEI και VUI. Τέτοια δεδομένα παρέχουν πληροφορίες για τον χρονισμό των εικόνων του video, την σωστή απεικόνιση του χρωματικού χώρου που χρησιμοποιείται στο σήμα του video , 3-D stereoscopic frame packing info display hint πληροφορίες κλπ.

C. Parallel Decoding Syntax και Modified Slice Structuring

Τέλος , τέσσερα καινούργια χαρακτηριστικά εισάγονται στο πρότυπο HEVC για να βελτιώσουν την ικανότητα παράλληλης επεξεργασίας ή να τροποποιήσουν τη δομή των slice δεδομένων για λόγους πακετοποίησης. Κάθε ένα από αυτά έχει πλεονεκτήματα σε συγκεκριμένα σενάρια εφαρμογών και είναι γενικά πάνω στον υλοποιητή του κωδικοποιητή ή αποκωδικοποιητή να αποφάσισε εάν και πως θα εκμεταλλευτεί αυτά τα χαρακτηριστικά.

1. **Tiles.** Η επιλογή του διαχωρισμού μιας εικόνας σε ορθογώνιες περιοχές ονόματι tiles έχει οριστεί. Ο κύριος σκοπός των tiles είναι να βελτιώσουν τη δυνατότητα παράλληλης επεξεργασίας παρά την ανθεκτικότητα σε σφάλματα. Τα tiles είναι ανεξάρτητες αποκωδικοποιούμενες περιοχές μιας εικόνας που κωδικοποιούνται με κάποιες κοινές header πληροφορίες. Τα tiles μπορούν επίσης να χρησιμοποιηθούν για το σκοπό του χωρικού random access σε τοπικές περιοχές εικόνων video. Ένα σύνηθες tile configuration μιας εικόνας αποτελείται από την τμηματοποίηση της εικόνας σε ορθογώνιες περιοχές με περίπου ίσο αριθμό από CTUs σε κάθε tile. Τα tiles προσφέρουν παραλληλισμό σε ένα πιο coarse επίπεδο granularity(εικόνας/υποεικόνας), και δε χρειάζεται προχωρημένος συγχρονισμός των threads για τη χρήση τους.
2. **Wavefront parallel processing.** Όταν το wavefront parallel processing (WPP) είναι ενεργοποιημένο, το slice χωρίζεται σε σειρές από CTUs. Η πρώτη γραμμή επεξεργάζεται με κανονικό τρόπο, και η δεύτερη γραμμή μπορεί να ξεκινήσει να επεξεργάζεται μόνο μετά την ολοκλήρωση επεξεργασίας 2 τουλάχιστον CTUs από την προηγούμενη σειρά , η τρίτη γραμμή μπορεί να αρχίσει να επεξεργάζεται μόνο μετά την ολοκλήρωση επεξεργασίας τουλάχιστον 2 CTUs από τη δεύτερη σειρά και ούτω καθεξής. Τα μοντέλα context του κωδικοποιητή εντροπίας σε κάθε γραμμή συμπεραίνονται από αυτά στην προηγούμενη γραμμή με μια καθυστέρηση επεξεργασίας 2 CTU. Το WPP παρέχει μια μορφή παράλληλης επεξεργασίας σε ένα λεπτό επίπεδο granularity π.χ. μέσα σε ένα slice. Το WPP συχνά προσφέρει καλύτερη απόδοση συμπίεσης από τα tiles (και αποφεύγει κάποια artifacts που μπορεί να προκληθούν από τα tiles).
3. **Dependent slice segments.** Μια δομή που αποκαλείται dependent slice segment επιτρέπει τα δεδομένα συσχετισμένα με ένα συγκεκριμένο σημείο εισόδου wavefront ή tile να μεταφερθεί σε ένα διαφορετικό NAL unit, οπότε να μπορεί πιθανά να κάνει τα δεδομένα διαθέσιμα στο ένα σύστημα για fragmented packetization με μικρότερη καθυστέρηση από το αν κωδικοποιούνταν όλο μαζί σε ένα slice. Το dependant slice segment για ένα wavefront σημείο εισόδου μπορεί να αποκωδικοποιηθεί αφού έχει ήδη ολοκληρωθεί η αποκωδικοποίηση ενός άλλου slice segment. Τα dependent slice segments είναι κυρίως χρήσιμα στη low-delay κωδικοποίηση, όπου τα παράλληλα εργαλεία μπορεί να μειώσουν τον απόδοση συμπίεσης.

2.2. Σύνταξη υψηλού επιπέδου

Η σύνταξη υψηλού επιπέδου του HEVC περιέχει πολλά στοιχεία που έχουν κληρονομηθεί από το NAL του H.264/MPEG-4 AVC. Το NAL παρέχει την ικανότητα να χαρτογραφεί τα δεδομένα του video coding layer (VCL) που αντιπροσωπεύουν το υλικό των εικόνων πάνω σε διάφορα επίπεδα μετάδοσης, συμπεριλαμβάνοντας τα RTP/IP, ISO MP4, and H.222.0/MPEG-2 συστήματα, και παρέχει ένα framework με αντοχή στην απώλεια δεδομένων. Τα NAL κατηγοριοποιούνται σε VCL και non-VCL, ανάλογα με το εάν περιέχουν κωδικοποιημένες εικόνες ή άλλα συσχετισμένα δεδομένα αντίστοιχα. Στο HEVC πρότυπο, περιέχονται πολλοί VCL NAL τύποι που περιγράφουν κατηγορίες εικόνων για την αρχικοποίηση του αποκωδικοποιητή και για το random access.

A. Χαρακτηριστικά Random access και Bitstream Splicing.

Ο καινούργιος σχεδιασμός υποστηρίζει ειδικά χαρακτηριστικά που επιτρέπουν το random access και το bitstream splicing. Στο H.264/MPEG-4 AVC, το bitstream πρέπει πάντα να ξεκινάει με ένα IDR access unit. Ένα IDR access unit περιέχει μια ανεξάρτητα κωδικοποιημένη εικόνα, δηλαδή μια εικόνα που μπορεί να αποκωδικοποιηθεί χωρίς να έχουν αποκωδικοποιηθεί άλλες εικόνες στο NAL stream. Η ύπαρξη ενός IDR access unit υποδεικνύει πως καμία επακόλουθη εικόνα στο bitstream δε θα χρειαστεί αναφορά σε εικόνες πριν από την εικόνα που περιέχει για την αποκωδικοποίηση. Η IDR εικόνα χρησιμοποιείται μέσα σε μια δομή κωδικοποίησης γνωστή ως closed GOP (Group Of Pictures). Η καινούργια clean random access (CRA) σύνταξη ορίζει τη χρήση μιας ανεξάρτητα κωδικοποιημένης εικόνας στη θέση ενός random access point (RAP), δηλαδή μια θέση στο bitstream από την οποία ο αποκωδικοποιητής μπορεί επιτυχώς να αρχίσει να αποκωδικοποιεί εικόνες χωρίς την ανάγκη αποκωδικοποίησης εικόνων που έχουν εμφανιστεί νωρίτερα στο bitstream, το οποίο επιτρέπει αποδοτική χρονική σειρά κωδικοποίησης γνωστή ως open GOP operation. Η καλή υποστήριξη του random access είναι σημαντική για την παροχή των δυνατοτήτων αλλαγής καναλιού, λειτουργίας αναζήτησης και streaming υπηρεσιών. Μερικές εικόνες που ακολουθούν μια CRA εικόνα σε σειρά αποκωδικοποίησης και προηγούνται αυτής σε σειρά επίδειξης, μπορεί να περιέχουν inter πρόβλεψης αναφορές σε εικόνες που δεν είναι διαθέσιμες στον αποκωδικοποιητή. Αυτές οι μη αποκωδικοποιήσιμες εικόνες πρέπει συνεπώς να απορριφθούν από τον αποκωδικοποιητή που ξεκινάει τη διαδικασία αποκωδικοποίησης σε ένα CRA σημείο. Για αυτόν το λόγο, τέτοιες μη αποκωδικοποιήσιμες εικόνες αναγνωρίζονται ως random access skipped leading (RASL) εικόνες. Η θέση των σημείων ένωσης από τα διαφορετικά αρχικά κωδικοποιημένα bitstreams, μπορεί να υποδειχθεί από broken link access (BLA) εικόνες. Μια διαδικασία ένωσης bitstream μπορεί να πραγματοποιηθεί απλά αλλάζοντας τον τύπο NAL της CRA εικόνας στο ένα bitstream στην τιμή που υποδεικνύει μια BLA εικόνα και ενώνοντας το καινούργιο bitstream στη θέση μιας RAP εικόνας στο άλλο bitstream. Μια RAP εικόνα μπορεί να είναι IDR, CRA, η BLA, και οι CRA και BLA μπορούν να ακολουθούνται από RASL εικόνες στο bitstream (ανάλογα με τη συγκεκριμένη τιμή του τύπου NAL που χρησιμοποιήθηκε για την BLA εικόνα). Οποιαδήποτε RASL εικόνα συσχετισμένη με μια BLA εικόνα πρέπει πάντα να απορρίπτεται από τον αποκωδικοποιητή, αφού μπορεί να περιέχει αναφορές σε εικόνες που δεν είναι παρούσες στο bitstream λόγω της διαδικασίας ένωσης. Ο άλλος τύπος εικόνας που μπορεί να ακολουθεί μια RAP εικόνα σε σειρά αποκωδικοποίησης και να προηγείται σε σειρά επίδειξης

είναι μια random access decodable leading (RADL) εικόνα, η οποία δε μπορεί να περιέχει αναφορές σε καμία εικόνα που προηγείται της RAP εικόνας σε σειρά αποκωδικοποίησης. Στις RASL και RADL εικόνες αναφερόμαστε συλλογικά ως leading pictures (LPs). Εικόνες που ακολουθούν μια RAP εικόνα σε σειρά αποκωδικοποίησης και επίδειξης, που είναι γνωστές ως trailing pictures, δε μπορούν να περιέχουν αναφορές σε LPs για inter πρόβλεψη.

B. Υποστήριξη χρονικής υποστρωμάτωσης

Παρόμοια με τη λειτουργία χρονικής κλιμάκωσης στην H.264/MPEG-4 AVC κλιμακωτής κωδικοποίησης βίντεο επέκτασης [47], το HEVC ορίζει έναν χρονικό αναγνωριστεί στο NAL unit header, το οποίο υποδεικνύει ένα επίπεδο στη δομή ιεραρχικής χρονικής πρόβλεψης. Αυτό εισήχθη για να επιτευχθεί η χρονική κλιμάκωση χωρίς την ανάγκη προσπέλασης άλλων κομματιών του bitstream εκτός του NAL unit header.

Υπό συγκεκριμένες συνθήκες, ο αριθμός αποκωδικοποιημένων χρονικών υποστρωμάτων μπορεί να ρυθμιστεί κατά τη διάρκεια της διαδικασίας αποκωδικοποίησης μιας κωδικοποιημένης ακολουθίας βίντεο. Η θέση ενός σημείου στο bitstream στο οποίο η αλλαγή υποστρώματος είναι δυνατή να ξεκινήσει την αποκωδικοποίηση υψηλότερων χρονικών στρωμάτων μπορεί να υποδειχθεί από την ύπαρξη temporal sublayer access (TSA) εικόνων και stepwise TSA (STSA) εικόνων. Στη θέση μιας TSA εικόνας, είναι δυνατή η αλλαγή αποκωδικοποίησης από ένα χαμηλότερο χρονικό υπόστρωμα σε ένα οποιοδήποτε υψηλότερο χρονικό υπόστρωμα, και στη θέση μιας STSA εικόνας, είναι δυνατή η αλλαγή από ένα χαμηλότερο χρονικό υπόστρωμα σε ένα συγκεκριμένο υψηλότερο υπόστρωμα (αλλά όχι τα πολύ μακρυνα στρώματα πάνω από αυτό, εκτός αν περιέχουν TSA ή STSA εικόνες).

C. Επιπρόσθετα Parameter Sets

Το VPS έχει προστεθεί ως μεταδεδομένο για να περιγράψει τα συνολικά χαρακτηριστικά των κωδικοποιημένων ακολουθιών video, συμπεριλαμβανομένων και των εξαρτήσεων μεταξύ χρονικών υποστρωμάτων. Ο κύριος σκοπός αυτού, είναι καταστήσει ικανή τη συμβατή επεκτασιμότητα του προτύπου σε ότι έχει να κάνει με την ανταλλαγή σημάτων στο επίπεδο συστημάτων.

D. Reference Picture Sets και Reference Picture Lists

Για διαχείριση εικόνων πολλαπλών αναφορών, ένα συγκεκριμένο σύνολο αποκωδικοποιημένων εικόνων χρειάζεται να είναι παρών στον decoded picture buffer, (DPB) για την αποκωδικοποίηση των υπολοίπων εικόνων στο bitstream. Για να αναγνωριστούν αυτές οι εικόνες, μια λίστα από picture order count (POC) αναγνωριστικών μεταδίδεται σε κάθε slice header. Το σύνολο των διατηρημένων εικόνων αναφοράς καλείται reference picture set (RPS).

Όπως στο H.264/MPEG-4 AVC, υπάρχουν δυο λίστες που δημιουργούνται ως λίστες εικόνων του decoded picture buffer (DPB), και καλούνται λίστες αναφοράς 0 και 1. Ένας δείκτης ο οποίος καλείται δείκτης εικόνας αναφοράς χρησιμοποιείται για να για να δεικτοδοτήσει μια συγκεκριμένη εικόνα σε μια από αυτές τις λίστες. Για πρόβλεψη μονής κατεύθυνσης, μια εικόνα μπορεί να επιλεγεί από οποιαδήποτε από τις 2 λίστες. Για πρόβλεψη διπλής κατεύθυνσης

επιλέγονται 2 εικόνες, μια από κάθε λίστα. Όταν η λίστα περιέχει μόνο μια εικόνα, ο δείκτης εικόνας αναφοράς έχει την τιμή 0 και δε χρειάζεται να μεταδοθεί στο bitstream.

Η σύνταξη υψηλού επιπέδου για την αναγνώριση του RPS και τη δημιουργία των λιστών εικόνων αναφοράς για inter πρόβλεψη είναι πιο ανθεκτική σε απώλεια δεδομένων από τον προηγούμενο σχεδιασμό στο H.264/MPEG-4 AVC, και είναι πιο σύμφωνη με λειτουργίες όπως το random access και το trick mode operation, π.χ. fast-forward, smooth rewind, seeking και adaptive bitstream switching). Ένα χαρακτηριστικό κλειδί αυτής της βελτιστοποίησης είναι πως η σύνταξη είναι πιο ξεκάθαρη, σε σχέση με το να βασίζεται σε συμπεράσματα από την αποθηκευμένη κατάσταση της διαδικασίας αποκωδικοποίησης όπως αποκωδικοποιείται η μια εικόνα με τα την άλλη. Επιπλέον, η συσχετισμένη σύνταξη για αυτές τις πλευρές του σχεδιασμού είναι στην πραγματικότητα πιο απλή από ότι ήταν στο H.264/MPEG-4 AVC.

2.3. HEVC Video τεχνικές κωδικοποίησης

Όπως και σε όλα τα προηγούμενα ITU-T και ISO/IEC JTC 1 πρότυπα κωδικοποίησης video από το H.261 [42], ο σχεδιασμός του HEVC ακολουθεί την κλασική block-based υβριδική προσέγγιση κωδικοποίησης. Ο βασικός αλγόριθμος κωδικοποίησης είναι ένα υβρίδιο inter πρόβλεψης για να εκμεταλλεύεται τις χρονικές στατιστικές εξαρτήσεις, intra πρόβλεψης για να εκμεταλλευτεί τις χωρικές στατιστικές εξάρτησης και κωδικοποίησης μετασχηματισμού των προβλεφθέντων residual σημάτων για την περαιτέρω εκμετάλλευση των χωρικών στατιστικών εξαρτήσεων. Δεν υπάρχει κανένα στοιχείο κωδικοποίησης στο HEVC που προσφέρει την πλειοψηφία της σημαντικής βελτιστοποίησης στην απόδοση συμπίεσης σε σχέση με τα προηγούμενα πρότυπα κωδικοποίησης. Είναι ένα σύνολο μικρότερων βελτιστοποιήσεων που αθροιστικά προσφέρουν σημαντικό κέρδος.

A. Sampled Representation of Pictures

Για την αναπαραγωγή σήματος έγχρωμου video, το HEVC συνήθως χρησιμοποιεί tristimulus YCbCr χρωματικό χώρο με 4:2:0 δειγματοληψία. Αυτό διαχωρίζει την αναπαράσταση του χρώματος σε 3 δομικά στοιχεία ονόματι Y, Cb, Cr. Το Y στοιχεία αποκαλείται επίσης luma, και αναπαριστά τη φωτεινότητα. Τα 2 chroma στοιχεία Cb και Cr αναπαριστούν το βαθμό στον οποίο το chroma παρεκκλίνει προς το μπλε και το κόκκινο αντίστοιχα. Επειδή το σύστημα της ανθρώπινης όρασης είναι πιο ευαίσθητο στο luma από το chroma, χρησιμοποιείται συνήθως η 4:2:0 δομή δειγματοληψίας, στην οποία κάθε chroma στοιχείο έχει ένα τέταρτο του αριθμού των δειγμάτων από το luma στοιχείο (μισό αριθμό δειγμάτων σε οριζόντια και κάθετη διάσταση). Κάθε δείγμα για κάθε στοιχείο αναπαρίσταται με 8 ή 10 bits ακριβείας, με την περίπτωση των 8 bits να είναι πιο συχνή.

Οι εικόνες του video συνήθως δειγματοληπτούνται με progressive τρόπο με τετράγωνα μεγέθη εικόνας $W \times H$, όπου W το πλάτος και H το ύψος της εικόνας σε όρους luma δειγμάτων. Κάθε πίνακας chroma δειγμάτων, με 4:2:0 δειγματοληψία, σπάει στη συνέχεια σε μεγέθη $W/2 \times H/2$. Με τέτοιο σήμα video, η σύνταξη του HEVC διαχωρίζει περαιτέρω τις εικόνες ως εξής.

B. Διαίρεση της εικόνας σε Coding Tree Units.

Μια εικόνα διαχωρίζεται σε coding tree units, (CTUs) τα οποία περιέχουν το κάθε ένα luma CTBs και chroma CTBs. Ένα luma CTB καλύπτει μια ορθογώνια περιοχή από $L \times L$ δείγματα του luma στοιχείου και τα αντίστοιχα chroma CTBs καλύπτουν κάθε $L/2 \times L/2$ δείγματα από κάθε chroma στοιχείο. Η τιμή του L μπορεί να είναι 16, 32, ή 64 όπως οριστεί δηλαδή από το κωδικοποιημένο στοιχείο σύνταξης που ορίζεται στο SPS. Σε σύγκριση με τα παραδοσιακά macroblocks που χρησιμοποιούν ένα προκαθορισμένο μέγεθος πίνακα 16×16 από luma δείγματα, όπως χρησιμοποιείται από τα προηγούμενα ITU-T και ISO/IEC JTC 1 πρότυπα κωδικοποίησης video από το H.261, το HEVC υποστηρίζει μεταβλητού μεγέθους CTBs που επιλέγονται ανάλογα με τις ανάγκες του κωδικοποιητή σε ότι έχει να κάνει με απαιτήσεις υπολογιστικής ισχύος και μνήμης. Η υποστήριξη μεγαλύτερων CTBs από τα προηγούμενα πρότυπα είναι ιδιαίτερα ωφέλιμη όταν κωδικοποιείται video υψηλής ανάλυσης. Τα luma CTB και τα 2 chroma CTBs μαζί με τη συσχετισμένη σύνταξη συγκροτούν ένα CTU. Το CTU είναι η βασική μονάδα επεξεργασίας που χρησιμοποιείται στο πρότυπο για τον ορισμό της διαδικασίας αποκωδικοποίησης.

C. Διαχωρισμός του CTB σε CBs

Τα blocks που ορίζονται ως luma και chroma CTBs μπορούν αμέσως να χρησιμοποιηθούν ως CBs ή μπορούν να διαχωριστούν περαιτέρω σε πολλαπλά CBs. Ο διαχωρισμός επιτυγχάνεται χρησιμοποιώντας δομές δέντρων. Ο διαχωρισμός δέντρων στο HEVC εφαρμόζεται ταυτόχρονα στα luma και chroma στοιχεία, όμως εξαιρέσεις εφαρμόζονται όταν τα ελάχιστα μεγέθη φτάνονται για τα chroma στοιχεία.

