

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή **Στα Συστήματα Ασύρματης Επικοινωνίας**



Video Coding Standards Comparison for Adaptive HTTP Streaming Applications

Λάζαρος Φεγγαρίδης

Επιβλέπων Καθηγητής
Δρ. Αντρέας Παναγίδης

Δεκέμβρης 2018

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Video Coding Standards Comparison for Adaptive HTTP Streaming Applications

Λάζαρος Φεγγαρίδης

**Επιβλέπων Καθηγητής
Δρ. Αντρέας Παναγίδης**

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

Δεκέμβρης 2018

Περίληψη

Οι ραγδαίες εξελίξεις στον τομέα της τεχνολογίας δεν αποτελούν πλέον είδηση. Στην καθημερινότητα μας μπορούμε να διακρίνουμε τα τεράστια βήματα που γίνονται όσον αφορά την ποιότητα του βίντεο που μπορούμε να παρακολουθήσουμε μέσω της τηλεόρασης, του υπολογιστή μας ή και του κινητού μας τηλεφώνου, που διαδραματίζει ολοένα και περισσότερο το ρόλο ενός υπολογιστή τσέπης. Ενώ μέχρι τώρα ένα βίντεο με ανάλυση Full HD (1920x1080) μπορεί να ήταν το ζητούμενο από μέρος μας, βλέπουμε την βιομηχανία του βίντεο να μας προσφέρει πλέον αναλύσεις 2K (2048x1080), 4K (4096x2160) και να προχωρά προς υλοποίηση αναλύσεων 8K (7680x4320) και 16K (15360x8640) για χρήστες οιαδήποτε μέσου αναπαραγωγής βίντεο.

Παράλληλα αντιλαμβανόμαστε καθημερινά και την εξέλιξη των ασύρματων δικτύων και των συσκευών ασύρματης δικτύωσης. Περνάμε από την εποχή του 4G στην 5G στην κινητή τηλεφωνία. Έτσι μπορούμε να διακρίνουμε μεγαλύτερες ταχύτητες μετάδοσης δεδομένων, με μικρότερη καθυστέρηση και με μεγαλύτερες και πιο απρόσκοπτες δυνατότητες σύνδεσης στα ασύρματα δίκτυα. Η μετάδοση βίντεο στις μέρες μας κυριαρχεί με ποσοστό άνω του 70% στο συνολικό ποσοστό μετάδοσης δεδομένων παγκόσμια. Μεγάλη είναι η ανάγκη λοιπόν για καλύτερες τεχνικές συμπίεσης που θα μεταφέρουν την ίδια ποιότητα βίντεο πιο γρήγορα και με μικρότερο μέγεθος.

Στόχος της μεταπτυχιακής διατριβής είναι να αναλυθούν διεξοδικά οι τεχνολογίες συμπίεσης ψηφιακού βίντεο και μέσω ερευνητικής διαδικασίας και αξιολόγησης να επιτευχθεί η σύγκριση της επίδοσης συμπίεσης των πρότυπων κωδικοποίησης ψηφιακού βίντεο που υπάρχουν αυτή την στιγμή, για την αποδοτικότερη αξιοποίηση τους δια μέσου του πρωτοκόλλου MPEG-DASH.

Η ερευνητική μέθοδος που ακολουθήθηκε προς εξαγωγή ασφαλών συμπερασμάτων ήταν να χρησιμοποιήσουμε πρότυπα και ασυμπίεστα βίντεο διαφορετικών αναλύσεων και μέσω της, ελεύθερου λογισμικού, πλατφόρμας επεξεργασίας ψηφιακής εικόνας FFMPEG, να τα επεξεργαστούμε με τους κωδικοποιητές H264, HEVC, VP8, VP9 και του νέου και πολλά υποσχόμενου AV-1. Τα αποτελέσματα βασίστηκαν σε αντικειμενικές μετρικές αξιολόγησης της ποιότητας εικόνας (όπως PSNR) και του απαιτούμενου εύρου ζώνης, καθώς και συνδυαστικά, με τη χρήση του αλγόριθμου BD-Rate/PSNR. Τα εξαχθείσα αποτελέσματα οδηγούν σε ασφαλή συμπεράσματα ως προς την υπεροχή του νέου κωδικοποιητή AV-1 σε σχέση με τους υπολοίπους.

Summary

The rapid developments in technology are no longer news. In our everyday life, we can distinguish the tremendous steps we take on the quality of video that we can watch through our television, computer or our mobile phone, which is increasingly playing the role of a pocket PC. While up to now a Full HD (1920x1080) video might have been ours, we are seeing the video industry now providing us with 2K (2048x1080), 4K (4096x2160) resolutions and progressing to 8K (7680x4320) and 16K (15360x8640) for users of any device than can reproduce video.

At the same time, we understand the development of wireless networks and wireless networking devices every day. We are going through the 4G era on 5G in mobile telephony. Thus, we can distinguish higher data transmission speeds, with less delay and greater and more seamless connectivity capabilities in wireless networks. Video transmission nowadays dominates by over 70% in the overall data transmission rate worldwide. There is a great need for better compression techniques that will transfer the same quality video faster and in smaller size.

The aim of the postgraduate dissertation is to thoroughly analyze digital video compression technologies and through a research process and evaluation to compare the compression performance of the existing digital video coding standards for their more efficient use through the MPEG-DASH protocol.

The research method used to extract safe conclusions was to use standard and uncompressed video of different analytics and through the free software FFMPEG digital image processing platform to process them with the H264, HEVC, VP8, VP9 and new and promising AV-1. The results were based on objective image quality assessment metrics (such as PSNR) and bandwidth, as well as combined using the BD-Rate / PSNR algorithm. The results obtained lead to safe conclusions on the superiority of the new AV-1 encoder over the rest.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω από καρδιάς, τον καθηγητή Δρ. Αντρέα Παναγίδη, όχι μόνο για την πολύτιμη βοήθεια του αλλά και γιατί σε αυτό το προσωπικό ταξίδι στην γνώση, υπήρξε άριστος καθοδηγητής, πηγή έμπνευσης και προπάντων ένας καταπληκτικός άνθρωπος – καθηγητής.