Το CTU περιέχει σύνταξη quadtree που επιτρέπει τον διαχωρισμό των CBs σε ένα επιλεγμένο και κατάλληλο μέγεθος βασισμένο στα χαρακτηριστικά του σήματος της περιοχής που καλύπτεται από το CTB. Η διαδικασία διαχωρισμού του CTB μπορεί να επαναληφθεί μέχρι το μέγεθος για τα luma CB φτάσει στο μικρότερο επιτρεπτό μέγεθος που επιλέγεται από τον κωδικοποιητή χρησιμοποιώντας τη σύνταξη στο SPS και είναι πάντα 8×8 ή μεγαλύτερο (σε μονάδες luma δειγμάτων). Τα όρια των εικόνων ορίζονται σε μονάδες του ελαχίστου επιτρεπτού luma CB μεγέθους. Ως αποτέλεσμα, στις δεξιά και κάτω άκρες της εικόνας, μερικά CTUs μπορεί να καλύπτουν περιοχές που είναι μερικώς εκτός των ορίων της εικόνας. Αυτή η κατάσταση αντιμετωπίζεται από τον αποκωδικοποιητή, και το CTU quadtree εννοούμενα χωρίζεται όπως πρέπει για να μειωθεί το μέγεθος του CB στο σημείο που θα μπορεί να χωρέσει ολόκληρο μέσα στην εικόνα.

D. PBs και PUs

Ο τρόπος πρόβλεψης για τα CUs σηματοδοτείται ως intra ή inter, ανάλογα με το εάν χρησιμοποιείται intra (χωρική) ή inter (χρονική) πρόβλεψη.

Όταν ο τρόπος πρόβλεψης σηματοδοτείται ως intra, το μέγεθος του PB το οποίο είναι το μέγεθος του block στο οποίο θα εφαρμοστεί intra πρόβλεψη είναι το ίδιο με το μέγεθος CB για όλα τα μεγέθη των block εκτός από το μικρότερο μέγεθος CB που επιτρέπεται στο bitstream. Για τη δεύτερη περίπτωση, ένα flag ενεργοποιείται που υποδεικνύει εάν το CB θα χωριστεί σε 4 PB quadrants όπου το κάθε ένα θα έχει το δικό του intra τρόπο πρόβλεψης. Ο λόγος που επιτρέπεται να γίνει αυτός ο διαχωρισμός είναι για να γίνουν δυνατές οι διακριτές επιλογές intra πρόβλεψης για blocks μέχρι 4×4 σε μέγεθος. Όταν η intra πρόβλεψη λειτουργεί με block μεγέθους 4×4 , η intra

πρόβλεψη για τα chroma blocks χρησιμοποιεί επίσης 4x4 blocks (με το κάθε ένα να καλύπτει την ίδια περιοχή όπως τα τέσσερα 4x4 luma blocks). Το πραγματικό μέγεθος της περιοχής στην οποία λειτουργεί η intra πρόβλεψη (η οποία είναι διακριτή από το μέγεθος του PB, στο οποίο εφαρμόζεται intra πρόβλεψη) βασίζεται στο residual coding partitioning που περιγράφεται παρακάτω.

Όταν ο τρόπος πρόβλεψης σηματοδοτείται ως inter, ορίζεται εάν τα luma και chroma CBs χωρίζονται σε ένα, δυο, ή 4 PBs. Ο διαχωρισμένος σε 4 PBs επιτρέπεται μόνο όταν το μέγεθος CB είναι ίσο με το ελάχιστο επιτρεπτό μέγεθος CB, χρησιμοποιώντας έναν αντίστοιχο τρόπο διαχωρισμού όπως θα μπορούσε αλλιώς να εφαρμοστεί στο επίπεδο CB του σχεδιασμού αντί στο PB επίπεδο. Όταν ένα CB χωρίζεται σε 4 PBs, κάθε PB καλύπτει ένα quadrant του CB. Όταν ένα CB χωρίζεται σε 2 PBs, 6 τύπου αυτού του διαχωρισμού είναι πιθανοί. Οι συνδυασμοί είναι, μη διαχωρισμός του $M \times M$ CB, διαχωρισμός του CB σε 2 PBs μεγέθους $M \times M/2$ ή $M/2 \times M$ ή διαχωρισμός σε 4 PBs μεγέθους $M/2 \times M/2$. Οι επόμενοι συνδυασμοί ονομάζονται asymmetric motion partitioning (AMP), και επιτρέπονται μόνο όταν το M είναι 16 ή μεγαλύτερο για τα luma στοιχεία. Ένα PB ασυμμετρικού διαχωρισμού έχει ύψος ή πλάτος $M/4$ και ύψος ή πλάτος M αντίστοιχα, και το υπόλοιπο PB γεμίζει το εναπομείναν CB έχοντας ύψος ή πλάτος $3M/4$ και ύψος ή πλάτος M . Σε κάθε inter προβλεφθέν PB ανατίθενται 1 ή 2 διανύσματα κίνησης και δείκτες εικόνων αναφοράς. Για την ελαχιστοποίηση της χειρότερης περίπτωσης σε θέμα εύρους ζώνης μνήμης, τα luma PBs μεγέθους 4x4 δεν επιτρέπεται να περάσουν inter πρόβλεψη, και τα luma PBs μεγέθους 4x8 και 8x4 δεν επιτρέπεται να περάσουν κωδικοποίηση πρόβλεψης μονής κατεύθυνσης.

E. Διαχωρισμός δομής δέντρου σε block και μονάδες μετασχηματισμού.

Για το residual coding, ένα CB μπορεί να διαχωριστεί αναδρομικά σε blocks μετασχηματισμού (TBs). Ο διαχωρισμός σηματοδοτείται από ένα residual quadtree.

Μόνο τετράγωνος CB και TB διαχωρισμός ορίζεται, όπου ένα block μπορεί να χωριστεί αναδρομικά σε quadrants. Για ένα luma CB μεγέθους $M \times M$, ένα flag σηματοδοτεί εάν χωρίζεται σε 4 blocks μεγέθους $M/2 \times M/2$. Εάν περαιτέρω διαχωρισμός είναι δυνατός, όπως σηματοδοτείται από ένα μέγιστο βάθος του residual quadtree που υποδεικνύεται στο SPS, σε κάθε quadrant ανατίθεται ένα flag που ορίζει εάν αυτό θα χωριστεί σε 4 άλλα quadrants. Τα blocks τα οποία είναι κόμβοι φύλλα που προέρχονται από το residual quadtree είναι τα blocks μετασχηματισμού τα οποία επεξεργάζονται περαιτέρω από την κωδικοποίηση μετασχηματισμού. Ο κωδικοποιητής υποδεικνύει το μέγιστο και το ελάχιστο luma TB μέγεθος που θα χρησιμοποιήσει. Ο διαχωρισμός είναι εννοούμενος όταν το μέγεθος του CB είναι μεγαλύτερο από το μέγιστο μέγεθος του TB. Ο μη διαχωρισμός είναι και αυτός εννοούμενος όταν το μέγεθος του luma TB που θα προέκυπτε θα ήταν μικρότερο από το ελάχιστο. Το μέγεθος του chroma TB είναι μισό του luma TB σε κάθε διάσταση, εκτός όταν το luma TB είναι 4x4, στη οποία περίπτωση ένα μοναδικό 4x4 chroma TB χρησιμοποιείται για την περιοχή που καλύπτεται από 4 4x4 luma TBs. Στην περίπτωση intra προβλεφθέν CUs, τα αποκωδικοποιημένα δείγματα των κοντινότερων γειτονικών TBs χρησιμοποιούνται ως δεδομένα αναφοράς για intra πρόβλεψη. Σε αντίθεση με τα προηγούμενα

πρότυπα, το HEVC επιτρέπει Ένα TB να απλώνεται πάνω από πολλά PBs, έτσι ώστε τα inter προβλεφθέντα CUs να μεγιστοποιούν την πιθανή οφέλη απόδοσης κωδικοποίησης της quadtree δομής του TB διαχωρισμού.

F. Slices και Tiles

Τα slices είναι μια ακολουθία από CTUs που επεξεργάζονται με σειρά raster scan. Μια εικόνα μπορεί να χωριστεί σε ένα ή πολλά slices ώστε μια εικόνα να είναι ένα σύνολο ενός ή περισσοτέρων slices. Τα slices είναι αυτόνομα, δηλαδή, εάν είναι διαθέσιμα τα ενεργά picture parameter sets, τα συντακτικά στοιχεία τους μπορούν να προσπελαστούν από το bitstream και οι τιμές των δειγμάτων στη συγκεκριμένη περιοχή της εικόνας μπορούν να αποκωδικοποιηθούν σωστά χωρίς τη χρήση δεδομένων από κάποιο άλλο slice στην ίδια εικόνα. Αυτό σημαίνει πως πρόβλεψη μέσα από την ίδια εικόνα δεν εφαρμόζεται πέρα από τα όρια των slices. Κάποιες πληροφορίες βέβαια μπορεί να χρειαστούν για να εφαρμοστεί in-loop φιλτράρισμα στις οριακές περιοχές. Κάθε slice μπορεί να κωδικοποιηθεί χρησιμοποιώντας διαφορετικούς τύπους κωδικοποίησης ως εξής.

- 1) H slice: Ένα slice στο οποίο όλα τα CUs του slice κωδικοποιούνται χρησιμοποιώντας μόνο intra πρόβλεψη.
- 2) P slice: Επιπροσθέτως στους τύπους κωδικοποίησης ενός H slice, μερικά από τα CUs ενός P slice μπορεί επίσης να κωδικοποιηθεί χρησιμοποιώντας inter πρόβλεψη με το πολύ ένα motion-compensated σήμα πρόβλεψης ανά PB. Τα P slices χρησιμοποιούν μόνο ένα reference picture list, το 0.
- 3) B slice: Επιπροσθέτως στους τύπους κωδικοποίησης ενός P slice, μερικά από τα CUs ενός B slice μπορούν επίσης να κωδικοποιηθούν χρησιμοποιώντας inter πρόβλεψη με το πολύ 2 motion-compensated σήματα πρόβλεψης ανά PB. Τα B slices χρησιμοποιούν και τα 2 reference picture list, 0 και 1.

Ο κύριος σκοπός των slices είναι ο επανασυγχρονισμός μετά από απώλεια δεδομένων. Επιπλέον, στα slices δεν επιτρέπεται η χρήση του μεγίστου αριθμού από bits. Συνεπώς, τα slices μπορεί συχνά να περιέχουν μεγάλο βαθμό ποικίλων CUs ανά slice με τρόπο που εξαρτάται από τις δραστηριότητες στην σκηνή του video. Μαζί με τα slices, το HEVC επίσης ορίζει και τα tiles, τα οποία είναι αυτόνομα και ανεξαρτήτως αποκωδικοποιήσιμες ορθογώνιες περιοχές στην εικόνα. Ο κύριος σκοπός των tiles είναι να καταστήσουν δυνατές τις αρχιτεκτονικές παράλληλης επεξεργασίας για κωδικοποίηση και αποκωδικοποίηση. Πολλά tiles μπορούν να μοιράζονται τις header πληροφορίες που περιέχει ένα slice. Αλλιώς, ένα tile μπορεί να περιέχει πολλαπλά slices. ένα tile αποτελείται από ορθογώνιες ομάδες από CUs. Επιπλέον, για να βοηθηθεί το granularity της πακετοποίησης των δεδομένων, ορίζονται ανεξάρτητα slices. Τέλος, με το WPP, ένα slice διαιρείται σε γραμμές από CUs. Η αποκωδικοποίηση μπορεί να ξεκινήσει μόλις παρθούν λίγες αποφάσεις που χρειάζονται για την πρόβλεψη και την προσαρμογή του κωδικοποιητή εντροπίας στην προηγούμενη σειρά. αυτό υποστηρίζει παράλληλη επεξεργασία των γραμμών των CUs χρησιμοποιώντας πολλά threads επεξεργασίας στον κωδικοποιητή, στον αποκωδικοποιητή ή και στα 2. Για λόγους απλότητας, το WPP δεν επιτρέπεται να χρησιμοποιηθεί σε συνδυασμό με τα tiles (ενώ αυτά τα χαρακτηριστικά θα μπορούσαν θεωρητικά να δουλέψουν μαζί).

G. Intra πρόβλεψη

Η intra πρόβλεψη λειτουργεί ανάλογα με το μέγεθος του TB, και τα προηγουμένως αποκωδικοποιημένα οριακά δείγματα από χωρικά γειτονικά TBs χρησιμοποιούνται για να σχηματίσουν το σήμα πρόβλεψης. Ορίζεται κατευθυντική πρόβλεψη με 33 διαφορετικές κατευθύνσεις για (τετράγωνα) TBs με μέγεθος από 4x4 μέχρι και 32x32. Για τα chroma δείγματα, οι οριζόντιοι, κάθετοι, planar και DC τρόποι μπορούν να σηματοδοτηθούν με ξεκάθαρο τρόπο η ο τρόπος πρόβλεψης chroma μπορεί να σηματοδοτηθεί ως ίδιος με των luma.

Κάθε CB μπορεί να κωδικοποιηθεί με έναν από τους πολλούς τρόπους, ανάλογα με τον τύπο slice. Παρόμοια με το H.264/MPEG-4 AVC, η κωδικοποίηση intra πρόβλεψης υποστηρίζεται από όλους τους τύπους slices. Το HEVC υποστηρίζει διάφορες μεθόδους κωδικοποίησης intra πρόβλεψης ονόματι Intra_Angular, Intra_Planar, and Intra_DC.

1) PB Partitioning

Ένα intra προβλεφθέν CB μεγέθους $M \times M$ μπορεί να έχει έναν από τους δυο τύπους PB διαχωρισμού ονόματι PART_2N×2N and PART_N×N από τα οποία το πρώτο υποδεικνύει πως το CB δε χωρίζεται και το δεύτερο πως το CB χωρίζεται σε 4 PBs ίσου μεγέθους. Όμως είναι δυνατή η αναπαράσταση των ίδιων περιοχών που θα ορίζονταν από 4 PBs χρησιμοποιώντας 4 μικρότερα CBs όταν το μέγεθος του τρέχοντος CB είναι μεγαλύτερο από το ελάχιστο μέγεθος CU. Οπότε, ο σχεδιασμός του HEVC επιτρέπει μόνο διαχωρισμό τύπου PART_N×N προς χρήση όταν το μέγεθος του τρέχοντος CB είναι ίσο με το ελάχιστο μέγεθος CU. Παρόλο που ο τρόπος intra πρόβλεψης εφαρμόζεται στο επίπεδο PB, η πραγματική διαδικασία πρόβλεψης εφαρμόζεται ξεχωριστά για κάθε TB.

2) Intra_Angular

PredictionH χωρική intra πρόβλεψη έχει χρησιμοποιηθεί επιτυχώς προηγουμένως στο H.264/MPEG-4 AVC. Η intra πρόβλεψη στο HEVC λειτουργεί παρόμοια στο χώρο, αλλά έχει επεκταθεί σημαντικά, κυρίως λόγω του αυξημένου μεγέθους των TB και του αυξημένου πλήθους των κατευθύνσεων πρόβλεψης. Συγκριτικά με τις 8 κατευθύνσεις πρόβλεψης του H.264/MPEG-4 AVC, το HEVC υποστηρίζει στο σύνολο 33 κατευθύνσεις πρόβλεψης, ονόματι Intra_Angular[k], όπου το k είναι ο αριθμός τύπου από το 2 μέχρι και το 34. Οι γωνιές σχεδιάστηκαν έτσι επίτηδες για να παρέχουν πιο πυκνή κάλυψη για σχεδόν οριζόντιες και σχεδόν κάθετες γωνιές και πιο coarse κάλυψη για σχεδόν διαγώνιες κατευθύνσεις για να αναπαραστήσουν τη συχνότητα των γωνιών και την αποτελεσματικότητα της διαδικασίας πρόβλεψης σήματος. Όταν χρησιμοποιείται ο τρόπος Intra_Angular, κάθε TB προβλέπεται κατευθυντικά από χωρικά γειτονικά δείγματα τα οποία είναι ανακατασκευασμένα (αλλά όχι ακόμα φιλτραρισμένα από τα in-loop φίλτρα) πρώτου χρησιμοποιηθούν για αυτήν την πρόβλεψη. Για ένα TB μεγέθους $N \times N$, ένα σύνολο από $4N+1$ χωρικά γειτονικά δείγματα θα πρέπει να χρησιμοποιηθεί για την πρόβλεψη. Όταν είναι διαθέσιμα από την προηγούμενη διαδικασία αποκωδικοποίησης, τα δείγματα από τα κάτω αριστερά TBs μπορούν να χρησιμοποιηθούν για την πρόβλεψη μαζί με δείγματα από TBs από αριστερά, πάνω και πάνω δεξιά από το τρέχον TB.

Η διαδικασία πρόβλεψης των Intra_Angular τρόπων μπορεί να περιέχει προεκβολή των δειγμάτων από μια projected reference sample περιοχή σύμφωνα με την δοθείσα

κατεύθυνση. Για να αφαιρεθεί η ανάγκη για δείγμα προς δείγμα αλλαγή μεταξύ γραμμής αναφοράς και column buffers, για το Intra_Angular[k] με $k \in \{2, 17\}$, τα δείγματα που βρίσκονται στην από πάνω γραμμή projected ως επιπλέον δείγματα που βρίσκονται στην αριστερή στήλη, και για $k \in \{18, 34\}$, τα δείγματα που βρίσκονται στην αριστερή στήλη προβαλλόμενα ως δείγματα που βρίσκονται στην από πάνω γραμμή.

Για να βελτιωθεί η ακρίβεια της intra πρόβλεψης, οι θέσεις των προβαλλόμενων δειγμάτων αναφοράς υπολογίζονται με $1/32$ ακρίβεια δείγματος. Bilinear παρεμβολή χρησιμοποιείται για την απόκτηση της τιμής των projected reference δειγμάτων χρησιμοποιώντας τα 2 πιο κοντινά δείγματα αναφοράς που βρίσκονται σε ακέραιες θέσεις. Η διαδικασία πρόβλεψης των Intra_Angular τρόπων είναι συνεχής σε όλα τα μεγέθη από blocks και σε όλες τις κατευθύνσεις πρόβλεψης, όπου το H.264/MPEG-4 AVC χρησιμοποιεί διαφορετικές μεθόδους για τα υποστηριζόμενα μεγέθη block 4x4, 8x8, 16x16. Αυτή η συνέχεια του σχεδιασμού είναι ιδιαίτερα επιθυμητή από τη στιγμή που το HEVC υποστηρίζει μια μεγαλύτερη ποικιλία από TB μεγέθη και ένα σημαντικά μεγαλύτερο πλήθος από κατευθύνσεις πρόβλεψης σε σύγκριση με το H.264/MPEG-4 AVC.

3) Intra_Planar και Intra_DC πρόβλεψη

Μαζί με την Intra_Angular πρόβλεψη που στοχεύει περιοχές με έντονες κατευθυντικές γωνίες, το HEVC υποστηρίζει ακόμα 2 εναλλακτικές μεθόδους πρόβλεψης, Intra_Planar και Intra_DC, για τις οποίες υπήρχαν παρόμοιοι τρόποι στο H.264/MPEG-4 AVC. Ενώ η Intra_DC μέθοδος χρησιμοποιεί τον μέσο όρο των αναφοράς τιμών για μια πρόβλεψη, η Intra_Planar μέθοδος παίρνει τη μέση τιμή 2 γραμμικών προβλέψεων χρησιμοποιώντας 4 γωνιακά δείγματα αναφοράς για την αποφυγή ασυνεχειών στα όρια των blocks. Η Intra_Planar πρόβλεψη υποστηρίζεται για όλα τα μεγέθη των block στο HEVC ενώ το H.264/MPEG-4 AVC υποστηρίζει μόνο plane πρόβλεψη όταν το μέγεθος του luma PB είναι 16x16 και η plane πρόβλεψη λειτουργεί διαφορετικά από την planar πρόβλεψη του HEVC.

4) Reference sample smoothing

Στο HEVC, τα δείγματα αναφοράς που χρησιμοποιούνται για την intra πρόβλεψη συνήθως φιλτράρονται από ένα φίλτρο εξομάλυνσης τριών εισόδων $[1 \ 2 \ 1]/4$ με τρόπο παρόμοιο με αυτόν που χρησιμοποιείται για την 8x8 intra πρόβλεψη στο H.264/MPEG-4 AVC. Το HEVC εφαρμόζει τις διαδικασίες εξομάλυνσης πιο προσαρμοστικά, ανάλογα με την κατεύθυνση, το ποσοστό της ασυνέχειας, και το μέγεθος του block. Όπως στο H.264/MPEG-4 AVC, το φίλτρο εξομάλυνσης δεν εφαρμόζεται σε 4x4 blocks. Για τα 8x8 blocks, μόνο οι διαγώνιες κατευθύνσεις, Intra_Angular[k] για $k \in \{2, 18, 34\}$ χρησιμοποιούν την εξομάλυνση δειγμάτων αναφοράς. Για τα 16x16 blocks, δείγματα αναφοράς φιλτράρονται στις περισσότερες κατευθύνσεις εκτός από τις σχεδόν οριζόντιες και σχεδόν κάθετες διευθύνσεις, με k στο πεδίο 9-11 και 25-27. Για τα 32x32 blocks, όλες οι κατευθύνσεις εκτός από την οριζόντια με $k=10$ και την κάθετη με $k=26$ χρησιμοποιούν το φίλτρο εξομάλυνσης και όταν το ποσοστό ασυνέχειας ξεπερνάει ένα σημείο, εφαρμόζεται

bilinear παρεμβολή από 3 γειτονικές περιοχές δειγμάτων για να επιτευχθεί μια ομαλή πρόβλεψη.