Το μεγαλύτερο ευχαριστώ βέβαια το χρωστάω στα παιδιά μου και στην σύζυγο μου, για την αγάπη τους και την υπομονή τους.

Στα παιδιά μου Βασιλική και Κυριάκο Γεώργιο

Στη σύζυγο μου Μαρία

ΠΕΡΙΕΧΟΜΕΝΑ

Μεταπτυχιακή Διατριβή Στα Συστήματα Ασύρματης Επικοινωνίας.....	i
Περίληψη.....	ii
Summary	iii
Ευχαριστίες.....	iv
Κεφάλαιο 1	7
Εισαγωγή.....	7
1.1 Εισαγωγή.....	7
1.2 Στόχος της Μεταπτυχιακής Διατριβής.....	8
1.3 Δομή της Μεταπτυχιακής Διατριβής.....	9
Κεφάλαιο 2	10
Θεμελιώδεις Αρχές και Πρότυπα Ψηφιακής Κωδικοποίησης Βίντεο.....	10
2.1 Αλγόριθμοι Κωδικοποίησης	11
2.2 Τύποι Πλαισίων.....	14
2.3 Πρότυπα	16
2.4 HTTP Adaptive Streaming.....	30
Κεφάλαιο 3	36
Τεχνολογίες και Λογισμικό Υλοποίησης	36
3.1 Αντικειμενικές Τεχνικές Αξιολόγησης Ποιότητας Βίντεο	36
3.2 Εύρος Ζώνης και Ποιότητα Βίντεο	38
3.3 Εφαρμογές Ανοικτού Κώδικα	38
Κεφάλαιο 4	41
Μεθοδολογία Υλοποίησης	41
4.1 Βάση Δεδομένων - Πρότυπα Ασυμπιεστα Βίντεο	41
4.2 Μεθοδολογία Σύγκρισης Προτύπων Ψηφιακής Κωδικοποίησης, Εξαγωγή και Καταγραφή Αποτελεσμάτων	44
4.3 Εντολές στην Πλατφόρμα FFmpeg.....	49
Κεφάλαιο 5	53
Ανάλυση Αποτελεσμάτων	53
5.1 Ανάλυση Αποτελεσμάτων Α΄ Φάσης	53
5.2 Ανάλυση Αποτελεσμάτων Β΄ Φάσης.....	59
5.3 Συνολική Αποτίμηση	63
Κεφάλαιο 6	65
Επίλογος.....	65
6.1 Γενικότερα Συμπεράσματα	65

6.2	Μελλοντική Εργασία	67
	Βιβλιογραφία	68

Κεφάλαιο 1

Εισαγωγή

Μπορείτε να φανταστείτε μια μέρα που δεν περιλαμβάνει προβολή ή κοινή χρήση ενός βίντεο σε μια από τις συσκευές σας; Η μετάδοση βίντεο σε ένα ευρύ φάσμα συσκευών και ψηφιακών πλατφόρμων παγκοσμίως έχει γίνει αναπόσπαστο κομμάτι της καθημερινότητας μας.

1.1 Εισαγωγή

Η μετάδοση βίντεο είναι τόσο σημαντική ώστε μέχρι το 2021, το 82% της παγκόσμιας κίνησης στο διαδίκτυο θα είναι βίντεο, σύμφωνα με την έρευνα της Cisco [1]. Καθώς οι προσδοκίες των χρηστών για γρήγορο και εξαιρετικά υψηλής ποιότητας βίντεο συνεχίζουν να αυξάνονται, στόχος των ηγετών της βιομηχανίας του βίντεο είναι να αναπτύξουν νέες τεχνολογίες προκειμένου να εκπληρώσουν και να ξεπεράσουν τις προσδοκίες των χρηστών.

Ταυτόχρονα παρακολουθούμε και την ραγδαία ανάπτυξη των ασύρματων δικτύων και των κινητών τηλεφώνων. Σύμφωνα με την ίδια έρευνα η μετάδοση δεδομένων μέσω κινητού τηλεφώνου μέχρι το 2021 θα αυξηθεί και θα φτάσει στο 33% της συνολικής παγκόσμιας κίνησης δεδομένων από 13% που ήταν το 2016, σε αντίθεση με τους προσωπικούς υπολογιστές που από το 46% του 2016 θα φτάσουν το 2021 στο 25% της παγκόσμιας κίνησης δεδομένων.

Consumer Internet Traffic, 2016–2021							
	2016	2017	2018	2019	2020	2021	CAGR 2016–2021
By Network (PB per Month)							
Fixed	52,678	67,081	83,518	103,696	127,152	154,023	24%
Mobile	5,953	9,345	14,029	20,556	29,343	41,417	47%
By Subsegment (PB per Month)							
Internet video	42,029	57,116	75,109	98,182	125,853	159,161	31%
Web, email, and data	9,059	10,681	12,864	15,120	17,502	19,538	17%
Online gaming	915	1,818	2,857	4,396	6,753	10,147	62%
File sharing	6,628	6,810	6,717	6,554	6,388	6,595	0%
By Geography (PB per Month)							
Asia Pacific	20,049	26,401	34,179	44,669	57,659	74,419	30%
North America	19,365	25,132	31,802	39,647	48,224	56,470	24%
Western Europe	8,929	11,475	14,344	17,857	22,011	27,211	25%
Central and Eastern Europe	4,206	5,152	6,321	7,960	10,155	12,822	25%
Middle East and Africa	1,771	2,801	4,218	6,209	9,059	13,229	50%
Latin America	4,311	5,466	6,683	7,909	9,387	11,288	21%
Total (PB per Month)							
Consumer Internet traffic	58,630	76,426	97,547	124,252	156,496	195,440	27%

Source: Cisco VNI, 2017

Πίνακας 1.1: Η μεταφορά βίντεο θα φτάσει το 82% της συνολικής κίνησης του διαδικτύου το 2021 [1].