Ο Intra_Planar τρόπος επίσης χρησιμοποιεί το φίλτρο εξομάλυνσης όταν το μέγεθος του block είναι μεγαλύτερο ή ίσο του 8x8, και η εξομάλυνση δε χρησιμοποιείται για την Intra_DC περίπτωση.

5) Boundary value smoothing

Για να αφαιρεθούν οι ασυνέχειες κατά το μήκος των οριων, σε 3 τρόπους, Intra_DC και Intra_Angular με $k=10$ ή 26 , τα οριακά δείγματα μέσα στο TB αντικαθίστανται από φίλτραρισμένα δείγματα όταν το μέγεθος του TB είναι μικρότερο από 32×32 . Για τον Intra_DC τρόπο, η πρώτη σειρά αλλά και στήλη του TB αντικαθίστανται από την έξοδο του $[3 \ 1]/4$ φίλτρου 2 εισόδων το οποίο παίρνει ως είσοδο την αρχική τιμή και το γειτονικό δείγμα αναφοράς. Στην οριζόντια πρόβλεψη, τα οριακά δείγματα της πρώτης στήλης του TB περνάνε από επεξεργασία ώστε μισή από τη διαφορά τους μεταξύ των γειτονικών δειγμάτων αναφοράς και των αριστερά πάνω δειγμάτων αναφοράς αθροίζεται. Αυτό κάνει το προβλεφθέν σήμα πολύ πιο ομαλό όταν υπάρχουν μεγάλες διακυμάνσεις κατά την κάθετη κατεύθυνση. Στην πρόβλεψη κάθετης κατεύθυνσης η ίδια τεχνική εφαρμόζεται στην πρώτη σειρά δειγμάτων.

6) Reference sample substitution

Τα γειτονικά δείγματα αναφοράς δεν είναι διαθέσιμα στα slice ή tile όρια. Επιπλέον, όταν το χαρακτηριστικό αντοχής σφαλμάτων ονόματι constrained intra prediction είναι ενεργοποιημένο, τα γειτονικά δείγματα αναφοράς μέσα σε οποιοδήποτε inter προβλεφθέν PB είναι και αυτά μη διαθέσιμα έτσι ώστε να αποφευχθεί η μετάδοση λαθών από πιθανά αλλοιωμένες προηγούμενες αποκωδικοποιημένες εικόνες στο σήμα πρόβλεψης. Ενώ στο H.264/MPEG-4 AVC, μόνο ο τρόπος πρόβλεψης Intra_DC επιτρέπεται σε τέτοιες περιπτώσεις, το HEVC επιτρέπει τη χρήση και άλλων τρόπων intra πρόβλεψης μετά από την αντικατάσταση των μη διαθέσιμων τιμών δειγμάτων αναφοράς με τις τιμές των γειτονικών δειγμάτων αναφοράς.

7) Mode coding

Το HEVC υποστηρίζει ένα σύνολο από τρόπους πρόβλεψης luma, 33 Intra_Angular, Intra_Planar και Intra_DC για όλα τα μεγέθη από blocks. Λόγω του αυξημένου πλήθους κατευθύνσεων, το HEVC υπολογίζει τους 3 πιο πιθανούς τρόπους (MPMs) όταν κωδικοποιεί τη luma intra πρόβλεψη προβλέψιμα, σε αντίθεση με τον μοναδικό πιο πιθανό τρόπο στο H.264/MPEG-4 AVC.

H. Inter Πρόβλεψη

1) PB partitioning

Σε σχέση με τα intra προβλεφθέντα CBs, το HEVC υποστηρίζει περισσότερα σχήματα διαχωρισμού για τα inter προβλεφθέντα CBs. Οι τρόποι διαχωρισμού των PART_2Nx2N, PART_Nx2N, and PART_2NxN υποδεικνύουν τις περιπτώσεις όταν το CB δε χωρίζεται, χωρίζεται σε 2 ίσα PBs οριζόντια και χωρίζεται σε 2 ίσα PBs κάθετα αντίστοιχα. Ο διαχωρισμός PART_NxN ορίζει πως το CB χωρίζεται σε 4 ίσου μεγέθους PBs, αλλά αυτός

ο τρόπος υποστηρίζεται μόνο όταν το μέγεθος του CB είναι ίσο με το μικρότερο επιτρεπτό μέγεθος. Επιπλέον, υπάρχουν άλλοι 4 τρόποι διαχωρισμού του CB σε 2 PBs έχοντας διαφορετικά μεγέθη: $PART-2N \times nU$, $PART-2N \times nD$, $PART-nL \times 2N$, και $PART-nR \times 2N$. Αυτοί οι τύποι είναι γνωστοί ως asymmetric motion partitions.

2) Fractional Sample Interpolation

Τα δείγματα ενός PB για ένα intra προβλεφθέν CB αποκτούνται από αυτά μιας αντίστοιχης block περιοχής στην εικόνα αναφοράς που αναγνωρίζεται από έναν δείκτη εικόνας αναφοράς, ο οποίος βρίσκεται σε μια θέση εκτοπισμένοι από τα οριζόντια και κάθετα στοιχεία του διανύσματος κίνησης. Εκτός από την περίπτωση όταν το διάνυσμα κίνησης έχει ακέραια τιμή, fractional sample interpolation χρησιμοποιείται για να δημιουργήσει τα δείγματα πρόβλεψης για μη ακέραιες θέσεις δειγματοληψίας. Όπως στο H.264/MPEG-4 AVC, το HEVC υποστηρίζει διανύσματα κίνησης με μονάδες ενός τετάρτου απόστασης μεταξύ των luma δειγμάτων. Για τα chroma δείγματα, η ακρίβεια των διανυσμάτων κίνησης αποφασίζεται σύμφωνα με το είδος της chroma δειγματοληψίας, το οποίο για 4:2:0 δειγματοληψία έχει ως αποτέλεσμα απόσταση έως όγδοου μεταξύ των chroma δειγμάτων.

Η προεκβολή των κλασματικών luma δειγμάτων στο HEVC χρησιμοποιεί διαχωριστεί εφαρμογή ενός 8 εισόδων φίλτρου για τις θέσεις των half-samples και ένα φίλτρο 7 εισόδων για τις quarter-sample θέσεις. Αντίθετα με το H.264/MPEG-4 AVC, όπου εφαρμόζεται μια διαδικασία παρεμβολής 2 επιπέδων, πρώτα παράγοντας τις τιμές ενός η 2 γειτονικών δειγμάτων με half-sample θέσεις χρησιμοποιώντας φίλτρο 6 εισόδων, στρογγυλοποιώντας τα ενδιάμεσα αποτελέσματα και μετά βρίσκοντας τον μέσο όρο των 2 τιμών σε ακέραια ή half-sample θέση. Το HEVC μόνο μια συνεχή διαδικασία παρεμβολής για την παραγωγή των κλασματικών θέσεων χωρίς ενδιάμεσες πράξεις στρογγυλοποίησης, η οποία βελτιώνει την ακρίβεια και απλοποιεί την αρχιτεκτονική της παρεμβολής κλασματικών δειγμάτων. Επίσης, η διαδικασία παρεμβολής στο HEVC βελτιώνεται χρησιμοποιώντας πιο μεγάλα φίλτρα, π.χ. 7 και 8 εισόδων σε σχέση με 6 εισόδων. Η χρήση μόνο φίλτρων 7 εισόδων για τον υπολογισμό των θέσεων των half-samples ήταν αρκετή για τις θέσεις quarter sample παρεμβολής από τη στιγμή που οι quarter-sample θέσεις είναι σχετικά κοντά με τις ακέραιες θέσεις, οπότε το πιο μακρινό δείγμα σε ένα φίλτρο παρεμβολής θα ήταν πιο μακριά από την περίπτωση half-sample. Οι τιμές των φίλτρων παρεμβολής προέρχονται εν μέρη από τις εξισώσεις του Διακριτού Μετασχηματισμού Συνημιτόνου.

Στο H.264/MPEG-4 AVC, μέχρι και 3 στάδια στρογγυλοποιήσεων απαιτούνται για την απόκτηση του δείγματος πρόβλεψης. Εάν χρησιμοποιείται πρόβλεψη διπλής κατεύθυνσης τότε ο συνολικός αριθμός στρογγυλοποιήσεων είναι 7 στη χειρότερη περίπτωση. Στο HEVC, το πολύ 2 στρογγυλοποιήσεις χρειάζονται για την απόκτηση των δειγμάτων στις quarter-sample θέσεις, έτσι 5 στρογγυλοποιήσεις είναι αρκετές για τη χειρότερη περίπτωση στην πρόβλεψη διπλής κατεύθυνσης. Επιπλέον στην πιο συχνή περίπτωση, όπου το bit depth B είναι 8 bits, ο συνολικός αριθμός στρογγυλοποιήσεων μειώνεται περαιτέρω σε 3. Λόγω του μικρότερου αριθμού στρογγυλοποιήσεων, το αθροιστικό

σφάλμα από αυτές μειώνεται και επιτρέπεται μεγαλύτερη ευελιξία σε σχέση με τις λειτουργίες που χρειάζεται να εκτελέσει ο αποκωδικοποιητής.

Η διαδικασία παρεμβολής κλασματικών δειγμάτων για τα chroma δείγματα είναι παρόμοια με αυτή για τα luma δείγματα, εκτός από τον αριθμό των εισόδων των φίλτρων που είναι 4 και η κλασματική ακρίβεια είναι 1/8 για την συνηθισμένη περίπτωση 4:2:0 chroma μορφής.

3) Merge mode

Οι πληροφορίες κίνησης συνήθως αποτελούνται από τις τιμές οριζόντιας και κάθετης μετατόπισης των διανυσμάτων κίνησης, έναν ή δυο δείκτες σε εικόνες αναφοράς και στην περίπτωση των περιοχών πρόβλεψης στα B slices, μια ταυτότητα της οποίας η λίστα εικόνων αναφοράς συσχετίζεται με κάθε δείκτη. Το HEVC περιέχει ένα merge mode που παράγει τις πληροφορίες κίνησης από χρονικά ή χωρικά γειτονικά blocks. Ονομάζεται merge mode αφού σχηματίζει μια κοινή περιοχή που μοιράζεται όλες τις πληροφορίες κίνησης. Το merge mode είναι παρόμοιο με το direct και skip mode στο H.264/MPEG-4 AVC. Όμως, υπάρχουν δυο σημαντικές διαφορές. Αρχικά, μεταδίδει πληροφορίες δεικτοδότησης για την επιλογή ενός από τους διαθέσιμους υποψηφίους, με έναν τρόπο που συνήθως ονομάζεται motion vector competition scheme. Επίσης αναγνωρίζει με σαφή τρόπο τη λίστα από εικόνες αναφοράς και τους δείκτες αυτών, όπου το direct mode παίρνει ως δεδομένο πως υπάρχουν ήδη τιμές. Το σύνολο των πιθανών υποψηφίων στο merge mode αποτελείται από χωρικά γειτονικούς υποψήφιους, έναν χρονικό υποψήφιο και παραγμένους υποψηφίους. Μετά την αξιολόγηση των χωρικών υποψηφίων, δυο είδη πλεονάζουσας πληροφορίας αφαιρούνται. Εάν η θέση του υποψήφιου για το τρέχον PU αναφερόταν στο πρώτο PU μέσα στο ίδιο CU, τότε αυτή η θέση αποκλείεται αφού το ίδιο merge θα μπορούσε να επιτευχθεί από ένα CU χωρίς το διαχωρισμό του σε partition πρόβλεψης. Επιπλέον, πλεονάζουσες καταχωρήσεις όπου οι υποψήφιοι έχουν ακριβώς τις ίδιες πληροφορίες κίνησης επίσης αποκλείονται. Για τον χρονικό υποψήφιο, χρησιμοποιείται η κάτω δεξιά θέση ακριβώς έξω από το PU του οποίου η εικόνα αναφοράς χρησιμοποιείται, εάν είναι διαθέσιμη. Αλλιώς, η κεντρική θέση χρησιμοποιείται. Ο τρόπος που επιλέγεται το collocated PU είναι παρόμοιος με αυτόν των προηγούμενων προτύπων, αλλά το HEVC επιτρέπει περισσότερη ευελιξία μεταδίδοντας έναν δείκτη για να ορίσει ποια λίστα εικόνων αναφοράς χρησιμοποιείται για την collocated εικόνα αναφοράς. Ένα πρόβλημα που σχετίζεται με τη χρήση χρονικών υποψηφίων, είναι το σύνολο μνήμης που χρειάζεται για την αποθήκευση των πληροφοριών κίνησης για τη reference εικόνα. Αυτό αντιμετωπίζεται περιορίζοντας το granularity για την αποθήκευση των χρονικών υποψηφίων κίνησης μόνο μέχρι το επίπεδο των 16x16 luma blocks, ακόμα και αν χρησιμοποιούνται μικρότερες PB δομές στις αντίστοιχες θέσεις της εικόνας αναφοράς. Επιπλέον, ένα PPS επιπέδου flag επιτρέπει στον κωδικοποιητή να απενεργοποιήσει τη χρήση χρονικού υποψήφιου, το οποίο είναι χρήσιμο για εφαρμογές με μετάδοση ευάλωτη στα σφάλματα. Ο μέγιστος αριθμός merge υποψηφίων C ορίζεται με ένα slice header. Εάν ο αριθμός των merge υποψηφίων που έχουν βρεθεί είναι μεγαλύτερος από C τότε μόνο οι πρώτοι C-1 χωρικοί υποψήφιοι και ο χρονικός υποψήφιος διατηρούνται. Αλλιώς, εάν ο αριθμός των merge υποψηφίων είναι μικρότερος από C, τότε επιπλέον υποψήφιοι

παράγονται μέχρι ο αριθμός να είναι ίσος με C. Αυτό απλοποιεί την προσπέλαση και την κάνει πιο ανθεκτική αφού η ικανότητα της προσπέλασης των κωδικοποιημένων δεδομένων δεν εξαρτάται από τη διαθεσιμότητα των merge υποψηφίων.

I. Μετασχηματισμός Scaling και Κβαντισμός.

Το HEVC χρησιμοποιεί κωδικοποίηση μετασχηματισμού του prediction error residual σήματος με παρόμοιο τρόπο όπως στα προηγούμενα πρότυπα. Το residual block χωρίζεται σε πολλά τετράγωνα TBs όπως περιγράφεται παραπάνω. Τα υποστηριζόμενα μεγέθη μετασχηματισμού είναι 4x4, 8x8, 16x16 και 32x32.

1) Core Transform

Υπολογίζονται μετασχηματισμοί 2-D εφαρμόζοντας 1-D μετασχηματισμούς στις οριζόντιες και κάθετες κατευθύνσεις. Τα στοιχεία των core transform πινάκων προέρχονται από τις βασικές εξισώσεις του DCT, μετά από προσαρμογή έχοντας υπόψιν τον περιορισμό του dynamic range για τον υπολογισμό του μετασχηματισμού και η μεγιστοποίηση της ακριβείας και μικρή απόκλιση από την ορθογωνιότητα όταν τα στοιχεία του πίνακα ορίζονται ως ακέραιες τιμές. Για λόγους απλότητας μόνο ένας ακέραιος πίνακας μήκους 32 στοιχείων ορίζεται και subsampled εκδοχές χρησιμοποιούνται για άλλα μεγέθη. Οι πίνακες για τους μετασχηματισμούς μήκους 4 και 8 μπορούν να προέλθουν από τα πρώτα 8 στοιχεία των γραμμών 0, 2, 4 και χρησιμοποιώντας τα πρώτα 4 στοιχεία των γραμμών 0, 4, 8, ... αντίστοιχα. Εάν και το πρότυπο ορίζει το μετασχηματισμό απλά ως τιμές ενός πίνακα, τα στοιχεία στον πίνακα επιλέχθηκαν έτσι ώστε να έχουν συμμετρικές ιδιότητες που επιτρέπουν γρήγορες μερικώς factored υλοποιήσεις με πολύ λιγότερες μαθηματικές πράξεις σε σχέση με έναν κανονικό πολλαπλασιασμό πινάκων και οι μεγαλύτεροι μετασχηματισμοί μπορούν να κατασκευαστούν χρησιμοποιώντας τους μικρότερους ως δομικά blocks. Λόγω του αυξημένου πλήθους των υποστηριζόμενων μετασχηματισμών, ο περιορισμός του dynamic range των ενδιάμεσων αποτελεσμάτων από την πρώτη φάση των μετασχηματισμών είναι αρκετά σημαντικός. Το HEVC εισάγει μια 7-b right shift και 16-b clipping λειτουργία μετά από την πρώτη φάση του 1-D αντιστρόφου μετασχηματισμού για να εξασφαλίσει πως όλα τα ενδιάμεσα αποτελέσματα μπορούν να αποθηκευτούν σε 16-b μνήμη (για 9-b video αποκωδικοποίηση).

2) Εναλλακτικός 4x4 μετασχηματισμός

Για το block μετασχηματισμού 4x4, ένας εναλλακτικός ακέραιος μετασχηματισμός έχει προέλθει από το DST και εφαρμόζεται στα luma residual blocks για τρόπους intra πρόβλεψης. Οι βασικές συναρτήσεις του DST ταιριάζουν περισσότερο στην στατιστική ιδιότητα πως το residual amplitude τείνει να αυξάνεται όσο αυξάνεται η απόσταση από τα οριακά δείγματα που χρησιμοποιούνται για την πρόβλεψη. Όσον αφορά την πολυπλοκότητα ο 4x4 DST-style μετασχηματισμός δεν είναι πολύ πάρα πάνω υπολογιστικά ακριβώς από τον 4x4 DCT-style μετασχηματισμό και παρέχει περίπου 1% μείωση bit-rate σε intra κωδικοποίηση. Η χρήση του DCT τύπου μετασχηματισμού περιορίζεται μόνο στα 4x4 blocks μετασχηματισμού από τη στιγμή που βρέθηκε πως σε άλλες περιπτώσεις η βελτίωση στην απόδοση προσθέτοντας και άλλο τύπο μετασχηματισμού ήταν οριακή

3) Scaling και κβαντισμός

Από τη στιγμή που οι γραμμές του πίνακα μετασχηματισμού είναι κοντινές προσεγγίσεις τιμών ομοιόμορφων scaled βασικών συναρτήσεων του ορθοκανονικού DCT, η prescale λειτουργία που ενσωματώθηκε στο dequantization του H.264/MPEG-4 AVC δε χρειάζεται στο HEVC. Αυτή η αποφυγή του scaling βασικών συναρτήσεων συχνότητας χρησιμεύει στη μείωση του μεγέθους της ενδιάμεσης μνήμης, ειδικά όταν σκεφτούμε πως το μέγεθος του μετασχηματισμού μπορεί να είναι και μέχρι 32x32. Για τον κβαντισμό, το HEVC χρησιμοποιεί το ίδιο URQ σχήμα που ελέγχεται από μια παράμετρο κβαντοποίησης (QP) όπως στο H.264/MPEG-4 AVC. Το εύρος των τιμών του QP ορίζεται από 0 μέχρι και 51, και μια αύξηση 6 μονάδων διπλασιάζει το μέγεθος βήματος κβαντοποίησης έτσι ώστε η αντιστοιχία των τιμών του QP στα μεγέθη των βημάτων να είναι περίπου λογαριθμική. Επίσης υποστηρίζονται scaling πίνακες. Για τη μείωση της απαιτούμενης μνήμης που χρειάζεται για την αποθήκευση των scaling τιμών συχνότητας, μόνο οι πίνακες κβαντισμού μεγέθους 4x4 και 8x8 χρησιμοποιούνται. Για τους μεγαλύτερους μετασχηματισμούς των μεγεθών 16x16 και 32x32 ένας 8x8 πίνακας στέλνεται και εφαρμόζεται μοιράζοντας τις τιμές μεταξύ 2x2 και 4x4 ομάδες συντελεστών στους υποχώρους συχνότητας, εκτός από τις τιμές στις DC θέσεις, όπου για τις οποίες στέλνονται και εφαρμόζονται διακριτές τιμές.

J. In-Loop φίλτρα

Στο HEVC 2 βήματα διεργασιών ονόματι deblocking filter (DBF) ακολουθούμενο από ένα SAO φίλτρο εφαρμόζονται στα ανακατασκευασμένα δείγματα πριν την εγγραφή τους στο decoded picture buffer στο βρόγχο του αποκωδικοποιητή. Το DBF έχει ως σκοπό να μειώσει τα blocking artifacts που δημιουργούνται λόγω block-based κωδικοποίησης. Το DBF είναι παρόμοιο με το DBF του H.264/MPEG-4 AVC, όπου το SAO εμφανίστηκε μαζί με το HEVC. Ενώ το DBF εφαρμόζεται σε δείγματα που βρίσκονται στα όρια των blocks, το SAO φίλτρο εφαρμόζεται προσαρμοστικά σε όλα τα δείγματα που ικανοποιούν συγκεκριμένες συνθήκες.