1.2 Στόχος της Μεταπτυχιακής Διατριβής

Στόχος της μεταπτυχιακής διατριβής είναι αφενός μέσω της υπάρχουσας βιβλιογραφίας και δημοσιευμένων μελετών να αναλύσει τις τεχνολογίες συμπίεσης ψηφιακού βίντεο καθώς και τον τρόπο μετάδοσης τους μέσω του πρωτοκόλλου MPEG-DASH [2] και [3] αφετέρου να αναδείξει την ερευνητική διαδικασία που ακολουθήθηκε ώστε να πραγματοποιηθεί σύγκριση μεταξύ των πρότυπων κωδικοποίησης ψηφιακού βίντεο που υπάρχουν αυτή την δεδομένη χρονική στιγμή και βέβαια να παρουσιάσει τα αποτελέσματά της.

Να απαντήσει δηλαδή στο ερώτημα, ποιος είναι ο πιο αποδοτικός κωδικοποιητής;

1.3 Δομή της Μεταπτυχιακής Διατριβής

Η μεταπτυχιακή διατριβή θα ακολουθήσει την εξής δομή:

Κεφάλαιο 2ο: Θα προσπαθήσουμε να εξηγήσουμε γιατί η κωδικοποίηση του ψηφιακού βίντεο, πριν την μεταφορά του, είναι μια απαραίτητη ενέργεια και στην συνέχεια να παραθέσουμε τους αλγόριθμους που χρησιμοποιούμε προς αυτή την κατεύθυνση. Επίσης θα γίνει αναλυτική αναφορά στους τύπους των Frames που απαρτίζουν το ψηφιακό βίντεο. Ακόμη, θα γίνει μια σύντομη ιστορική αναφορά καθώς και επεξήγηση του αλγορίθμου λειτουργίας των πρότυπων ψηφιακής κωδικοποίησης βίντεο καθώς και του πρωτοκόλλου MPEG-DASH. Τέλος θα γίνει σύντομη αναφορά και σε άλλα πρωτόκολλα που υπάρχουν.

Κεφάλαιο 3ο: Θα παρουσιαστεί η αντικειμενική τεχνική αξιολόγησης ποιότητας βίντεο PSNR με Bitrate καθώς και οι εφαρμογές ανοικτού κώδικα που χρησιμοποιήθηκαν.

Κεφάλαιο 4ο: Στο κεφάλαιο αυτό θα δούμε βήμα βήμα όλες τις ενέργειες που πραγματοποιήθηκαν ώστε να υλοποιηθεί το προσωπικό ερευνητικό κομμάτι της μεταπτυχιακής διατριβής.

Κεφάλαιο 5ο: Θα γίνει ανάλυση των αποτελεσμάτων της έρευνας, σε δύο φάσεις. Η πρώτη θα περιλαμβάνει τα αποτελέσματα των βίντεο με ανάλυση 768x432 και στην δεύτερη τα βίντεο με ανάλυση 1440x1080. Τέλος θα γίνει μια συνολική αποτίμηση των ερευνητικών ευρημάτων.

Κεφάλαιο 6ο: Στον επίλογο της μεταπτυχιακής διατριβής θα αναφέρουμε κάποια γενικότερα συμπεράσματα καθώς και κάποιες σκέψεις για μελλοντική συνέχιση της έρευνας μας.

Κεφάλαιο 2

Θεμελιώδεις Αρχές και Πρότυπα Ψηφιακής Κωδικοποίησης Βίντεο

Μα είναι τόσο απαραίτητη η χρήση των κωδικοποιητών; Δεν μπορεί να αποφευχθεί; Ας υποθέσουμε ότι έχουμε μια FullHD εικόνα. Αυτό σημαίνει ότι η διάσταση της θα είναι 1920x1080 pixels.



Εικόνα 2.1: Αριθμός Pixels σε διαφορετικές αναλύσεις εικόνας [4].

Δηλαδή η εικόνα μας αναλύεται σε 1920 κουκίδες ή εικονοστοιχεία ή pixels οριζόντια και σε 1080 pixels κάθετα. Το τι χρώμα θα έχει το κάθε pixel είναι ένας συνδυασμός τριών χρωμάτων Red, Green, Blue. Το κάθε ένα από αυτά τα χρώματα για να αναπαρασταθεί η τιμή του, χρειάζεται τουλάχιστον 8 bits (σε Deep Colour ανάλυση μπορεί να φτάσει και τα 48 bits). Έτσι ως σύνολο έχουμε 24 bits για κάθε pixel. Τέλος τα Frames Per Second ή αλλιώς τα πλαίσια που χρησιμοποιούνται για την απεικόνιση της εικόνας κάθε δευτερόλεπτο, ας υποθέσουμε ότι είναι 30.

Συνολικά δηλαδή θα χρειαστούμε $24 \times 1920 \times 1080 \times 30 = 1.492.992.000$ bits per second.

Στην χωρά μας σήμερα μια καλή σύνδεση Internet φτάνει τα 50 Mbps, δηλαδή 50.000.000 bits per second. Χρειαζόμαστε λοιπόν 30 συνδέσεις Internet οι οποίες θα πρέπει να χρησιμοποιούνται αποκλειστικά για να βλέπουμε την εικόνα σε FullHD. Καταλαβαίνουμε λοιπόν ότι η συμπίεση του αρχικού βίντεο αποτελεί μονόδρομο. Ποιες όμως είναι οι τεχνικές που χρησιμοποιούνται για αυτή την συμπίεση [5] [6];