1) Deblocking φίλτρο:

Το Deblocking φίλτρο εφαρμόζεται σε όλα τα δείγματα γειτονικά σε ένα PU ή TU άκρο εκτός από την περίπτωση ορίου της εικόνας η όταν το deblocking είναι απενεργοποιημένο κατά το μήκος slice ή tile ορίων. Θα έπρεπε να σημειωθεί πως τα PU και TU όρια θα έπρεπε να ληφθούν υπόψιν αφού τα PU όρια δεν είναι ευθυγραμμισμένα με τα TU όρια σε μερικές περιπτώσεις inter πρόβλεψης των CBs. Αντίθετα με το H.264/MPEG-4 AVC, όπου το deblocking φίλτρο εφαρμόζεται σε 4x4 βάση πλέγματος ενώ το HEVC εφαρμόζει το deblocking φίλτρο μόνο στις άκρες που είναι ευθυγραμμισμένες σε ένα 8x8 πλέγμα, για τα luma και chroma δείγματα. Αυτός ο περιορισμός μειώνει την πολυπλοκότητα της χειρότερης περίπτωσης χωρίς ορατή μείωση στην ποιότητα. Επίσης βελτιώνει τη λειτουργία παράλληλης επεξεργασίας αποτρέποντας την αλληλεπικαλυπτόμενη αλληλεπίδραση μεταξύ κοντινών λειτουργιών φιλτραρίσματος. Η ισχύς του φίλτρου ελέγχεται από τις τιμές πολλών συντακτικών στοιχείων παρόμοια με το σχήμα στο H.264/MPEG-4 AVC, αλλά μόνο 3 επίπεδα ισχύεις χρησιμοποιούνται αντί 5. Έστω P και Q 2 γειτονικά blocks με ένα κοινό 8x8 όριο πλέγματος, το φίλτρο ισχύεις 2 ανατίθεται

όταν ένα από τα blocks έχει προβλεφθεί με intra τρόπο. Αλλιώς, το φίλτρο με ισχύει 1 ανατίθεται εάν κάποια από τις παρακάτω συνθήκες ικανοποιείται.

- 1) Το P η το Q έχει έναν τουλάχιστον μη μηδενικό συντελεστή μετασχηματισμού.
- 2) Οι δείκτες αναφοράς των P και Q δεν είναι ίσοι
- 3) Τα διανύσματα κινείζ των P και Q δεν είναι ίσα.
- 4) Η διαφορά μεταξύ ενός στοιχείου του διανύσματος κίνησης του P και Q είναι μεγαλύτερη η ίση από ένα ακέραιο δείγμα.

Εάν καμία συνθήκη από τις παραπάνω δεν ικανοποιείται τότε ανατίθεται το φίλτρο ισχύει 0 που σημαίνει πως η διαδικασία deblocking φιλτραρίσματος δεν εφαρμόζεται.

Σύμφωνα με την ισχύει του φίλτρου και τη μέση παράμετρο κβαντοποίησης των P και Q, 2 όρια, τ_c και β αποφασίζονται από προκαθορισμένους πίνακες. Για τα luma δείγματα, μια από τις 3 περιπτώσεις μη φιλτραρίσματος, δυνατού φιλτραρίσματος η ήπιου φιλτραρίσματος επιλέγεται με βάση το β . Σημειώνεται πως αυτή η απόφαση μοιράζεται μεταξύ 4 luma γραμμών ή στηλών χρησιμοποιώντας τις τελευταίες γραμμές ή στήλες για μείωση της υπολογιστικής πολυπλοκότητας.

Για τα chroma δείγματα υπάρχουν μόνο 2 περιπτώσεις, μη φιλτραρίσματος και ήπιου φιλτραρίσματος. Το ήπιο φιλτράρισμα εφαρμόζεται μόνο όταν η ισχύει του φίλτρου είναι μεγαλύτερο από 1. Η διαδικασία έπειτα εφαρμόζεται χρησιμοποιώντας τις μεταβλητές ελέγχου τ_c και β . Στο HEVC η σειρά επεξεργασίας του deblocking φίλτρου ορίζεται ως οριζόντιο φιλτράρισμα για τις κάθετες άκρες και για όλη την εικόνα αρχικά, και μετά κάθετο φιλτράρισμα για τις οριζόντιες άκρες. Η συγκεκριμένη σειρά επιτρέπει είτε πολλαπλό οριζόντιο φιλτράρισμα είτε καθετί εφαρμογή αυτού σε παράλληλα threads η μπορεί ακόμα να υλοποιηθεί σε μια CTB-προς-CTB βάση μόνο με μια μικρή καθυστέρηση.

2) SAO:

Το SAO είναι μια διαδικασία η οποία τροποποιεί τα δείγματα προσθέτοντας υπό συνθήκη μια offset τιμή σε κάθε δείγμα μετά από την εφαρμογή του deblocking φίλτρου, που βασίζεται πάνω σε look-up πίνακες που μεταδίδονται από τον κωδικοποιητή. Το SAO φιλτράρισμα εφαρμόζεται με βάση την περιοχή και βασίζεται σε έναν τύπο φιλτραρίσματος για κάθε CTB από το συντακτικό στοιχείο `sao_type_idx`. Η τιμή 0 για το `sao_type_idx` υποδεικνύει πως το SAO φίλτρο δεν εφαρμόζεται στο CTB και οι τιμές 1 και 2 υποδεικνύουν τη χρήση band offset και edge offset τύπους φιλτραρίσματος αντίστοιχα.

Στον band offset τρόπο που υποδεικνύεται από την τιμή `sao_type_idx=1`, η επιλεγμένη offset τιμή άμεσα βασίζεται στο amplitude του δείγματος. Σε αυτόν τον τρόπο, ολόκληρο το amplitude εύρος χωρίζεται σε 32 κομμάτια που καλούνται bands και τα δείγματα που ανήκουν στα 4 από αυτά τα bands τροποποιούνται προσθέτοντας τιμές ονόματι band offsets, οι οποίες μπορούν να είναι θετικές η αρνητικές. Ο κύριος λόγος για τη χρήση 4 συνεχόμενων bands είναι πως στις λείες επιφάνειες όπου τα banding artifacts μπορούν να εμφανιστούν να amplitudes των δειγμάτων σε ένα CTB τείνουν να συγκεντρώνονται μόνο

σε λίγα bands. Επιπλέον, η σχεδιαστική επιλογή της χρήσης 4 bands, ενώνεται με τον edge offset τρόπο λειτουργίας ο οποίος χρησιμοποιεί επίσης 4 offset τιμές.

Στον edge offset τρόπο που ορίζεται από την τιμή `sao_type_idx=2`, το συντακτικό στοιχείο με τιμές 0-3 υποδεικνύει εάν οριζόντια, καθετί ή 2 διαγώνιες κατευθύνσεις χρησιμοποιούνται για edge offset ταξινόμηση στο CTB.

2.4. HEVC Profiles και Level Definitions

Η ελαχιστοποίηση των profiles παρέχει το μέγιστο βαθμό διαλειτουργικότητας μεταξύ συσκευών. Τα 3 κύρια profiles Main, Main 10 και Main Still Picture αποτελούνται από εργαλεία κωδικοποίησης και σύνταξη υψηλού επιπέδου που περιγράφονται παραπάνω ενώ υπάρχουν οι παρακάτω περιορισμοί.

- 1) Υποστηρίζεται μόνο 4:2:0 δειγματοληψία chroma
- 2) Όταν ο κωδικοποιείται μια εικόνα χρησιμοποιώντας πολλαπλά tiles, δεν είναι δυνατή η χρήση wavefront παράλληλης επεξεργασίας και κάθε tile θα πρέπει να είναι τουλάχιστον 256 luma δείγματα σε πλάτος και 64 luma δείγματα σε ύψος.
- 3) Στα Main και Main Still Picture profiles, μόνο video ακριβείας 8b ανά δείγμα υποστηρίζεται ενώ στο Main 10 υποστηρίζεται video έως 10bit ανά δείγμα.
- 4) Στο Main Still Picture profile, ολόκληρο το bitstream περιέχει μόνο μια κωδικοποιημένη εικόνα (οπότε δεν υποστηρίζεται inter πρόβλεψη).

3^ο ΚΕΦΑΛΑΙΟ

ΤΕΧΝΙΚΕΣ ΠΑΡΑΛΛΗΛΟΠΟΙΗΣΗΣ

Η σύνταξη των υποκεφαλαίων 3.1 και 3.2 βασίστηκε σε απόδοση στα Ελληνικά της [5]

3.1. Slices

Ένας αποκωδικοποιητής video αποτελείται από 2 βασικές διαδικασίες: α) την προσπέλαση του bit-stream που πραγματοποιείται από τον αποκωδικοποιητή εντροπίας και β) την ανακατασκευή εικόνας που πραγματοποιείται από τη μηχανή επεξεργασίας των pixels. Το bit-stream του video είναι οργανωμένο με τέτοιο τρόπο που τα βήματα προσπέλασης και ανακατασκευής για την τρέχουσα θέση βασίζονται σε πληροφορίες που εμφανίζονται πιο νωρίς στο bit-stream. Σε πρακτικές εφαρμογές, ένα video bit-stream μπορεί να μεταδοθεί πάνω από απολεστικά κανάλια πριν φτάσει στον αποκωδικοποιητή. Η απώλεια κομματιού του bit-stream θα μπορούσε να οδηγήσει σε ανικανότητα προσπέλασης η και ανακατασκευής πληροφοριών χρήσιμες αργότερα στο bit-stream. Αυτή η εξάρτηση διαδίδεται και έτσι ένα σφάλμα μπορεί να οδηγήσει σε ανικανότητα επεξεργασίας μεγάλου κομματιού του bit-stream που βρίσκεται μετά από αυτό. Για να περιοριστεί η διαδώση του σφάλματος είναι σημαντική η διακοπή αυτής της εξάρτησης στην επεξεργασία. Προηγούμενα πρότυπα όπως [46] [3] [48] το πετύχαιναν αυτό οργανώνοντας το bit-stream σε ανεξαρτήτως προσπελάσιμες μονάδες ονόματι slices.

Μέσα σε μια εικόνα, ξεχωριστά slices μπορούν ανεξάρτητα να ανακατασκευαστούν. Στο HEVC, τα slices ορίζουν ομάδες από ανεξαρτήτως προσπελάσιμα coding tree blocks (CTBs). Τα slices περιέχουν CTBs τα οποία ακολουθούν σειρά raster scan μέσα σε μια εικόνα. Προηγούμενα πρότυπα όπως το H.264/AVC περιέχουν εργαλεία όπως flexible macroblock ordering (FMO) το οποίο επιτρέπει τον ορισμό slices με τυχαίο σχήμα. Όμως, ενώ το FMO παρέχει καλές δυνατότητες στον ορισμό σχημάτων slices, απαιτεί διαχωρισμό του deblocking filter στο επίπεδο εικόνας από την υπόλοιπη διαδικασία αποκωδικοποίησης. Έτσι, στο H.264/AVC, δεν είναι δυνατό το deblocking filter βασισμένο στα macroblocks με τη χρήση FMO. Κυρίως ως αποτέλεσμα αυτής της ιδιότητας, το FMO δεν περιέχεται στο HEVC, επειδή το deblocking στο επίπεδο εικόνας αυξάνει σημαντικά το εύρος ζώνης μνήμης και οδηγεί σε μείωση της απόδοσης του κωδικοποίησης.

Στο HEVC ο διαχωρισμός των tiles μπορεί να βασίζεται σε περιορισμούς όπως το maximum transmission unit (MTU) ή περιορισμούς επεξεργασίας pixels όπως ο μέγιστος αριθμός από CTBs που περιέχεται σε κάθε slice. Ακολουθώντας raster scan σειρά σε μια εικόνα, καταλήγουμε σε διαχωρισμό ο οποίος έχει μικρότερο επίπεδο χωρικής συσχέτισης μέσα στην εικόνα. Επιπλέον, κάθε slice περιέχει μέσα του ένα συσχετισμένο slice header το οποίο προσθέτει σημαντικό φόρτο σε μικρά bitrates. Επομένως, η απόδοση της κωδικοποίησης πέφτει.

3.2. Tiles

Μια εικόνα στο HEVC χωρίζεται σε coding tree blocks. Επιπλέον, κάθε εικόνα μπορεί να χωριστεί σε γραμμές και στήλες από CTBs, με τη διασταύρωση αυτών να καταλήγει στη δημιουργία ενός

tile, το οποίο είναι πάντα ευθυγραμμισμένο με τα όρια των CTBs. Ως αποτέλεσμα της ευελιξίας διαχωρισμού, ένα tile μπορεί να είναι χωρικά πιο συμπίεσμένο από ένα slice που περιέχει τον ίδιο αριθμό από CTBs. Αυτό έχει το πλεονέκτημα μεγαλύτερης συσχέτισης μεταξύ των pixels σε σύγκριση με τα slices. Ένα επιπλέον πλεονέκτημα είναι πως τα tiles δεν περιέχουν headers για επίτευξη καλύτερης απόδοσης συμπίεσης. Τα CTBs μέσα σε ένα tile περνάνε από raster scan, και τα tiles μέσα σε μια εικόνα περνάνε επίσης από raster scan.

Ο τροποποιημένος τρόπος scan έχει το πλεονέκτημα μειωμένων απαιτήσεων σε buffers γραμμών για το motion estimation. Συγκεκριμένα, για την πρόβλεψη ενός CTB απαιτείται η αποθήκευση των δεδομένων ανακατασκευασμένων pixels που είναι υποψήφια για motion compensation. Χωρίς τα tiles το raster scan μιας εικόνας έχει ως αποτέλεσμα την αποθήκευση πλήθους δειγμάτων ίσο με $PicW * (2 * SRy + CtbHeight)$, όπου PicW είναι το πλάτος της εικόνας, SRy είναι το μέγιστο κάθετο μέγεθος ενός διανύσματος κίνησης σε μονάδα πλήρους δείγματος και το CtbHeight είναι το ύψος ενός CTB, πάλι σε μονάδες πλήρους δείγματος. Με τα tiles, το πλήθος των δειγμάτων είναι περίπου ίσο με $(TileW + 2 * SRx) * (2 * SRy + CtbHeight)$, όπου TileW είναι το πλάτος ενός tile και SRx είναι το μέγιστο οριζόντιο μέγεθος ενός διανύσματος κίνησης. Με τη χρήση tiles, το πλήθος των αποθηκευμένων πληροφοριών μειώνεται ουσιαστικά όταν το PicW είναι μεγαλύτερο από $(TileW + 2 * SRx)$, το οποίο είναι σύνηθες. Να σημειωθεί πως η παραπάνω ανάλυση προϋποθέτει έναν κωδικοποιητή που τρέχει σε έναν πυρήνα και επεξεργάζεται τα tiles σειριακά.

Η διάσπαση των εξαρτήσεων κωδικοποίησης στα όρια των tiles συνεπάγεται πως ένας αποκωδικοποιητής μπορεί να επεξεργαστεί πολλά tiles παράλληλα. Για να γίνει αυτό δυνατό, οι θέσεις των tiles θα πρέπει να σηματοδοτηθούν στο bit-stream. Στο HEVC οι θέσεις όλων των tiles μεταδίδονται εκτός από του πρώτου που υπάρχει στο slice header [49] [50] [51].

Τα tiles έχει αποδειχθεί πως είναι ένα αποτελεσματικό εργαλείο παραλληλισμού για εξισορρόπηση φόρτου [52] [22] [23].

3.3. SIMD

[38] Οι Intel® Advanced Vector Extensions (Intel AVX) είναι ένα σύνολο εντολών για λειτουργίες Single Instruction Multiple Data σε Intel αρχιτεκτονικής επεξεργαστές. Αυτές οι εντολές επεκτείνουν τις προηγούμενες MMX και SSE εντολές προσθέτοντας τα επόμενα χαρακτηριστικά.

- Επέκταση 128-bit SIMD καταχωρητών σε 256 bits. Ο Intel AVX σχεδιασμός φτιάχτηκε έτσι ώστε να υποστηρίξει ακόμα και 1024 bits στο μέλλον.
- Εντολές τριών συντελεστών έχουν προστεθεί. Οι προηγούμενες εντολές λειτουργούσαν με τρόπο όπως $A = A + B$, όπου ο ένας συντελεστής αντικαθίσταται, ενώ οι καινούργιοι συντελεστές μπορούν να λειτουργήσουν με τρόπο $A = B + C$, όπου οι συντελεστές μένουν όπως έχουν.
- Μερικές εντολές δέχονται μέχρι και 4 συντελεστές, καθιστώντας έτσι δυνατή τη σχεδίαση μικρότερου και γρηγορότερου κώδικα αφαιρώντας αχρείαστες εντολές.
- Οι λειτουργίες ευθυγράμμισης της μνήμης είναι πια πιο χαλαρές.

- Ένα καινούργιο σχήμα επέκτασης κωδικοποίησης (VEX) έχει προστεθεί για να επιτρέψει την πιο εύκολη μελλοντική επέκταση και να επιτρέψει πιο μικρές και πιο γρήγορες εντολές.
- Παρόμοιες με τις παραπάνω επεκτάσεις είναι και οι καινούργιες εντολές Fused-Multiply-Add (FMA) οι οποίες επιτρέπουν πιο γρήγορες και πιο ακριβείς ειδικές λειτουργίες όπως $A = A * B + C$. Άλλα καινούργια χαρακτηριστικά περιέχουν εντολές για τη διαχείριση κρυπτογράφησης Advanced Encryption Standard (AES).

Οι SIMD εντολές επιτρέπουν την επεξεργασία πολλαπλών δεδομένων ταυτόχρονα σε ένα βήμα, επιταχύνοντας εργασίες, από κωδικοποίηση video μέχρι ανάλυση προσομοιώσεων. Το υλικό που υποστηρίζει Intel AVX (και FMA) αποτελείται από 16 256-bit YMM καταχωρητές YMM0-YMM15 και έναν καταχωρητή ελέγχου/κατάστασης 32-bit ονόματι MXCSR. Οι AVX καταχωρητές χρησιμοποιούν τους παλιούς XMM καταχωρητές ως το χαμηλότερο κομμάτι τους

3.4. Υλοποίηση

Χρησιμοποιήσαμε τον HM 16.7 reference κωδικοποιητή [36] για αναπτυξιακούς σκοπούς. Το reference software υλοποιεί τα Slices και τα Tiles αλλά σε με σειριακό τρόπο. Για την παραλληλοποίηση χρησιμοποιήσαμε το OpenMP [37] και ακολουθήσαμε παρόμοια προσέγγιση με αυτή που περιγράψαμε για παραλληλοποίηση επιπέδου Slice στο [6]. Ειδικότερα, τα threads φτιάχνονταν σε μια κλάση υψηλού επιπέδου ιεραρχίας (TEncTop κλάση). Με αυτόν τον τρόπο κάθε thread είχε το δικό του αντικείμενο κωδικοποίησης εικόνας (pcPic αντικείμενο), αλλά ήταν υπεύθυνο μόνο για την κωδικοποίηση ενός συγκεκριμένου Tile/Slice μέσα σε μια εικόνα. Το master thread συγκέντρωνε τα κωδικοποιημένα Slices/Tiles στο δικό του pcPic αντικείμενο και έβγαζε το αποτέλεσμα. Θα πρέπει να σημειωθεί πως η προαναφερθείσα στρατηγική παραλληλισμού επιλέχθηκε για το λόγο πως επιβάλλει μικρή προσπάθεια υλοποίησης από τη στιγμή που οι απαιτούμενες συνάρτησης χειρισμού δεδομένων ανάμεσα στα pcPic αντικείμενα και η σχετική διαχείριση μνήμης είχαν ήδη υλοποιηθεί από το reference software. Όσον αφορά τα στοιχεία του διαχωρισμού των Slices και των Tiles, χρησιμοποιήσαμε τις προεπιλεγμένες ομοιόμορφες ρυθμίσεις του κωδικοποιητή. Τέλος, θα έπρεπε να σημειωθεί πως η υλοποίηση στο σύνολο της ήταν ρυθμισμένη για την περίπτωση που υπάρχει ένα προς ένα ανταπόκριση μεταξύ threads, Tiles/Slices και πυρήνων CPU. AVX2 εντολές χρησιμοποιήθηκαν στο RD (Rate Distortion) κομμάτι του κωδικοποιητή, συγκεκριμένα στις συναρτήσεις που σχετίζονται με τους SAD και SSE υπολογισμούς που περιλαμβάνονται στην TComRdCost κλάση. Η ίδια κλάση περιέχει συναρτήσεις για τον μετασχηματισμό Hadamard, αλλά η υλοποίηση τους έμεινε για μελλοντική δουλειά. Το reference software περιλαμβάνει 10 SAD και 7 SSE συναρτήσεις με κάθε μια να λειτουργεί με διαφορετικό αριθμό από pixels ως είσοδο. Η

α) παρουσιάζει κομμάτια κώδικα τις συναρτήσεις xGetSAD32 όπως ήταν υλοποιημένα στο HM16.7. Η συνάρτηση υπολογίζει το SAD 32 ζευγαριών από pixels σε κάθε επανάληψη, εκτελώντας με σειριακό τρόπο 32 αφαιρέσεις, υπολογισμούς απόλυτης τιμής, προσθέσεις και αναθέσεις. Η

β) παρουσιάζει το αντίστοιχο κομμάτι κώδικα που χρησιμοποιεί AVX2. Αρχικά, τέσσερις 256-bit καταχωρητές ορίζονται και ο κάθε ένας φορτώνεται με 16 pixels (16-bit ακέραιοι). Συνεπώς, κάθε αντίστοιχο ζευγάρι καταχωρητών π.χ. (avx_register_A_1, avx_register_A_2) συνολικά αποθηκεύει τα 32 pixels όπως δεικτοτοτούνται από τα piOrg και piCur. Έπειτα δυο διανυσματικές αφαιρέσεις εκτελούνται χρησιμοποιώντας την εντολή _mm256_subs_epi16 η οποία αντιμετωπίζει το 256-bit περιεχόμενο ως πολλαπλάσιο των 16-bit ακεραίων. Με παρόμοιο τρόπο εφαρμόζεται και ο υπολογισμός των απόλυτων τιμών μετά τις αφαιρέσεις και το αποτέλεσμα προστίθεται διανυσματικά σε έναν _m265i καταχωρητή (sad_sum) που περιέχει το τελικό αποτέλεσμα ως ένα διάνυσμα από 16 16-bit ακέραιους που στη συνέχεια πρέπει να αθροιστούν. Αυτό συμβαίνει σε ένα τελικό βήμα έξω από τον κύριο βρόγχο. προφανώς, μια ένα προς ένα αντιστοίχιση του κώδικα της

β) με της

α) θα προϋπέθετε τον κύριο βρόγχο να τοποθετηθεί μέσα στον πρώτο. Όμως, πειραματικά βρέθηκε πως η διαφορά στην απόδοση (λόγω της μικρής πιθανότητας υπερχείλισης) ήταν αμελητέα, οπότε αποφασίσαμε να επιταχύνουμε την διαδικασία. Η υλοποίηση της συνάρτησης που υπολογίζει το SSE, `xGetSSE32` παρουσιάζεται στην Εικόνα 4. Παρομοίως με την `xGetSAD32`, 4 καταχωρητές (`avx_register_A`, `avx_register_B`, `avx_register_A_2`, και `avx_register_A_2`) χρησιμοποιούνται για να αποθηκεύσουν τα pixels εισόδου. Οι καταχωρητές αφαιρούνται ο ένας από τον άλλον σε ζευγάρια και το αποτέλεσμα αποθηκεύεται σε 2 άλλους καταχωρητές (`avx_register_mul` και `avx_register_mul_2`) που στη συνέχεια χρησιμοποιούνται για πολλαπλασιασμό. Θα έπρεπε να σημειωθεί πως ο 16-bit ακέραιος πολλαπλασιασμός με τη χρήση της εντολής `_mm256_mullo_epi16` επιστρέφει τα 16 λιγότερο σημαντικά bits του αποτελέσματος. Αυτό κρίθηκε επαρκές, από τη στιγμή που σε προκαταρκτικά πειράματα, αναγνωρίστηκε πως η περίπτωση μη μηδενικών περισσότερο σημαντικών 16-bits ήταν σπάνια. Οπότε επιλέξαμε ξανά την επιτάχυνση της απόδοσης στο χρόνο από την απόλυτη ακρίβεια.

```

581 Distortion TComRdCost::xGetSAD32( DistParam* pcDtParam )
582 {
583     if ( pcDtParam->bApplyWeight )
584     {
585         return TComRdCostWeightPrediction::xGetSADw( pcDtParam );
586     }
587     const Pel* piOrg = pcDtParam->pOrg;
588     const Pel* piCur = pcDtParam->pCur;
589     Int iRows = pcDtParam->iRows;
590     Int iSubShift = pcDtParam->iSubShift;
591     Int iSubStep = ( 1 << iSubShift );
592     Int iStrideCur = pcDtParam->iStrideCur*iSubStep;
593     Int iStrideOrg = pcDtParam->iStrideOrg*iSubStep;
594
595     Distortion uiSum = 0;
596
597     for( ; iRows != 0; iRows-=iSubStep )
598     {
599         uiSum += abs( piOrg[0] - piCur[0] );
600         uiSum += abs( piOrg[1] - piCur[1] );
601         uiSum += abs( piOrg[2] - piCur[2] );
602         uiSum += abs( piOrg[3] - piCur[3] );
603         uiSum += abs( piOrg[4] - piCur[4] );
604         uiSum += abs( piOrg[5] - piCur[5] );
605         uiSum += abs( piOrg[6] - piCur[6] );
606         uiSum += abs( piOrg[7] - piCur[7] );
607         uiSum += abs( piOrg[8] - piCur[8] );
608         uiSum += abs( piOrg[9] - piCur[9] );
609         uiSum += abs( piOrg[10] - piCur[10] );
610         uiSum += abs( piOrg[11] - piCur[11] );
611         uiSum += abs( piOrg[12] - piCur[12] );
612         uiSum += abs( piOrg[13] - piCur[13] );
613         uiSum += abs( piOrg[14] - piCur[14] );
614         uiSum += abs( piOrg[15] - piCur[15] );
615         uiSum += abs( piOrg[16] - piCur[16] );
616         uiSum += abs( piOrg[17] - piCur[17] );
617         uiSum += abs( piOrg[18] - piCur[18] );
618         uiSum += abs( piOrg[19] - piCur[19] );
619         uiSum += abs( piOrg[20] - piCur[20] );
620         uiSum += abs( piOrg[21] - piCur[21] );
621         uiSum += abs( piOrg[22] - piCur[22] );
622         uiSum += abs( piOrg[23] - piCur[23] );
623         uiSum += abs( piOrg[24] - piCur[24] );
624         uiSum += abs( piOrg[25] - piCur[25] );
625         uiSum += abs( piOrg[26] - piCur[26] );
626         uiSum += abs( piOrg[27] - piCur[27] );
627         uiSum += abs( piOrg[28] - piCur[28] );
628         uiSum += abs( piOrg[29] - piCur[29] );
629         uiSum += abs( piOrg[29] - piCur[29] );
630         uiSum += abs( piOrg[30] - piCur[30] );
631         uiSum += abs( piOrg[31] - piCur[31] );
632
633         piOrg += iStrideOrg;
634         piCur += iStrideCur;
635     }
636
637     uiSum <<= iSubShift;
638     return ( uiSum >> DISTORTION_PRECISION_ADJUSTMENT(pcDtParam->bitDepth-8) );
639 }

```

Εικόνα 2. Συνάρτηση υπολογισμού SAD με είσοδο 32 pixels.

α) Χωρίς χρήση AVX2

```

562 Distortion TComRdCost::xGetSAD32( DistParam* pcDtParam )
563 {
564     if ( pcDtParam->bApplyWeight )
565     {
566         return TComRdCostWeightPrediction::xGetSADw( pcDtParam );
567     }
568     const Pel* piOrg = pcDtParam->pOrg;
569     const Pel* piCur = pcDtParam->pCur;
570     Int iRows = pcDtParam->iRows;
571     Int iSubShift = pcDtParam->iSubShift;
572     Int iSubStep = ( 1 << iSubShift );
573     Int iStrideCur = pcDtParam->iStrideCur*iSubStep;
574     Int iStrideOrg = pcDtParam->iStrideOrg*iSubStep;
575
576     Distortion uiSum = 0;
577
578     __m256i sad_sum = __m256_setzero_si256();
579     for( ; iRows != 0; iRows-=iSubStep )
580     {
581         __m256i avx_register_A_1 = __m256_loadu_si256((__m256i *)&piOrg[0]);
582         __m256i avx_register_B_1 = __m256_loadu_si256((__m256i *)&piCur[0]);
583         __m256i avx_register_A_2 = __m256_loadu_si256((__m256i *)&piOrg[16]);
584         __m256i avx_register_B_2 = __m256_loadu_si256((__m256i *)&piCur[16]);
585
586         sad_sum = __m256_adds_epi16(sad_sum, __m256_abs_epi16(__m256_subs_epi16(avx_register_A_1, avx_register_B_1)));
587         sad_sum = __m256_adds_epi16(sad_sum, __m256_abs_epi16(__m256_subs_epi16(avx_register_A_2, avx_register_B_2)));
589
589         piOrg += iStrideOrg;
590         piCur += iStrideCur;
591     }
592     uiSum = __m256_extract_epi16(sad_sum, 0)+__m256_extract_epi16(sad_sum, 1)
593           + __m256_extract_epi16(sad_sum, 2)+__m256_extract_epi16(sad_sum, 3)
594           + __m256_extract_epi16(sad_sum, 4)+__m256_extract_epi16(sad_sum, 5)
595           + __m256_extract_epi16(sad_sum, 6)+__m256_extract_epi16(sad_sum, 7)
596           + __m256_extract_epi16(sad_sum, 8)+__m256_extract_epi16(sad_sum, 9)
597           + __m256_extract_epi16(sad_sum, 10)+__m256_extract_epi16(sad_sum, 11)
598           + __m256_extract_epi16(sad_sum, 12)+__m256_extract_epi16(sad_sum, 13)
599           + __m256_extract_epi16(sad_sum, 14)+__m256_extract_epi16(sad_sum, 15);
600
601     uiSum <<= iSubShift;
602     return ( uiSum >> DISTORTION_PRECISION_ADJUSTMENT(pcDtParam->bitDepth-8) );
603 }

```

β) Με χρήση AVX2

Στις περιπτώσεις χειρισμού πλήθους εικονοστοιχείων πολλαπλάσιου του 16 οι προαναφερθείσες εντολές μας ήταν περισσότερο από αρκετές. Στις συναρτήσεις όμως με είσοδο pixels πλήθους υποπολλαπλασίου ή μη πολλαπλασίου του 16 απαιτούνταν ειδικός χειρισμός. Συγκεκριμένα, στις περιπτώσεις αυτές θα υπάρχουν AVX2 καταχωρητές οι οποίοι δε θα είναι πλήρως αξιοποιημένοι. Γι' αυτό το λόγο χρησιμοποιούνται διαφορετικές συναρτήσεις φόρτωσης για τους καταχωρητές. Αυτές οι συναρτήσεις είναι τύπου mask-load αντί απλού load. Χρησιμοποιούν μια μάσκα, η οποία είναι και αυτή ένας `_m256i` 256-bit καταχωρητής, σύμφωνα με την οποία γεμίζουν τους καταχωρητές. Ειδικότερα, η εντολή mask-load θα γεμίσει τις θέσεις του καταχωρητή, ανάλογα με τις θέσεις στις οποίες η μάσκα έχει ως περιεχόμενο προσημασμένο αριθμό με το πρώτο του bit να είναι 1, δηλαδή έναν αρνητικό αριθμό. Τα υπόλοιπα bits των περιεχομένων της μάσκας δεν λαμβάνονται υπόψιν. Στην προκειμένη, όπως φαίνεται στην

β) η μάσκα ορίζεται και με την εντολή `_mm256_setr_epi16` παίρνει στις 12 πρώτες θέσεις το -1 και στις υπόλοιπες 4 το 0. Οι AVX2 βελτιστοποιήσεις, με χρήση `mask-load`, γίναν μόνο στις συναρτήσεις υπολογισμού SAD επειδή τα προκαταρκτικά πειράματα δεν έδειξαν βελτίωση στο χρόνο συμπίεσης στις περιπτώσεις εφαρμογής αυτών στις συναρτήσεις υπολογισμού SSE.

```

500  Distortion TComRdCost::xGetSAD12( DistParam* pcDtParam )
501  {
502      if ( pcDtParam->bApplyWeight )
503      {
504          return TComRdCostWeightPrediction::xGetSADw( pcDtParam );
505      }
506      const Pel* piOrg = pcDtParam->pOrg;
507      const Pel* piCur = pcDtParam->pCur;
508      Int iRows = pcDtParam->iRows;
509      Int iSubShift = pcDtParam->iSubShift;
510      Int iSubStep = ( 1 << iSubShift );
511      Int iStrideCur = pcDtParam->iStrideCur*iSubStep;
512      Int iStrideOrg = pcDtParam->iStrideOrg*iSubStep;
513
514      Distortion uiSum = 0;
515
516      for( ; iRows != 0; iRows-=iSubStep )
517      {
518          uiSum += abs( piOrg[0] - piCur[0] );
519          uiSum += abs( piOrg[1] - piCur[1] );
520          uiSum += abs( piOrg[2] - piCur[2] );
521          uiSum += abs( piOrg[3] - piCur[3] );
522          uiSum += abs( piOrg[4] - piCur[4] );
523          uiSum += abs( piOrg[5] - piCur[5] );
524          uiSum += abs( piOrg[6] - piCur[6] );
525          uiSum += abs( piOrg[7] - piCur[7] );
526          uiSum += abs( piOrg[8] - piCur[8] );
527          uiSum += abs( piOrg[9] - piCur[9] );
528          uiSum += abs( piOrg[10] - piCur[10] );
529          uiSum += abs( piOrg[11] - piCur[11] );
530
531          piOrg += iStrideOrg;
532          piCur += iStrideCur;
533      }
534
535      uiSum <<= iSubShift;
536      return ( uiSum >> DISTORTION_PRECISION_ADJUSTMENT(pcDtParam->bitDepth-8) );
537  }

```

Εικόνα 3. Συνάρτηση υπολογισμού SAD με είσοδο 12 pixels.

α) Χωρίς χρήση AVX2


```

496 Distortion TComRdCost::xGetSAD12( DistParam* pcDtParam )
497 {
498     if ( pcDtParam->bApplyWeight )
499     {
500         return TComRdCostWeightPrediction::xGetSADw( pcDtParam );
501     }
502     const Pel* piOrg = pcDtParam->pOrg;
503     const Pel* piCur = pcDtParam->pCur;
504     Int iRows = pcDtParam->iRows;
505     Int iSubShift = pcDtParam->iSubShift;
506     Int iSubStep = ( 1 << iSubShift );
507     Int iStrideCur = pcDtParam->iStrideCur*iSubStep;
508     Int iStrideOrg = pcDtParam->iStrideOrg*iSubStep;
509
510     Distortion uiSum = 0;
511
512     __m256i sad_sum = _mm256_setzero_si256();
513     __m256i mask = _mm256_setr_epi16(-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 0, 0, 0, 0);
514     for( ; iRows != 0; iRows-=iSubStep )
515     {
516         __m256i avx_register_A_1 = _mm256_maskload_epi32((const int*)piOrg, mask);
517         __m256i avx_register_B_1 = _mm256_maskload_epi32((const int*)piCur, mask);
518
519         sad_sum = _mm256_adds_epi16(sad_sum, _mm256_abs_epi16(_mm256_subs_epi16(avx_register_A_1, avx_register_B_1)));
520
521         piOrg += iStrideOrg;
522         piCur += iStrideCur;
523     }
524     uiSum = _mm256_extract_epi16(sad_sum, 0)+_mm256_extract_epi16(sad_sum, 1)
525           + _mm256_extract_epi16(sad_sum, 2)+_mm256_extract_epi16(sad_sum, 3)
526           + _mm256_extract_epi16(sad_sum, 4)+_mm256_extract_epi16(sad_sum, 5)
527           + _mm256_extract_epi16(sad_sum, 6)+_mm256_extract_epi16(sad_sum, 7)
528           + _mm256_extract_epi16(sad_sum, 8)+_mm256_extract_epi16(sad_sum, 9)
529           + _mm256_extract_epi16(sad_sum, 10)+_mm256_extract_epi16(sad_sum, 11)
530           + _mm256_extract_epi16(sad_sum, 12)+_mm256_extract_epi16(sad_sum, 13)
531           + _mm256_extract_epi16(sad_sum, 14)+_mm256_extract_epi16(sad_sum, 15);
532
533     uiSum <<= iSubShift;
534     return ( uiSum >> DISTORTION_PRECISION_ADJUSTMENT(pcDtParam->bitDepth-8) );
535 }

```

β) Με χρήση AVX2

```

1055 Distortion TComRdCost::xGetSSE32( DistParam* pcDtParam )
1056 {
1057     if ( pcDtParam->bApplyWeight )
1058     {
1059         assert( pcDtParam->iCols == 32 );
1060         return TComRdCostWeightPrediction::xGetSSEw( pcDtParam );
1061     }
1062     const Pel* piOrg = pcDtParam->pOrg;
1063     const Pel* piCur = pcDtParam->pCur;
1064     Int iRows = pcDtParam->iRows;
1065     Int iStrideOrg = pcDtParam->iStrideOrg;
1066     Int iStrideCur = pcDtParam->iStrideCur;
1067
1068     Distortion uiSum = 0;
1069     UInt uiShift = DISTORTION_PRECISION_ADJUSTMENT((pcDtParam->bitDepth-8) << 1);
1070
1071     Intermediate_Int iTemp;
1072
1073     for( ; iRows != 0; iRows-- )
1074     {
1075
1076         iTemp = piOrg[ 0] - piCur[ 0]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1077         iTemp = piOrg[ 1] - piCur[ 1]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1078         iTemp = piOrg[ 2] - piCur[ 2]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1079         iTemp = piOrg[ 3] - piCur[ 3]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1080         iTemp = piOrg[ 4] - piCur[ 4]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1081         iTemp = piOrg[ 5] - piCur[ 5]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1082         iTemp = piOrg[ 6] - piCur[ 6]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1083         iTemp = piOrg[ 7] - piCur[ 7]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1084         iTemp = piOrg[ 8] - piCur[ 8]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1085         iTemp = piOrg[ 9] - piCur[ 9]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1086         iTemp = piOrg[10] - piCur[10]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1087         iTemp = piOrg[11] - piCur[11]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1088         iTemp = piOrg[12] - piCur[12]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1089         iTemp = piOrg[13] - piCur[13]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1090         iTemp = piOrg[14] - piCur[14]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1091         iTemp = piOrg[15] - piCur[15]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1092         iTemp = piOrg[16] - piCur[16]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1093         iTemp = piOrg[17] - piCur[17]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1094         iTemp = piOrg[18] - piCur[18]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1095         iTemp = piOrg[19] - piCur[19]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1096         iTemp = piOrg[20] - piCur[20]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1097         iTemp = piOrg[21] - piCur[21]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1098         iTemp = piOrg[22] - piCur[22]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1099         iTemp = piOrg[23] - piCur[23]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1100         iTemp = piOrg[24] - piCur[24]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1101         iTemp = piOrg[25] - piCur[25]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1102         iTemp = piOrg[26] - piCur[26]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1103         iTemp = piOrg[27] - piCur[27]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1104         iTemp = piOrg[28] - piCur[28]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1105         iTemp = piOrg[29] - piCur[29]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1106         iTemp = piOrg[30] - piCur[30]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1107         iTemp = piOrg[31] - piCur[31]; uiSum += Distortion(( iTemp * iTemp ) >> uiShift);
1108
1109         piOrg += iStrideOrg;
1110         piCur += iStrideCur;
1111     }
1112
1113     return ( uiSum );
1114 }

```

Εικόνα 4. Συνάρτηση υπολογισμού SSE με είσοδο 32 pixels.