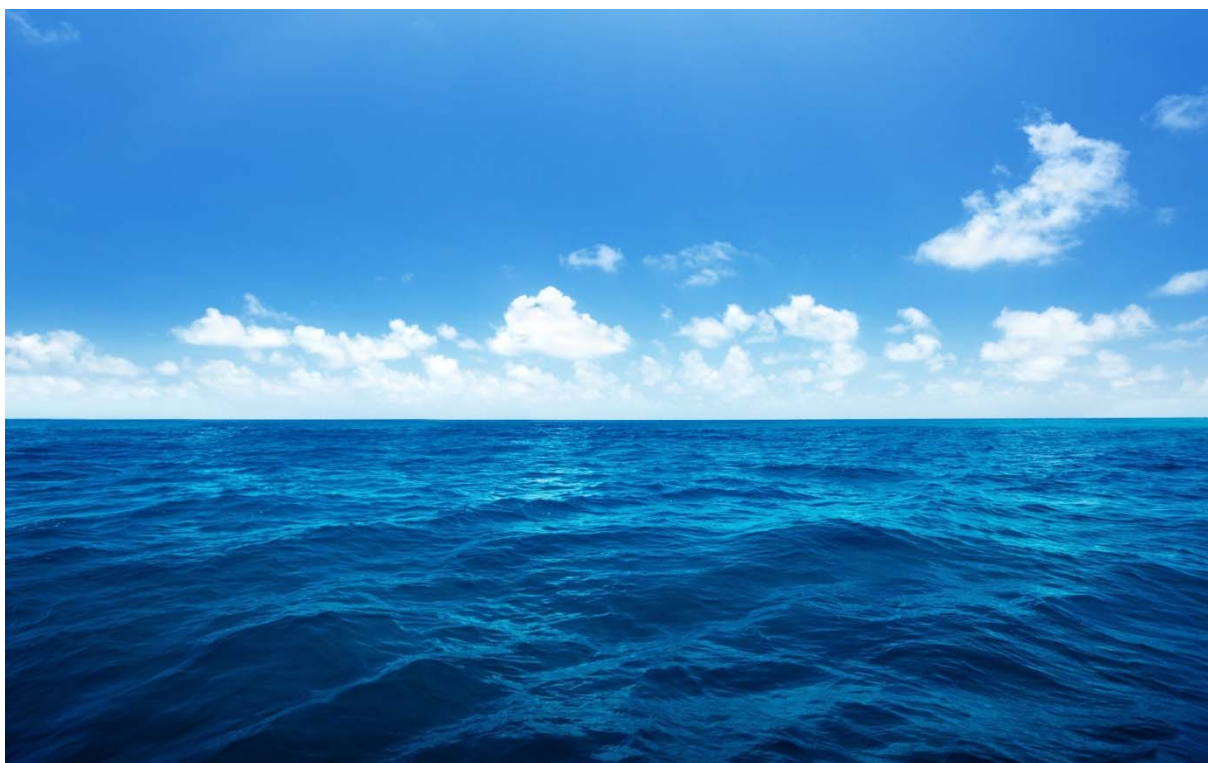
2.1 Αλγόριθμοι Κωδικοποίησης

Στο παράδειγμα μας παραπάνω αναφέραμε ότι χρειαστήκαμε 30 Frames per Second. Δηλαδή σε κάθε δευτερόλεπτο έχουμε 30 εικόνες που διαδέχεται η μια την άλλη. Ο όγκος των δεδομένων που δημιουργείται με αυτό τον τρόπο είναι τεράστιος. Άραγε επιτρέπεται εδώ να επέμβει κάποιος αλγόριθμος συμπίεσης, χωρίς να χαθεί η πιστότητα της πληροφορίας; Η απάντηση είναι

ναι. Αυτό συμβαίνει βάση της φυσιολογίας και της ανατομίας ματιού και αυτιού. Το μάτι μας δεν διακρίνει όλα τα χρώματα και το αυτί μας δεν αντιλαμβάνεται όλες τις συχνότητες του ήχου. Μπορούμε δηλαδή να πούμε ότι υπάρχουν πληροφορίες που δεν τις χρειαζόμαστε γιατί απλά δεν μπορούμε να τις αντιληφθούμε. Αυτές οι πληροφορίες ονομάζονται πλεονασμός.

Τον πλεονασμό τον διακρίνουμε σε δύο κατηγορίες:

Χωρικός Πλεονασμός (Spatial Redundancy): Σε αυτή την περίπτωση τα γειτονικά pixels μιας εικόνας μοιάζουν μεταξύ τους. Για παράδειγμα η εικόνα του ουρανού ή της θάλασσας. Εκεί το μάτι μας αντιδρά κάπως νωχελικά και αφήνει να περάσει ένα ποσοστό αλλοίωσης ή ακόμα και παραμόρφωσης της εικόνας χωρίς να το αντιληφθεί [6].



Εικόνα 2.2: Παράδειγμα χωρικού πλεονασμού.

Χρονικός Πλεονασμός (Temporal Redundancy): Εδώ συμβαίνει οι εικόνες μεταξύ τους να μην είναι και τόσο διαφορετικές. Μια προηγούμενη με μια επόμενη να έχουν ελάχιστες διαφορές. Έτσι κωδικοποιούνται μόνο οι διαφορές που υπάρχουν μεταξύ των 2 εικόνων και προστίθενται στην ήδη υπάρχουσα πληροφορία. Για παράδειγμα η μετάδοση ενός αγώνα snooker. Σε αυτόν βλέπουμε την κάμερα ακίνητη να είναι πάνω από το τραπέζι και το μόνο που κουνιέται είναι ο παίκτης. Ένα πολύ μεγάλο ποσοστό, άνω του 70% της εικόνας, μένει ακίνητο [7] [8].



Εικόνα 2.3: Παράδειγμα χρονικού πλεονασμού.

Τέλος μπορούμε να κατηγοριοποιήσουμε τις μεθόδους συμπίεσης βάση της σχέσης του αρχικού βίντεο με το τελικό αποτέλεσμα της αποσυμπίεσης. Έτσι έχουμε τις **τεχνικές χωρίς απώλειες** όπου το τελικό αποσυμπιεσμένο βίντεο είναι πιστό αντίγραφο του αρχικού και με το ίδιο μέγεθος και τις **τεχνικές με απώλειες** όπου έχουν αφαιρεθεί οι πλεονάζουσες πληροφορίες που περιγράψαμε παραπάνω και φυσικά έχει επιτευχθεί ένα μεγάλο ποσοστό συμπίεσης [9].