α) Χωρίς χρήση AVX2


```

932     Distortion TComRdCost::xGetSSE32( DistParam* pcDtParam )
933     {
934     if ( pcDtParam->bApplyWeight )
935     {
936         assert( pcDtParam->iCols == 32 );
937         return TComRdCostWeightPrediction::xGetSSEw( pcDtParam );
938     }
939     const Pel* piOrg = pcDtParam->pOrg;
940     const Pel* piCur = pcDtParam->pCur;
941     Int iRows = pcDtParam->iRows;
942     Int iStrideOrg = pcDtParam->iStrideOrg;
943     Int iStrideCur = pcDtParam->iStrideCur;
944
945     Distortion uiSum = 0;
946     UInt uiShift = DISTORTION_PRECISION_ADJUSTMENT((pcDtParam->bitDepth-8) << 1);
947
948     __m256i sse_sum = _mm256_setzero_si256();
949     for( ; iRows != 0; iRows-- )
950     {
951         __m256i avx_register_A = _mm256_loadu_si256((__m256i *)&piOrg[0]);
952         __m256i avx_register_B = _mm256_loadu_si256((__m256i *)&piCur[0]);
953         __m256i avx_register_A_2 = _mm256_loadu_si256((__m256i *)&piOrg[16]);
954         __m256i avx_register_B_2 = _mm256_loadu_si256((__m256i *)&piCur[16]);
955
956         __m256i avx_register_mul = _mm256_subs_epi16(avx_register_A, avx_register_B);
957         __m256i avx_register_mul_2 = _mm256_subs_epi16(avx_register_A_2, avx_register_B_2);
958
959
960         sse_sum += _mm256_srli_epi16(_mm256_mullo_epi16(avx_register_mul, avx_register_mul), uiShift);
961         sse_sum += _mm256_srli_epi16(_mm256_mullo_epi16(avx_register_mul_2, avx_register_mul_2), uiShift);
962
963         piOrg += iStrideOrg;
964         piCur += iStrideCur;
965     }
966
967     uiSum = _mm256_extract_epi16(sse_sum, 0)+_mm256_extract_epi16(sse_sum, 1)
968         + _mm256_extract_epi16(sse_sum, 2)+_mm256_extract_epi16(sse_sum, 3)
969         + _mm256_extract_epi16(sse_sum, 4)+_mm256_extract_epi16(sse_sum, 5)
970         + _mm256_extract_epi16(sse_sum, 6)+_mm256_extract_epi16(sse_sum, 7)
971         + _mm256_extract_epi16(sse_sum, 8)+_mm256_extract_epi16(sse_sum, 9)
972         + _mm256_extract_epi16(sse_sum, 10)+_mm256_extract_epi16(sse_sum, 11)
973         + _mm256_extract_epi16(sse_sum, 12)+_mm256_extract_epi16(sse_sum, 13)
974         + _mm256_extract_epi16(sse_sum, 14)+_mm256_extract_epi16(sse_sum, 15);
975
976     return ( uiSum );
977 }

```

β) Με χρήση AVX2

4^ο ΚΕΦΑΛΑΙΟ

ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

4.1. Μεθοδολογία

Τα πειράματα πραγματοποιήθηκαν σε έναν Linux server με 32GB μνήμης και έναν Intel i7- 7700k υπερχρονισμένο στα 4.8 Ghz. Αξιολογήσαμε τις προτεινόμενες τροποποιήσεις χρησιμοποιώντας το HM16.7 software. Σε όλα τα πειράματα το LD (LowDelay) σενάριο χρησιμοποιήθηκε με ένα προπορευόμενο H Frame ακολουθούμενο από P Frames. Οι υπόλοιπες ρυθμίσεις του κωδικοποιητή έχουν ως εξής. Μέγεθος GOP=4, μέγεθος CTU=64x64, μέγιστο βάθος=4 και τρόπος αναζήτησης=TZ. Η αξιολόγηση απόδοσης εκτελέστηκε χρησιμοποιώντας κοινές δοκιμαστικές ακολουθίες video κλάσης A και B με τις λεπτομέρειες να συνοψίζονται στον

Πίνακας 2.

Profile		Quantization	
Profile	main	QP	37
		MaxDeltaQP	0
Unit definition		MaxCuDQPDepth	0
MaxCUWidth	64	DeltaQpRD	0
MaxCUHeight	64	RDOQ	1
MaxPartitionDepth	4	RDOQTS	1
QuadtreeTULog2MaxSize	5		
		Deblock Filter	
QuadtreeTULog2MinSize	2	LoopFilterOffsetInPPS	1
		LoopFilterDisable	0
QuadtreeTUMaxDepthInter	3	LoopFilterBetaOffset_div2	0
QuadtreeTUMaxDepthIntra	3	LoopFilterTcOffset_div2	0
		DeblockingFilterMetric	0
Coding Structure			
IntraPeriod	-1	Misc.	
DecodingRefreshType	0	InternalBitDepth	8
GOPSize	4		
		Coding Tools	
		SAO	1
Motion Search		AMP	1
FastSearch	1	TransformSkip	1
SearchRange	64	TransformSkipFast1	
BipredSearchRange4		SAOLcuBoundary	0
HadamardME	1		
FEN	1	WaveFront	

FDM	1	WaveFrontSynchro	0
Slices			
SliceMode	1	Quantization Matrix	
SliceArgument	128	ScalingList	0
ThreadNumber4		ScalingListFile	scaling_list.txt
LFCrossSliceBoundaryFlag	0		
		Lossless	
PCM		TransquantBypassEnableFlag	0
PCMEnabledFlag	0	CUTransquantBypassFlagForce	0
PCMLog2MaxSize	5		
PCMLog2MinSize	3	Rate Control	
PCMInputBitDepthFlag	1	RateControl	0
PCMFilterDisableFlag	0	TargetBitrate	1000000
		KeepHierarchicalBit	2
Tiles		LCULevelRateControl	1
TileUniformSpacing	0	RCLCUSeparateModel	1
		InitialQP	0
NumTileColumnsMinus1	0	RCForceIntraQP	0
TileColumnWidthArray	1		
NumTileRowsMinus1	0		
TileRowHeightArray	1		
LFCrossTileBoundaryFlag	1		

Πίνακας 1. Παράδειγμα μέρους αρχείου ρυθμίσεων για την ακολουθία BasketballDrive.

Δεδομένου πως οι διαθέσιμοι φυσικοί πυρήνες CPU που τρέχουν στα πειράματα στον server είναι 4, αξιολογήσαμε τα σενάρια που περιείχαν 1, 2, 3, 4(S1, S2, S3, S4) Slices μαζί με ένα σενάριο με 4 Tiles (T4). Τα αποτελέσματα που αποκτήθηκαν ήταν για QP(Παράμετρος κβαντοποίησης) =22, 27 και 32. Ο Πίνακας 3 δείχνει τη μείωση χρόνου που επετεύχθη με τη χρήση AVX2 εντολών ως το ακόλουθο ποσοστό: $(\text{TimeWithoutAVX2} - \text{TimeWithAVX2}) / \text{TimeWithoutAVX2}$.

4.2. Αποτελέσματα

Μπορεί να παρατηρηθεί ως η χρήση AVX2 μειώνει περαιτέρω το χρόνο κωδικοποίησης σε επίπεδο tile και slice παραλληλοποίησης περίπου μεταξύ 4% και 31%. Η μέση μείωση για όλες τις περιπτώσεις ήταν 11.38%. Στη συνέχεια αξιολογήσαμε την απόδοση χρόνου χρησιμοποιώντας Slice ή και Tile επιπέδου παραλληλοποίηση μαζί με SIMD βελτιστοποιήσεις. Ο

Πίνακας 4 απεικονίζει τα αποτελέσματα της επιτάχυνσης που επιτεύχθηκε σε σχέση με τον αρχικό κωδικοποιητή με ένα Slice ($\text{TimeOfOriginal} / \text{TimeOfAdapted}$). Τα θετικά χαρακτηριστικά των βελτιστοποιήσεων AVX2 φαίνονται από το γεγονός πως στο σενάριο του ενός Slice (S1) η επιτάχυνση που επετεύχθη είναι μεγαλύτερη από 1. Επιπλέον, τα πλεονεκτήματα του συνδυασμού της παραλληλοποίησης επιπέδου Tile και Slice εκφράζονται από τις ιδιαίτερα υψηλές επιταχύνσεις που επετεύχθησαν στα S2, S3, S4 και T4 σενάρια. Επίσης προσέξουμε πως σε περισσότερες από τις μισές περιπτώσεις, παρουσιάζεται superlinear επιτάχυνση. Ιδιαίτερα εντυπωσιακά είναι τα αποτελέσματα με 4 Slices/Tiles(S4 και T4) όπου οι επιταχύνσεις έφτασαν το 4.40 και 4.52 αντίστοιχα. Γενικά τα αποτελέσματα του

Πίνακας 4 είναι ιδιαίτερα ενθαρρυντικά προς την επεκτασιμότητα της προτεινόμενης υλοποίησης.

Στη συνέχεια αξιολογήσαμε την επίδραση στην ποιότητα video. Ο Πίνακας 5 παρουσιάζει τη διαφορά σε PSNR(Peak Signal to Noise Ratio) για το Y κομμάτι από το YUV μεταξύ σε Slice/Tile παραλληλοποίηση με και χωρίς AVX2 εντολές. Όπως γίνεται αντιληπτό, οι διαφορές στις περισσότερες περιπτώσεις εμφανίζονται στο τρίτο δεκαδικό ψηφίο, οπότε θεωρούνται αμελητέες. Παρόμοια αποτελέσματα πάρθηκαν και για το bitrate. Αυτά τα αποτελέσματα υποδηλώνουν πως τα πλεονεκτήματα από την AVX2 υλοποίηση που περιγράφονται έρχονται χωρίς επιπλέον φόρτο σε θέματα ποιότητας.

Τέλος, αξιολογήθηκε η επίπτωση του συνδυασμού Slice/Tile παραλληλοποίησης με AVX2 εντολές στην ποιότητα της εικόνας. Οι παρουσιάζουν της Rate Distortion γραφικές παραστάσεις για QP=22, 27, 32. Οι παραστάσεις δείχνουν την απόδοση των S1, S2, S3, S4 και T4, με τον x άξονα σε λογαριθμική κλίμακα.

Από τις γραφικές είναι προφανές πως η διαφορά στην απόδοση μεταξύ S4 και T4 είναι οριακή. Σε σύγκριση με το S1 μπορεί να ειπωθεί πως τα S4 και T4 οδηγούν σε μικρή μείωση απόδοσης με τις μεγαλύτερες διαφορές να υπάρχουν στην ακολουθία Kimono. Παρόλα αυτά,

ακόμα και σε αυτήν την περίπτωση η επιτάχυνση που επιτευχθεί αντισταθμίζει τη μικρή διαφορά στην ποιότητα. Παρόμοια αποτελέσματα παρουσιάζονται και στις υπόλοιπες ακολουθίες.

Συνοψίζοντας τα πειράματα, μπορούμε να δηλώσουμε πως ο συνδυασμός του παραλληλισμού σε επίπεδο Slice/Tile και των AVX2 βελτιστοποιήσεων οδηγεί σε μια κλιμακούμενη υλοποίηση του κωδικοποιητή HEVC που επιτυγχάνει συνήθως superlinear επιτάχυνση με οριακή πτώση στην ποιότητα. Όλα τα αποτελέσματα των πειραμάτων φαίνονται καθαρά και στις Rate-Distortion Optimization γραφικές παραστάσεις, στις εικόνες Εικόνα 5 Εικόνα 6 **Εικόνα 7** Εικόνα 8 Εικόνα 9 Εικόνα 10 Εικόνα 11 Εικόνα 12 Εικόνα 13 Εικόνα 14 Εικόνα 15 Εικόνα 16 Εικόνα 17 Εικόνα 18.

4.3. Γραφικές παραστάσεις και πίνακες αποτελεσμάτων

Όνομα	Ανάλυση	Καρέ ανά δευτερόλεπτο	Συνολικά καρέ	CTUs ανά καρέ
PeopleOnStreet	2560×1600	30	150	100
Traffic	2560×1600	30	150	100
BasketballDrive	1920×1080	50	500	510
BQTerrace	1920×1080	60	600	510
Cactus	1920×1080	50	500	510
Kimono	1920×1080	24	240	510
ParkScene	1920×1080	24	240	510

Πίνακας 2. Χαρακτηριστικά Ακολουθιών Βίντεο

	QP	S1	S2	S3	S4	T4
People On Street	22	7.13%	11.58%	10.98%	10.91%	19.17%
	27	11.32%	12.39%	13.33%	13.08%	21,01%
	32	14.02%	14.90%	14.79%	14.47%	19.97%
Traffic	22	7.19%	5.45%	6.29%	5.63%	5.63%
	27	7.14%	6.27%	7.26%	6.55%	7.91%
	32	7.47%	8.41%	8.65%	6.29%	8.80%
Basket ball Drive	22	10.54%	11,27%	11.99%	11.09%	11.84%
	27	12.99%	15.92%	15.64%	13.67%	14.63%
	32	12.28%	15.82%	15.08%	15.69%	16.25%
BQ Terrace	22	5.23%	4.63%	28.60%	31.23%	4.51%
	27	8.48%	5.86%	28.33%	30.52%	5.81%
	32	7.14%	7.65%	4.95%	13.28%	6.60%
Cactus	22	4.06%	23.55%	8.20%	12.53%	8.57%
	27	5.23%	26.06%	11.80%	17.78%	11.14%
	32	10.32%	11.73%	13.74%	11.78%	13.05%
Kimono	22	10.49%	11.94%	10.15%	9.94%	10.40%
	27	11.20%	13.72%	11.89%	11.58%	12.30%
	32	12.79%	13.06%	12.68%	12.71%	13.23%
Park Scene	22	4.46%	7.42%	5.98%	6.48%	5,34%
	27	5.87%	13.88%	7.94%	7.26%	8.05%
	32	8.14%	8.82%	8.62%	8.45%	9.54%

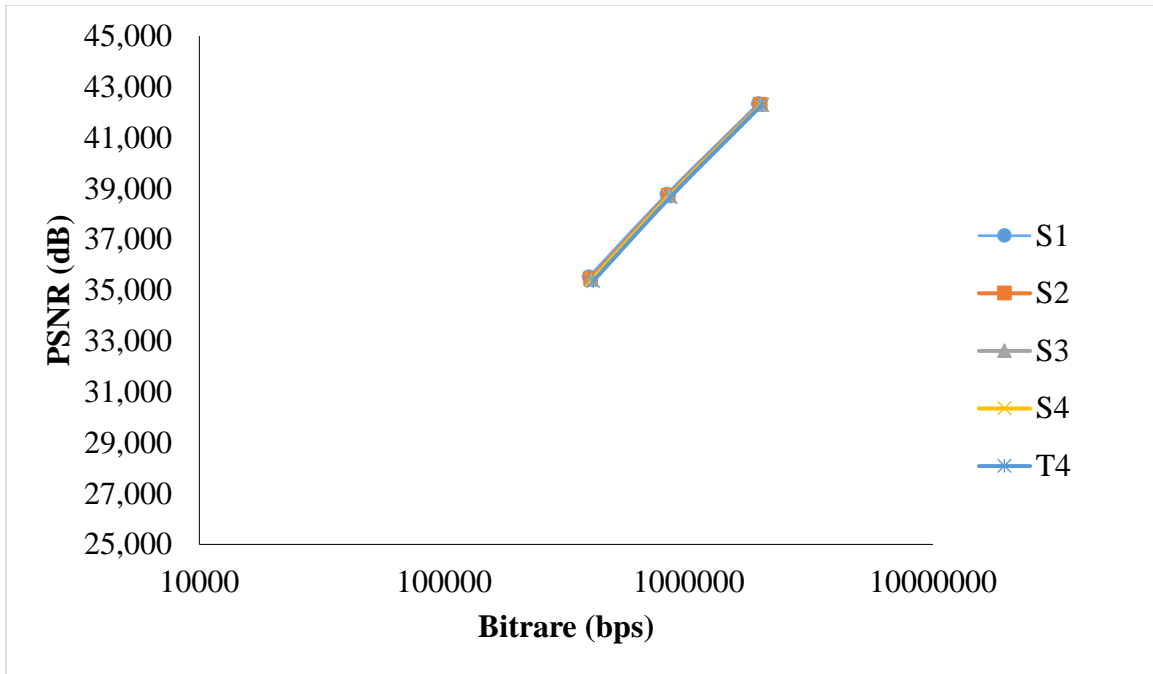
Πίνακας 3. Μείωση χρόνου συμπίεσης με τη χρήση AVX2

	QP	S1	S2	S3	S4	T4
People On Street	22	1.08	2.18	3.38	4.40	4.46
	27	1.13	2.11	3.19	4.13	4.31
	32	1.16	2.28	3.24	4.20	4.40
Traffic	22	1.08	2.25	3.27	4.14	4.52
	27	1.08	2.05	2.76	3.62	3.94
	32	1.08	2.11	2.82	3.73	3.90
Basket ball Drive	22	1.12	1.82	2.83	3.79	3.68
	27	1.15	1.81	2.87	3.79	3.63
	32	1.14	2.14	3.09	3.92	3.72
BQ Terrace	22	1.06	1.96	2.88	3.76	3.96
	27	1.09	1.95	2.87	3.67	3.86
	32	1.08	2.02	2.17	3.84	3.94
Cactus	22	1.04	2.11	2.47	3.80	3.94
	27	1.06	2.19	2.57	4.19	3.75
	32	1.12	2.24	2.44	4.33	3.88
Kimono	22	1.12	1.58	3.01	3.92	3.93
	27	1.13	1.68	3.11	4.04	4.05
	32	1.15	2.14	3.15	4.13	4.14
Park Scene	22	1.05	2.08	3.27	4.25	4.32
	27	1.06	1.91	2.96	3.79	3.88
	32	1.09	2.10	3.05	3.94	3.89

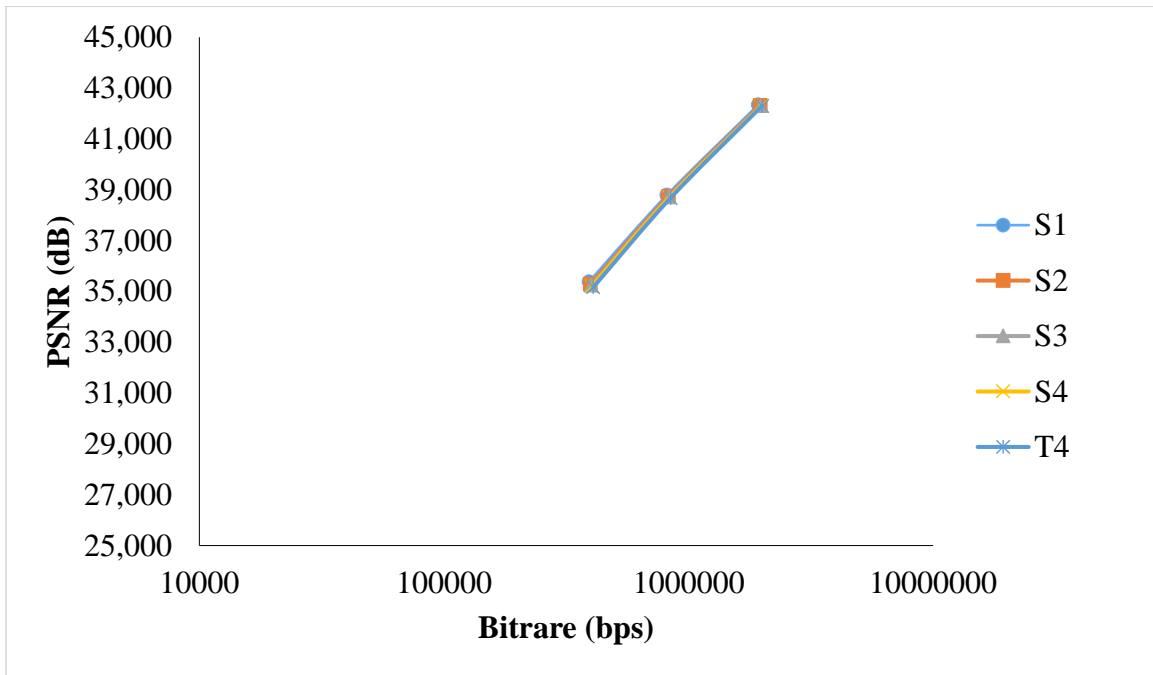
Πίνακας 4. Επιτάχυνση Slice και Tile παραλληλοποίησης σε συνδυασμό με AVX2 βελτιστοποιήσεις

	QP	S1	S2	S3	S4	T4
People On Street	22	0.000	0.001	0.000	0.000	0.000
	27	0.001	0.000	0.002	0.000	-0.001
	32	0.150	0.177	0.188	0.208	0.199
Traffic	22	0.000	0.002	0.000	0.000	0.000
	27	0.002	-0.004	0.001	0.001	0.002
	32	0.003	0.004	0.006	0.002	0.003
Basket ball Drive	22	0.000	0.000	0.000	0.000	0.000
	27	0.001	0.002	0.001	0.003	0.001
	32	0.407	0.433	0.434	0.438	0.407
BQ Terrace	22	-0.001	0.000	0.000	-0.001	-0.001
	27	0.005	0.005	0.006	0.010	0.005
	32	0.251	0.274	0.261	0.311	0.251
Cactus	22	0.000	0.000	0.000	0.000	0.000
	27	0.000	0.000	0.001	0.001	0.000
	32	0.054	0.056	0.058	0.068	0.054
Kimono	22	0.001	0.001	0.000	0.000	0.001
	27	0.001	0.003	0.000	0.002	0.001
	32	0.009	0.005	0.010	0.002	0.009
Park Scene	22	0.001	0.000	-0.001	0.001	0.001
	27	0.008	0.005	0.006	0.002	0.008
	32	0.049	0.070	0.057	0.056	0.049

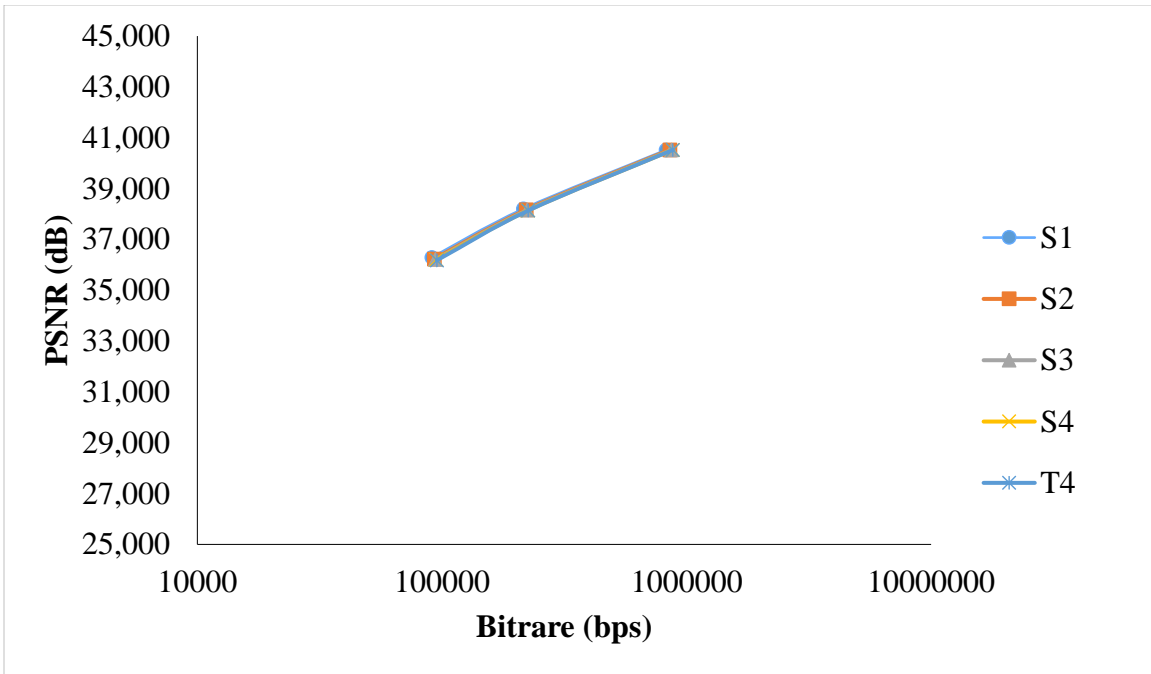
Πίνακας 5. Διαφορά στο PSNR (Y) από τη χρήση AVX2



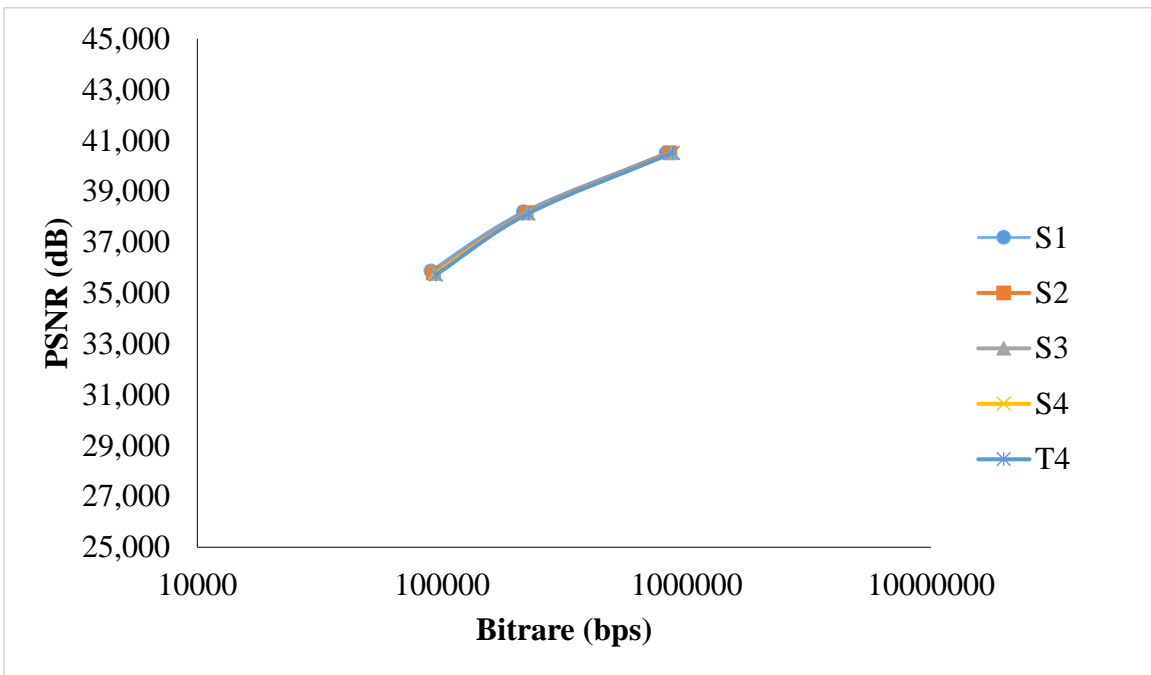
Εικόνα 5. RD Γραφική παράσταση με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία PeopleOnStreet.



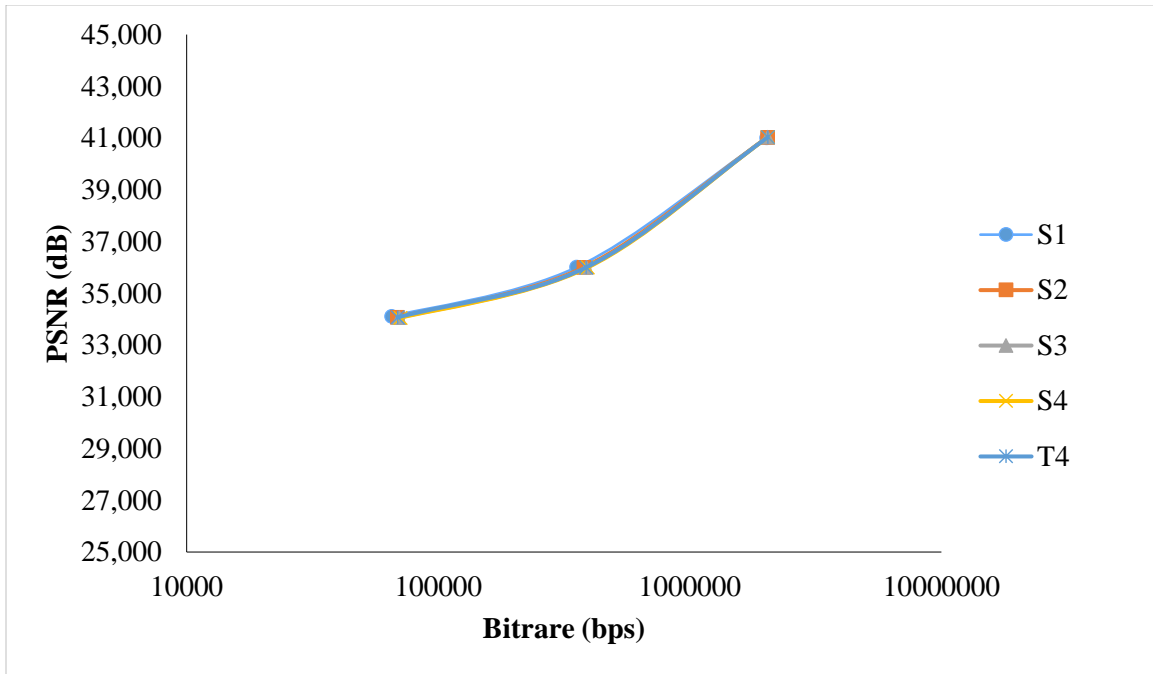
Εικόνα 6. RD Γραφική παράσταση) με τη χρήση Slices/Tiles με AVX2 για την ακολουθία PeopleOnStreet.



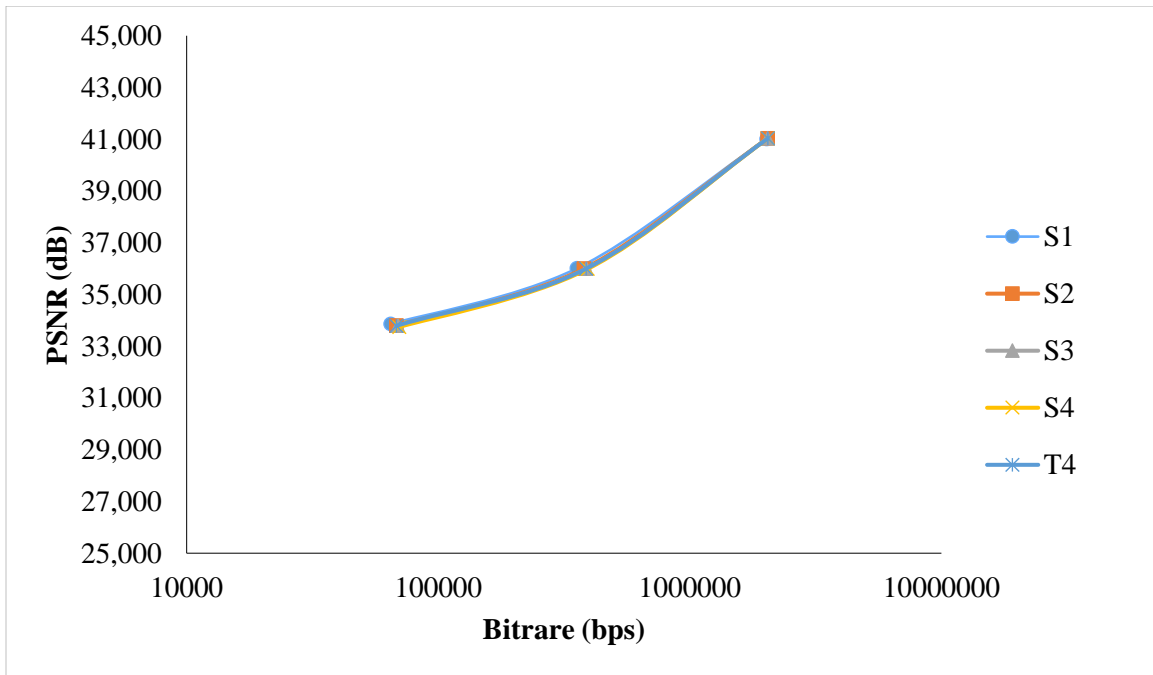
Εικόνα 7. RD Γραφική παράσταση με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία BasketballDrive.



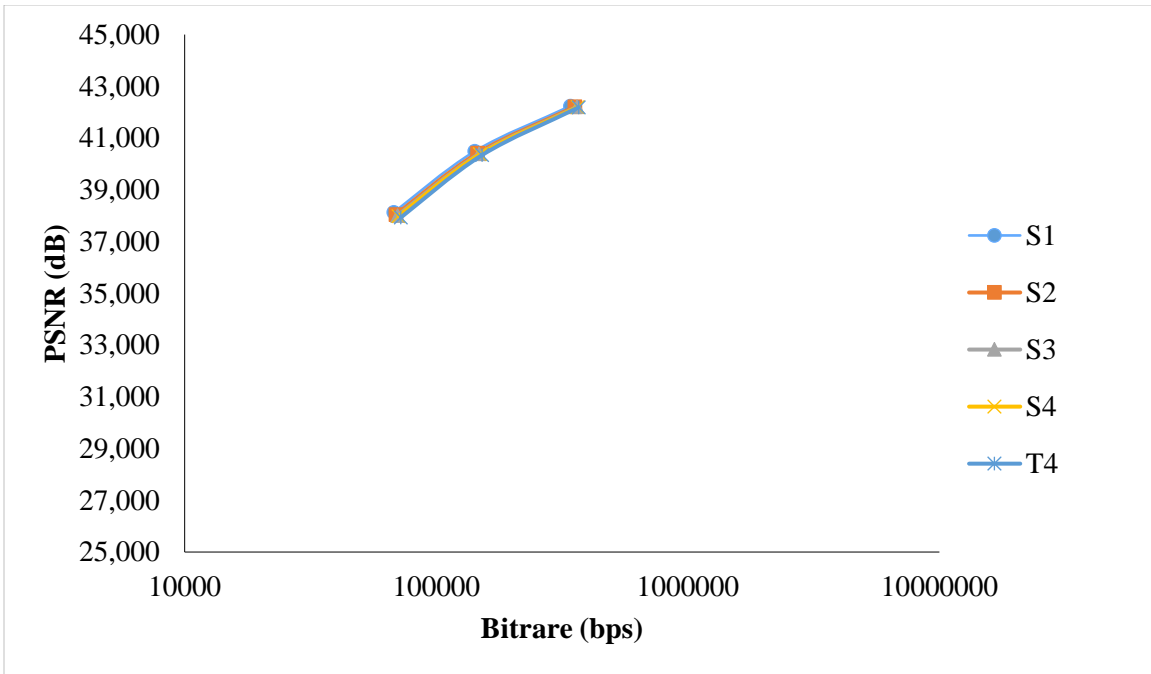
Εικόνα 8. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία BasketballDrive.



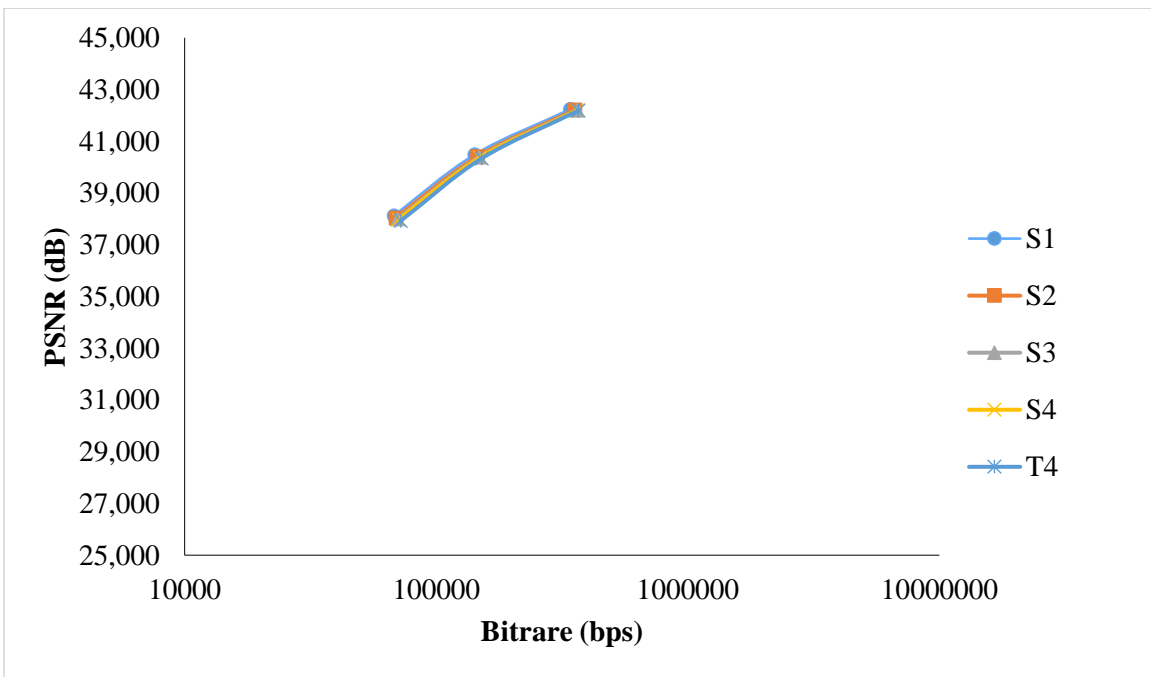
Εικόνα 9. RD Γραφική παράσταση με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία BQTerrace.



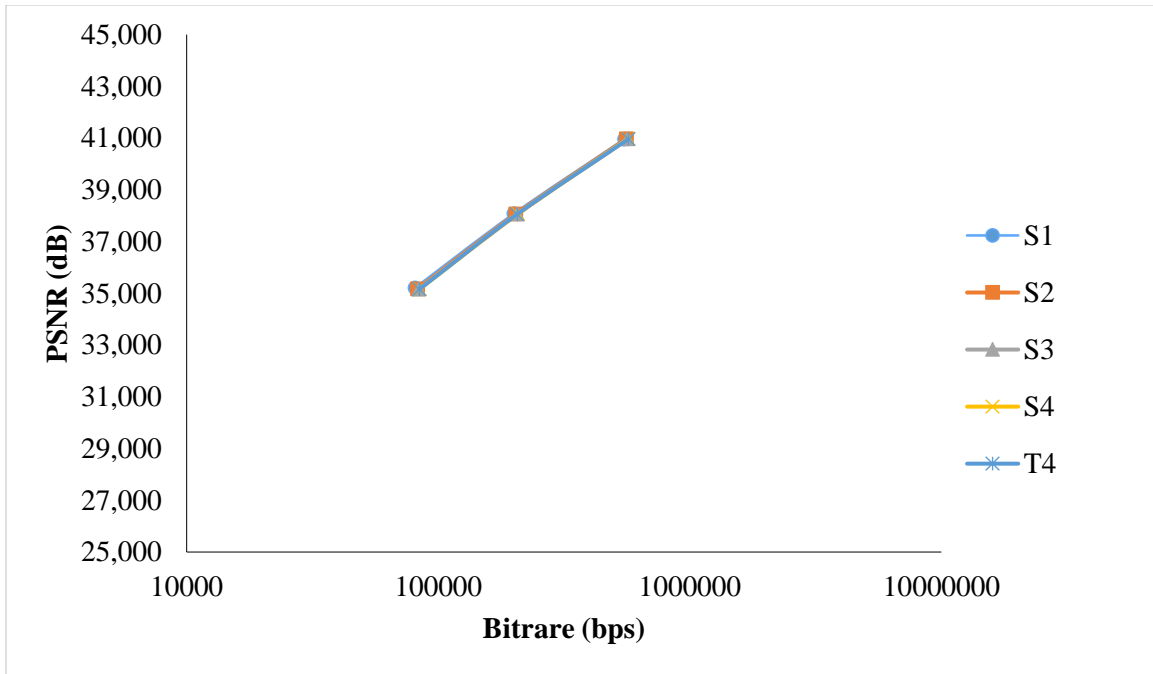
Εικόνα 10. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία BQTerrace.



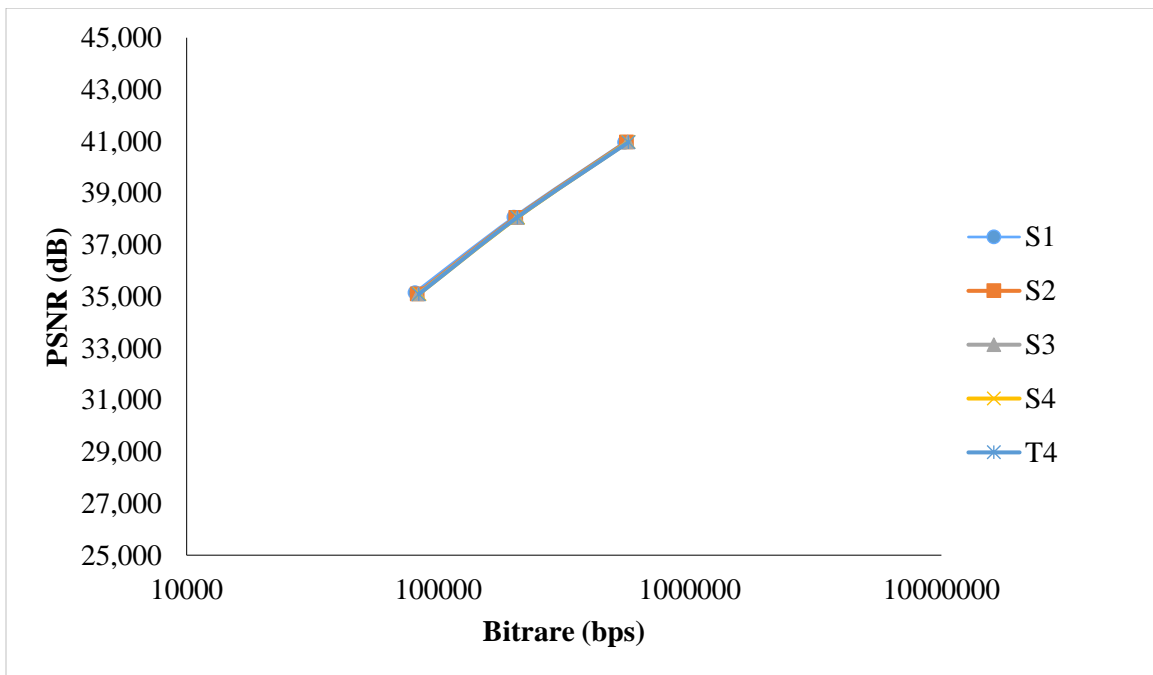
Εικόνα 11. RD Γραφική παράσταση) με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία Kimono.