2.1.1 Κωδικοποίηση Εντροπίας

Εδώ έχουμε να κάνουμε με μια τεχνική χωρίς απώλειες. Το κυριότερο χαρακτηριστικό της είναι ότι βλέπει την πληροφορία ως ακολουθία από bits και αδιαφορεί για το είδος της. Μπορεί να υλοποιηθεί με δύο τρόπους. Ο πρώτος είναι αυτός του Περιορισμού των Επαναλαμβανόμενων Ακολουθιών (Suppression of repetitive sequences), όπου αν έχουμε για παράδειγμα ένα χαρακτήρα που επαναλαμβάνεται 10 φορές, ο χαρακτήρας αυτός αντικαθίσταται από έναν ειδικό χαρακτήρα (σημαία) και τον αριθμό των επαναλήψεων. Ο δεύτερος τρόπος ονομάζεται Στατιστική Κωδικοποίηση (Statistical encoding) και η οποία βρίσκει τις πιο συχνά εμφανιζόμενες ακολουθίες χαρακτήρων και τις αντικαθιστά με μικρότερο αριθμό ψηφίων. Απαιτείται η ύπαρξη λεξικού όπου αποθηκεύονται οι ακολουθίες και οι αντικαταστάτριες τους, για να μπορέσει να γίνει η αποσυμπίεση [6].

2.1.2 Κωδικοποίηση με Πρόβλεψη

Στην περίπτωση της κωδικοποίησης με πρόβλεψη γίνεται προσπάθεια να εκμεταλλευτεί το γεγονός ότι υπάρχει πλεονασμός μεταξύ των γειτονικών pixels ή ενδιάμεσα μεταξύ διαδοχικών πλαισίων. Σύμφωνα λοιπόν με την τιμή του εικονοστοιχείου που έχει ήδη μεταδοθεί και μας είναι γνωστή και σύμφωνα με την τιμή των γειτονικών pixels γίνεται πρόβλεψη για την νέα τιμή που μπορεί να πάρει. Η διαφορά ανάμεσα στην πραγματική και την προβλεπόμενη τιμή είναι και αυτή που κωδικοποιείται [10].

2.1.3 Πρόβλεψη με Αντιστάθμιση Κίνησης

Σε αυτή την περίπτωση ας υποθέσουμε ότι αποστέλλουμε το πρώτο πλαίσιο μιας εικόνας σε κίνηση. Στο επόμενο πλαίσιο κάποια κομμάτια της θα παραμείνουν ίδια και κάποια άλλα θα αλλάξουν. Η ιδέα της πρόβλεψης με αντιστάθμιση κίνησης είναι να χωρίσει την δεύτερη εικόνα σε ίσα τμήματα και να δημιουργήσει ένα πίνακα με τα τμήματα που έχουν μείνει ίδια και διανύσματα που να δείχνουν την νέα θέση των τμημάτων που έχουν αλλάξει θέση. Αυτό λέγεται εκτίμηση κίνησης. Με αυτό τον τρόπο στέλνουμε την δεύτερη εικόνα ως ένα πίνακα διανυσμάτων και κάποιων σταθερών σημείων που προφανώς έχει μικρότερο μέγεθος [10].

2.2 Τύποι Πλαισίων

Στην συμπίεση ενός βίντεο, τρεις είναι οι βασικές κατηγορίες όπου μπορούμε να κατατάξουμε τα πλαίσια (frames) του βίντεο, βάσει των ιδιοτήτων και των χαρακτηριστικών τους. Στην εικόνα 2.4 βλέπουμε μια συνήθη ακολουθία πλαισίων [11].

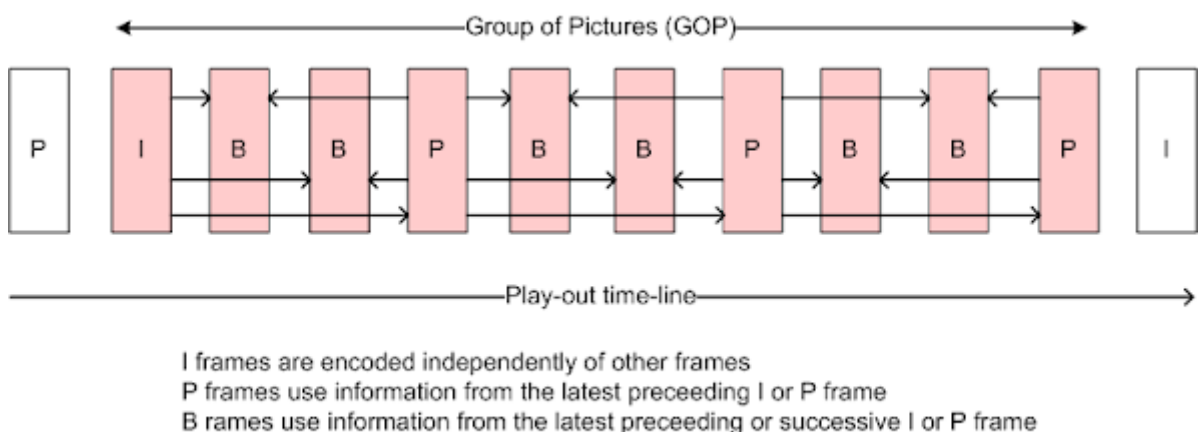


Image created by Sachin Agarwal

Εικόνα 2.4: Συνηθισμένη ακολουθία πλαισίων [12].

2.2.1 I-Frames

Τα I-Frames (Intracoded Frames) είναι ανεξάρτητα από τα άλλα είδη πλαισίων. Όταν αποκωδικοποιούνται δηλαδή δεν γίνεται αναφορά σε άλλο είδος ή σε προηγούμενο ή επόμενο πλαίσιο. Είναι τα μεγαλύτερα σε μέγεθος και τα καλύτερα σε ποιότητα πλαίσια. Θεωρούνται τα πιο σημαντικά πλαίσια και μάλιστα είναι και τα σημεία αναφοράς κατά την τυχαία προσπέλαση του βίντεο, για αυτό τοποθετούνται ανά τακτά χρονικά διαστήματα [13].