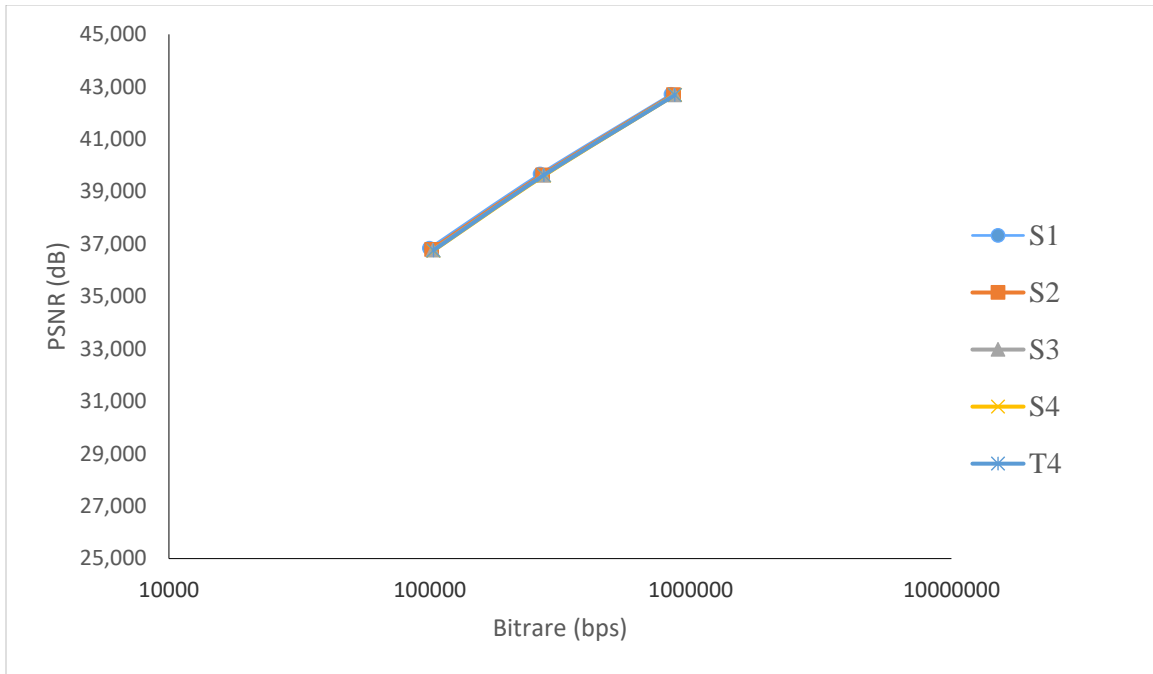
Εικόνα 12. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία Kimono.



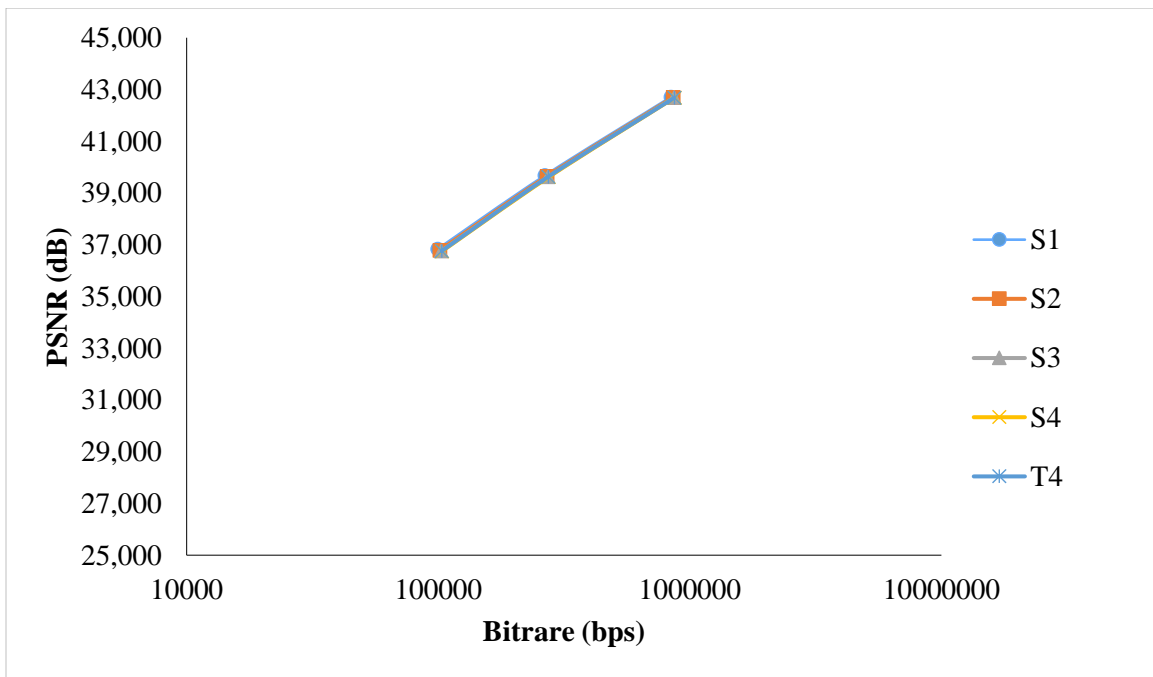
Εικόνα 13. RD Γραφική παράσταση) με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία ParkScene.



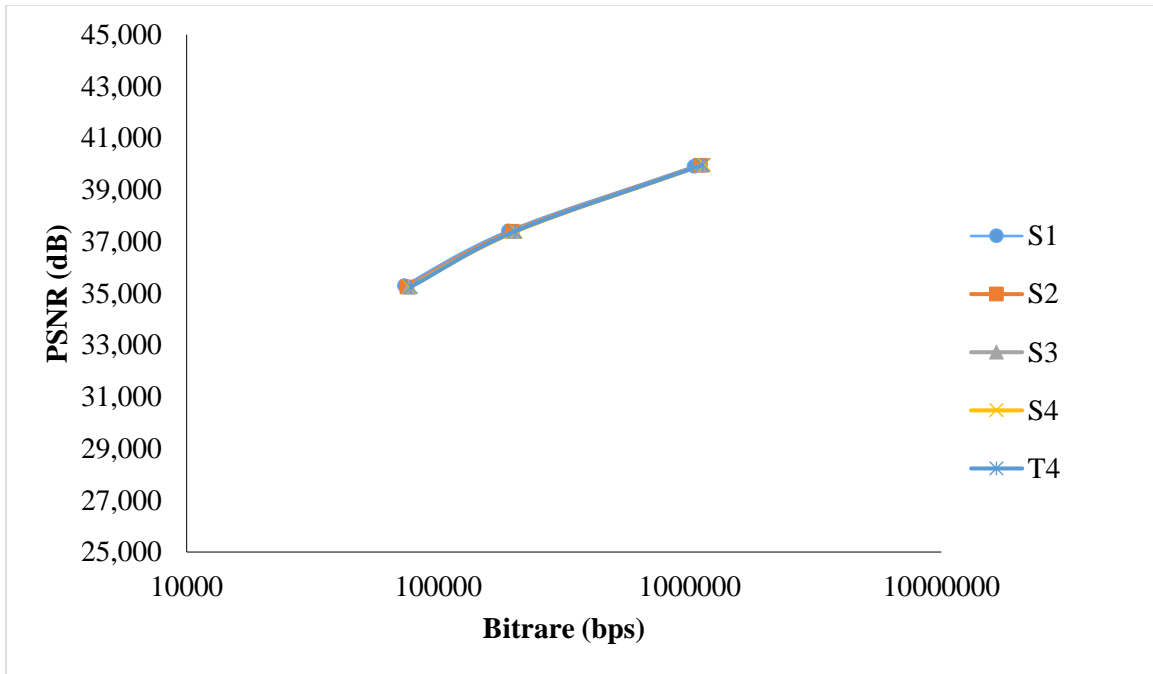
Εικόνα 14. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία ParkScene.



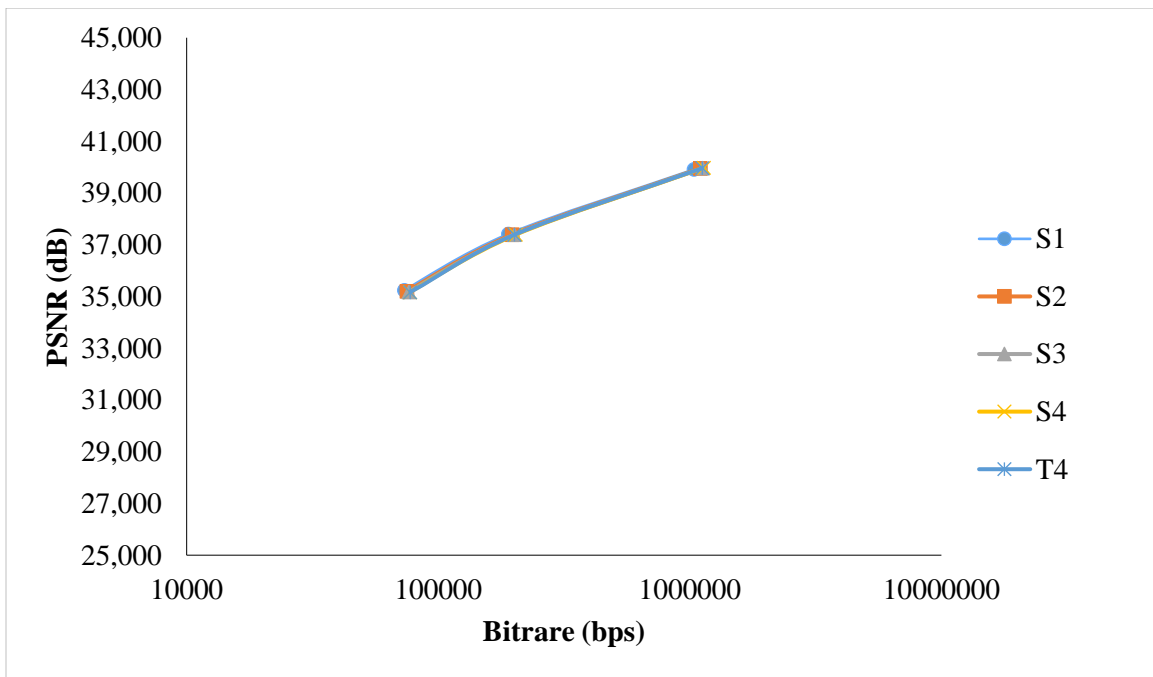
Εικόνα 15. RD Γραφική παράσταση με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία Traffic.



Εικόνα 16. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία Traffic.



Εικόνα 17. RD Γραφική παράσταση με τη χρήση Slices/Tiles χωρίς AVX2 για την ακολουθία Cactus.



Εικόνα 18. RD Γραφική παράσταση με τη χρήση Slices/Tiles με AVX2 για την ακολουθία Cactus.

5^ο ΚΕΦΑΛΑΙΟ

ΕΠΙΛΟΓΟΣ

5.1. Σε αυτήν την εργασία ερευνήσαμε τον συνδυασμό της slice και tile επιπέδου παραλληλοποίηση με SIMD βελτιστοποιήσεις χρησιμοποιώντας AVX2. Το τελικό σχήμα κωδικοποίησης του HEVC έδειξε να έχει καλή πιθανότητα κλιμάκωσης και επιτάχυνση της διαδικασίας κωδικοποίησης χωρίς σημαντική μείωση της απόδοσης όσον αφορά την ποιότητα βίντεο. Ως μέρος της μελλοντικής μας δουλειάς είναι η έρευνα του συνδυασμού AVX2 εντολών και προσαρμοστικού τρόπου διαχωρισμού tiles και slices. Επίσης σκοπεύουμε να αναπτύξουμε slice και tile προσεγγίσεις βασισμένες σε κάρτες γραφικών για τον υπολογισμό SAD, SSE και Hadamard μετασχηματισμού.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] L Hanzo, P J Cherriman, J Streit. Video Compression and Communications Second Edition. s.l. : John Wiley & Sons, Ltd, 2007.
- [2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Techn.* December 2012, pp. 1649-1668.
- [3] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, A. Luthra. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology.* July 2003, pp. 560–576.
- [4] C. C. Chi, M. A. Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, T. Schierl. Parallel Scalability and Efficiency of HEVC Parallelization Approaches. *IEEE Transactions on Circuits and Systems for Video Technology.* December 2012, Vol. 22, 12, pp. 1827-1838.
- [5] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, M. Zhou. An Overview of Tiles in HEVC. *IEEE Journal of Selected Topics in Signal Processing.* December 2013, pp. 969-977.
- [6] P. K. Papadopoulos, M. G. Koziri, N. Tziritas, T. Loukopoulos, I. Anagnostopoulos, G. I. Stamoulis. Performance Evaluation of Batch Encodings in HEVC Using Slice Level Parallelism. *Proceedings of the 20th Pan-Hellenic Conference on Informatics (PCI 2016).* 2016.
- [7] J.-F. Franche, S. Coulombe. Fast H.264 to HEVC Transcoder based on Post-Order Traversal of Quadtree Structure. *Proceedings of the 22nd IEEE International Conference on Image Processing (ICIP 2015).* pp. 477-481.
- [8] M. Koziri, P. 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.*
- [9] P. Oikonomou, M. G. Koziri, N. Tziritas, T. Loukopoulos, C.-Z. Xu. Scheduling Heuristics for Live Video Transcoding on Cloud Edges. *ZTE Comminations.* April 2017, Vol. 15, pp. 35-41.
- [10] M. G. Koziri, P. K. Papadopoulos, N. Tziritas, A. N. Dadaliaris, T. Loukopoulos, G. I. Stamoulis. A Framework for Scheduling the Encoding of Multiple Smart User Videos. *Proceedings of the 11th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP 2016).* 2016, pp. 29-34.
- [11] Yun Q. Shi, Huiyang Sun. *Image and Video Compression for Multimedia Engineering* Second Edition. s.l. : Taylor & Francis Group, LLC., 2007.
- [12] Pearson, D. E. Developments in model-based video coding. *Proceedings of the IEEE.* December 5–9, 1995, Vol. 83, pp. 892–906.
- [13] Codecs for videoconferencing using primary digital group transmission. Recommendation H.120. 1989.

- [14] Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios. ITU-R Recommendation BT.601-5.
- [15] S. M. Akramullah, I. Ahmad, M.L. Liou. A Data-Parallel Approach for Real-Time MPEG-2 Video Encoding. *J. of Parallel and Distributed Computing*. November 1, 1995, 30, pp. 129-146.
- [16] M. G. Koziri, P. K. Papadopoulos, N. Tziritas, A. N. Dadaliaris, T. Loukopoulos, S. U. Khan. Slice-Based Parallelization in HEVC Encoding: Realizing the Potential through Efficient Load Balancing. *Proceedings of the 18th IEEE International Workshop on Multimedia Signal Processing (MMSP 2016)*. 2016, pp. 1-6.
- [17] P. Piñol, H. M. Gomis, O. M. L. Granado, M. P. Malumbres. Slice-Based Parallel Approach for HEVC Encoder. *Journal of Supercomputing*. May 2015, pp. 1882-1892.
- [18] L. Zhao, J. Xu, Y. Zhou, M. Ai. A Dynamic Slice Control Scheme for Slice-Parallel Video Encoding. In *Proceedings of the 19th IEEE International Conference on Image Processing (ICIP 2012)*. pp. 713-716.
- [19] M. Shafique, M. U. K. Khan, J. Henkel. Power Efficient and Workload Balanced Tiling for Parallelized High Efficiency Video Coding. *Proceedings of the 21st IEEE International Conference on Image Processing (ICIP 2014)*. 2014, pp. 1253-1257.
- [20] Y.-J. Ahn, T.-J. Hwang, D.-G. Sim, W.-J. Han. Complexity Model Based Load-Balancing Algorithm for Parallel Tools of HEVC. In *Proceedings of the 2013 Visual Communications and Image Processing (VCIP 2013)*. pp. 1-5.
- [21] C. Blumenberg, D. Palomino, S. Bampi, B. Zatt. Adaptive Content-Based Tile Partitioning Algorithm for the HEVC Standard. *Proceedings of the 30th Picture Coding Symposium (PCS 2013)*. pp. 185-188.
- [22] M. G. Koziri, P. K. Papadopoulos, N. Tziritas, A. N. Dadaliaris, T. Loukopoulos, S. U. Khan, C.-Z. Xu. Adaptive Tile Parallelization for Fast Video Encoding in HEVC. *Proceedings of the 2016 IEEE International Conference on Green Computing and Communications (GreenCOM 2016)*. 2016, pp. 738-743.
- [23] M. Koziri, P. K. Papadopoulos, N. Tziritas, N. Giachoudis, T. Loukopoulos, S. U. Khan, G. I. Stamoulis. Heuristics for Tile Parallelism in HEVC. *Proceedings of the 25th European Signal Processing Conference (EUSIPCO 2017)*.
- [24] K. Chen, Y. Duan, L. Yan, J. Sun, Z. Guo. Efficient SIMD Optimization of HEVC Encoder over x86 Processors. *Proceedings of the 2012 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA 2012)*. 2012, pp. 1-4.
- [25] A. Lemmetti, A. Koivula, M. Viitanen, J. Vanne, T. D. Hämmäläinen. AVX2-Optimized Kvazaar HEVC Intra Encoder. *Proceedings of the 23rd IEEE International Conference on Image Processing (ICIP 2016)*. 2016, pp. 549-553.

- [26] Jim Bankoski, Paul Wilkins, Yaowu Xu. TECHNICAL OVERVIEW OF VP8, AN OPEN SOURCE VIDEO CODEC FOR THE WEB. IEEE International Conference on Multimedia and Expo. 2011.
- [27] J. Bankoski, J. Koleszar, L. Quillio, J. Salonen, P. Wilkins, Y. Xu. VP8 Data Format and Decoding Guide. [Ηλεκτρονικό] January 2011. <http://datatracker.ietf.org/doc/rfc6386/>.
- [28] Generic Coding of Moving Pictures and Associated Audio Information- Part 2: Video. ITU-T and ISO/IEC JTC 1, ITU-T Recommendation H.262 and ISO/IEC 13 818-2 (MPEG-2). 1994.
- [29] ITU-T. Video Coding for Low Bit Rate Communication. ITU-T Recommendation H.263 version 1. 1995.
- [30] Advanced Video Coding for Generic Audiovisual Services. ITU-T Recommendation H.264. 2007.
- [31] Debargha Mukherjee, Jim Bankoski, Adrian Grange, Jingning Han, John Koleszar, Paul Wilkins, Yaowu Xu, Ronald Bultje. The latest open-source video codec VP9 – An overview and preliminary results. Picture Coding Symposium (PCS). December 2013.
- [32] [Online] <http://xiph.org/vorbis/>.
- [33] [Online] <http://corecodec.com/products/matroska>.
- [34] J. Han, A. Saxena, K. Rose. Towards jointly optimal spatial prediction and adaptive transform in video/image coding. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). March 2010, pp. 726–729.
- [35] Draft ITU-T recommendation and final draft international standard of joint video specification.
- [36] HM 16.7 reference software. [Online] <http://hevc.hhi.fraunhofer.de>.
- [37] OpenMP API. [Online] <http://openmp.org>.
- [38] Intel Corp. Intel 64 and IA-32 Architectures Optimization Reference Manual. [Online] June 2016. <https://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-optimization-manual.pdf>.
- [39] Bossen, F. Common Test Conditions and Software Reference Configurations. JCTVC-H1100. 2012.
- [40] D. Skoumpourdis, P. K. Papadopoulos, M. G. Koziri, N. Tziritas, T. Loukopoulos, and I. Anagnostopoulos. On Improving the Speedup of Slice and Tile Level Parallelism in HEVC Using AVX2. Proc. 21st Panhellenic Conf. on Informatics (PCI). September 2017. (to appear).
- [41] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, T. Wiegand. High Efficiency Video Coding (HEVC) Text Specification Draft 9. 2012.
- [42] Video Codec for Audiovisual Services at px64 kbit/s. ITU-T Rec. H.261.

- [43] Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s—Part 2: Video.
- [44] Coding of Audio-Visual Objects—Part 2: Visual. 1999.
- [45] Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video. ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG 2 Video).
- [46] Advanced Video Coding for Generic Audio-Visual Services. ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC).
- [47] H. Schwarz, D. Marpe, T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Trans. Circuits Syst. Video Technol.* 2007.
- [48] S. Wenger, M. Horowitz. FMO: Flexible macroblock ordering. *Proc. JVT-C089.* 2002.
- [49] K. Misra, A. Segall. Parallel decoding with tiles. *Proc. JCT-VC F594.* 2011.
- [50] Hendry, S. Jeong, S. W. Park, B. M. Jeon, K. Misra, A. Segall. AHG4: Harmonized method for signalling entry points of tiles and WPP substreams. *Proc. JCTVC-H0556.* 2012.
- [51] Y.-K. Wang, A. Segall, M. Horowitz, Hendry, W. Wade, F. Henry, T. Lee. Text for tiles, WPP and entropy slices. *Proc. JCTVC-H0737.* 2012.
- [52] M. Zhou, V. Sze, M. Budagavi. Parallel tools in HEVC for high throughput. *Proc. SPIE* 8499. 2012.
- [53] D. Marpe, H. Schwarz, T. Wiegand. Context-adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Trans. Circuits Syst. Video Technol.* 2003, pp. 620–636.
- [54] Maria G. Koziri, Panos K. Papadopoulos, Nikos Tziritas, Antonios N. Dadaliaris, Thanasis Loukopoulos, Georgios I. Stamoulis. On planning the adoption of new video standards in social media networks: a general framework and its application to HEVC. *Social Netw. Analys. Mining.* 2017.