2.2.2 P-Frames

Τα P-Frames (Predicted Frames) δημιουργούνται με την τεχνική της πρόβλεψης με αντιστάθμιση της κίνησης και σχετίζονται πάντα με ένα προηγούμενο I ή ένα P πλαίσιο. Περιέχει δηλαδή μόνο τις αλλαγές από το προηγούμενο πλαίσιο. Τις νέες θέσεις των macroblocks που έχουν μετακινηθεί. Όπως καταλαβαίνουμε εύκολα, το μέγεθος τους είναι κατά πολύ μικρότερο από τα I-Frames και δεν έχουν την ίδια ακρίβεια. Είναι υπεύθυνα για την δημιουργία σφαλμάτων αλλά και για την διάδοση τους, σύμφωνα με τον τρόπο δημιουργίας τους.

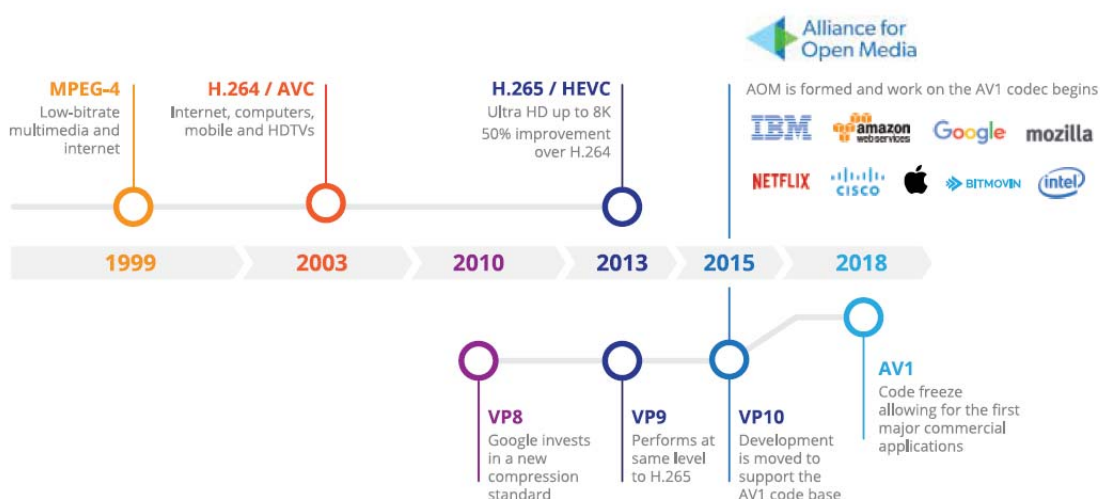
2.2.3 B-Frames

Τα B-Frames (Bidirectional Frames) είναι τα αμφίδρομα πλαίσια. Ονομάζονται έτσι γιατί κάνουν πρόβλεψη προς τα πίσω και προς τα μπροστά, των προηγούμενων ή επόμενων I,P Frames, ανάμεσα στα οποία παρεμβάλλονται. Συμβάλουν στην μείωση του σφάλματος αφού παίρνουν τον μέσο όρο από τα δύο πλαίσια [13].

Οι μεγάλες εξελίξεις στην προσφερόμενη ποιότητα βίντεο καθώς και στην ασύρματη μετάδοση δεδομένων, έχουν κάνει αδήριτη την ανάγκη για ακόμα καλύτερους κωδικοποιητές εικόνας που θα προσφέρουν πολύ καλύτερες αναλύσεις εικόνας σε ακόμα μικρότερο μέγεθος.

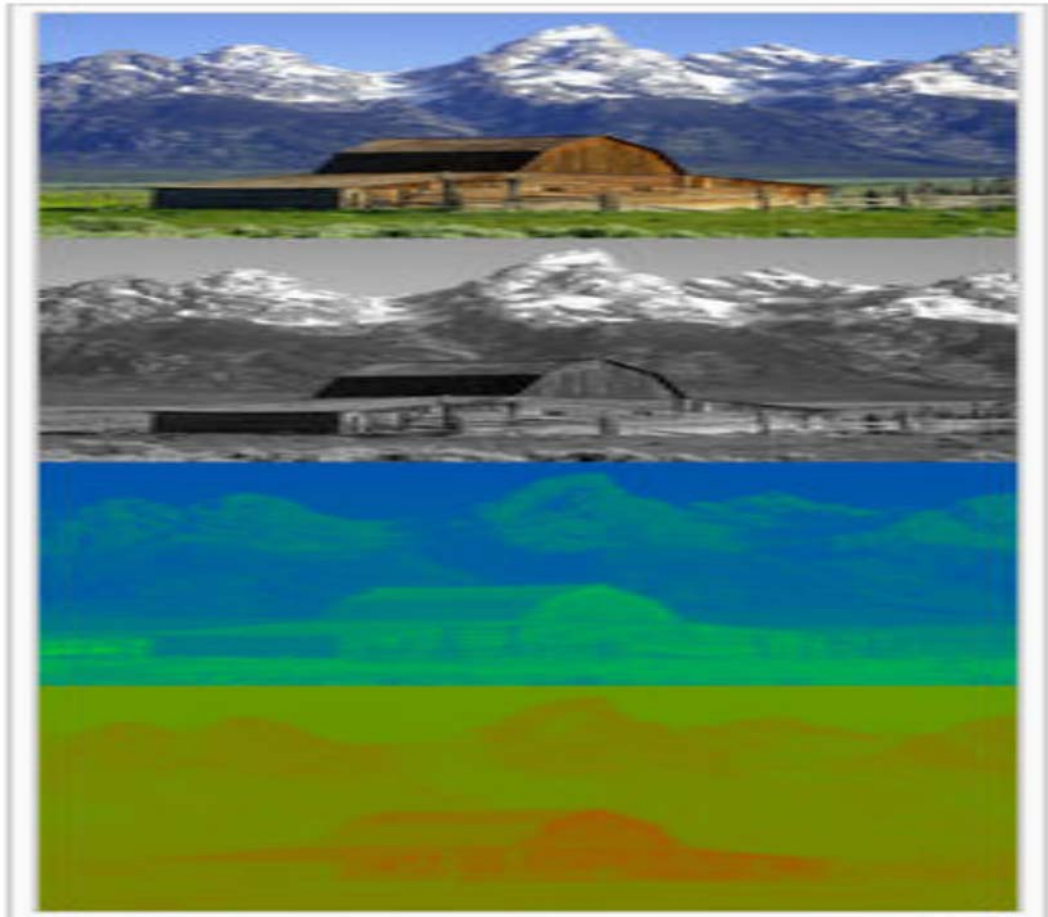
2.3 Πρότυπα

Η έρευνα μας επικεντρώνεται στους ακόλουθους 5 κωδικοποιητές που είναι και οι επικρατέστεροι στις μέρες μας. Τα τελευταία δεκαπέντε χρόνια λοιπόν έχουμε την ανάπτυξη των H264, HEVC, VP8, VP9 και στον νεότερο AV-1. Ιδιαίτερα από το 2010 και μετά βλέπουμε ο τομέας της έρευνας για την κωδικοποίηση της εικόνας να προσελκύει όλες τις μεγάλες εταιρείες από τον χώρο της πληροφορικής και της τηλεόρασης



Εικόνα 2.5: Χρονική Εμφάνιση των Κωδικοποιητών [14].

Υπάρχουν κάποιες βασικές αρχές που διέπουν όλους τους κωδικοποιητές στην διαδικασία της επεξεργασίας της εικόνας; Η απάντηση είναι ναι. Ας προσπαθήσουμε να δώσουμε μια πιο γενική εικόνα της διαδικασίας. Καταρχάς κάθε πλαίσιο της εικόνας διαίρεται, με τεχνικές που διαφέρουν σε κάθε κωδικοποιητή, σε blocks εικονοστοιχείων κάποιας διάστασης. Εκεί έχουμε τρία είδη blocks που αφορούν την ίδια εικόνα. Τα Υ που είναι τα Luma Blocks δηλαδή τα Blocks φωτεινότητας και τα Chroma Blocks U & V (Blue and Red), χρωματικά Blocks.



Εικόνα 2.6: Η εικόνα και από κάτω η απεικόνιση της σύμφωνα με τις συνιστώσες της Y,U,V κατά σειρά [15].

Στην συνέχεια και μέσω των τεχνικών κωδικοποίησης που αναλύσαμε στο προηγούμενο κεφάλαιο, προσπαθούν να δημιουργήσουν συνθήκες πρόβλεψης για τα Blocks. Τέλος χρησιμοποιώντας μια σειρά εργαλείων όπως τον Διακριτό Μετασχηματισμό Συνημίτονου, την κβαντοποίηση και άλλες φτάνουν στο επιθυμητό αποτέλεσμα. Ας δούμε λοιπόν πως λειτουργεί ο κάθε κωδικοποιητής και ποιες διαφορές έχει ο ένας από τον άλλο και με τι αντίκτυπο στα αποτελέσματα.

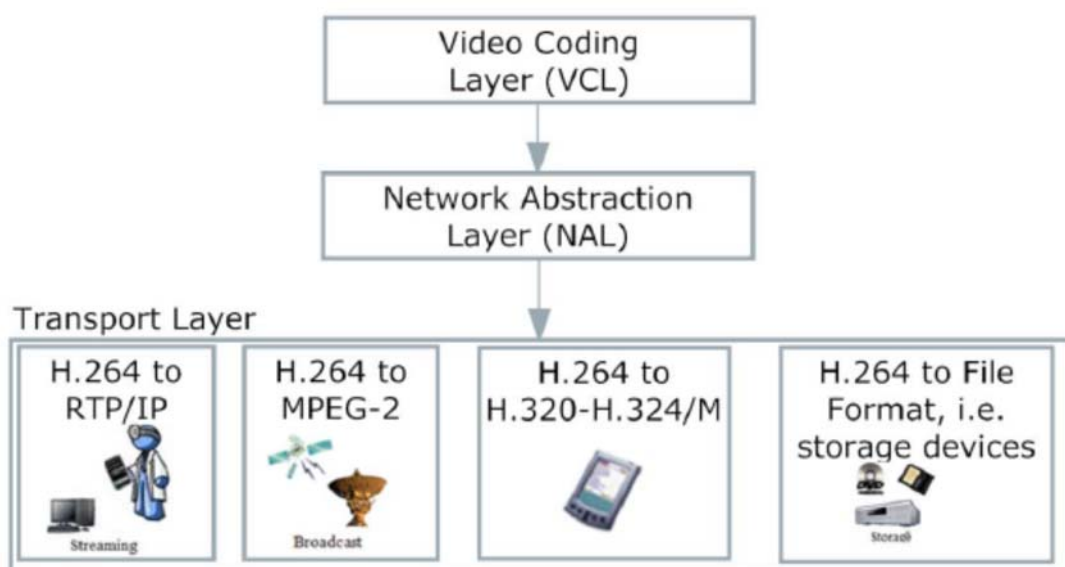
2.3.1 H264

Ένα από τα πλέον διαδεδομένα πρότυπα που προσφέρουν καλύτερη απόδοση στην κωδικοποίηση βίντεο υψηλής ευκρίνειας. Το πρότυπο αυτό είναι μια δημιουργία από την συνεργασία δυο ομάδων το 2003, της ITU-T Video Expert Group (VCEG) και την ISO/IEC Moving Experts Group (MPEG) δίνοντας την ονομασία στην ομάδα αυτή ως Joint Video Team (JVT). Μία νέα καινοτομία που είχε εισάγει το H.264 είναι ο νοητός διαχωρισμός του κωδικοποιητή σε δύο

βασικά στρώματα: το στρώμα κωδικοποίησης εικονοσειρών (VCL - Video Coding Layer) το οποίο είναι υπεύθυνο για την αναπαράσταση και κωδικοποίηση της εικονοσειράς, και το στρώμα δικτύου (NAL - Network Abstraction Layer), που αναλαμβάνει να προσαρμόσει την αναπαράσταση του στρώματος VCL στο μέσο μετάδοσης ή αποθήκευσης που θα χρησιμοποιηθεί [16].

Το στρώμα κωδικοποίησης εικονοσειρών ακολουθεί υβριδική προσέγγιση κωδικοποίησης βασισμένη σε μπλοκ στην οποία κάθε εικόνα αναπαριστάτε από ορθογώνιες επιφάνειες που καλούνται Macroblocks. Δεν υπάρχει συγκεκριμένο στοιχείο της διαδικασίας κωδικοποίησης που στην οποία οφείλεται η βελτίωση στην ικανότητα συμπίεσης, σε σχέση με τα προηγούμενα πρότυπα συμπίεσης. Αντίθετα, οι μικρές βελτιώσεις κάθε χαρακτηριστικού, όλες μαζί, δίνουν το τελικό αποτέλεσμα που κάνει το πρότυπο αυτό αποδοτικό [17].

Το στρώμα δικτύου είναι σχεδιασμένο έτσι ώστε να προσφέρει φιλικότητα προς το δίκτυο και να κάνει απλή και αποδοτική την προσαρμογή της πληροφορίας που προέρχεται από το στρώμα κωδικοποίησης σε μια πλειάδα εφαρμογών και συστημάτων. Τα βασικά στοιχεία του στρώματος αυτού παρουσιάζονται ως εξής: RTP/IP για όλα τα είδη ενσύρματων ή ασύρματων υπηρεσιών πραγματικού χρόνου. Διάφοροι τύποι αρχείων όπως για παράδειγμα ISO MP4 και MMS. H.329 για ενσύρματες και ασύρματες διαδραστικές υπηρεσίες. Συστήματα MPEG2 για υπηρεσίες ευρείας μετάδοσης [18].



Εικόνα 2.7: Τα δυο στρώματα κωδικοποίησης και δικτύου στον H264 [19].

Κατά την κωδικοποίηση το κάθε πλαίσιο του βίντεο επεξεργάζεται σε Macroblocks 16x16. Στο αρχικό στάδιο γίνεται πρόβλεψη του Macroblock. Αυτό μπορεί να γίνει με δυο τρόπους που έχουν ως εξής: intra prediction όπου η πρόβλεψη του macroblock γίνεται με την χρήση κάποιας προηγούμενης κωδικοποίησης ή ακόμα και από το ίδιο το πλαίσιο. Η δεύτερη πρόβλεψη ονομάζεται inter prediction και γίνεται από πλαίσια που έχουν ήδη κωδικοποιηθεί και έχουν ήδη αποσταλεί. Στην συνέχεια ένα σετ από δείγματα μετασχηματίζεται χρησιμοποιώντας ένα 4x4 ή 8x8 μετασχηματισμό ακέραιων παρόμοιο με τον Διακριτό Μετασχηματισμό Συνημίτονου. Ο μετασχηματισμός αυτός παράγει ένα σύνολο από συντελεστές όπου ο καθένας χρησιμοποιείται σαν βάρος στο σετ των δειγμάτων για την πρότυπη βάση της ακολουθίας [20] [21].

Συνδυάζοντας την πρότυπη βάση με βάρη ξαναδημιουργούνται τα δείγματα στο αρχικό σετ. Αφού γίνει ο μετασχηματισμός, η έξοδος του μετασχηματισμού κβαντοποιείται, δηλαδή ο κάθε συντελεστής διαιρείται με μια ακέραια τιμή. Κατά την κβαντοποίηση η ακρίβεια του μετασχηματισμού μειώνεται σύμφωνα με την παράμετρο κβαντοποίησης και έτσι μια μεγάλη τιμή της παραμέτρου σημαίνει και μεγαλύτερη συμπίεση, άρα μείωση της ποιότητας της εικόνας [22].

2.3.2 HEVC

Το πρότυπο High Efficiency Video Coding (HEVC) είναι το πιο πρόσφατο έργο (2013) της Joint Collaborative Team on Video Coding (JCT-VC), δηλαδή την από κοινού συνεργασία της ITU-T VCEG και της ISO/IEC MPEG [23]. Ο κύριος στόχος της προσπάθειας του προτύπου HEVC είναι να αποφέρει σημαντικά βελτιωμένη απόδοση συμπίεσης σε σχέση με τα υπάρχοντα πρότυπα που θα αγγίζει τη μείωση κατά 50% του ρυθμού δεδομένων για ίση ποιότητα βίντεο [24]. Η αυξανόμενη δημοτικότητα των βίντεο υψηλής ευκρίνειας (High Definition - HD) και η απαίτηση για αναλύσεις μεγαλύτερες από HD (π.χ. 4kx2k ή 8kx4k ανάλυση) δημιουργούν ακόμη πιο ισχυρή ανάγκη για αποδοτικότερη κωδικοποίηση, ανώτερη της δυνατότητας του H.264/MPEG-4 AVC. Η ανάγκη αυτή είναι ακόμη μεγαλύτερη όταν η υψηλότερη ανάλυση συνοδεύεται από multiview απεικόνιση [25]. Επιπλέον, η κίνηση δεδομένων που προκαλείται από εφαρμογές βίντεο που στοχεύουν κινητές συσκευές, ταμπλέτες, υπολογιστές, καθώς και οι ανάγκες μετάδοσης για video-on-demand υπηρεσίες, επιβάλλουν σοβαρές προκλήσεις στα σημερινά δίκτυα [26]. Η αυξημένη επιθυμία για υψηλότερη ποιότητα και ανάλυση προκύπτει επίσης και σε κινητές εφαρμογές [27]. Το HEVC έχει σχεδιαστεί για να αντιμετωπίσει ουσιαστικά όλες τις υπάρχουσες εφαρμογές του H.264/MPEG-4 AVC και να επικεντρωθεί κυρίως σε δύο βασικά

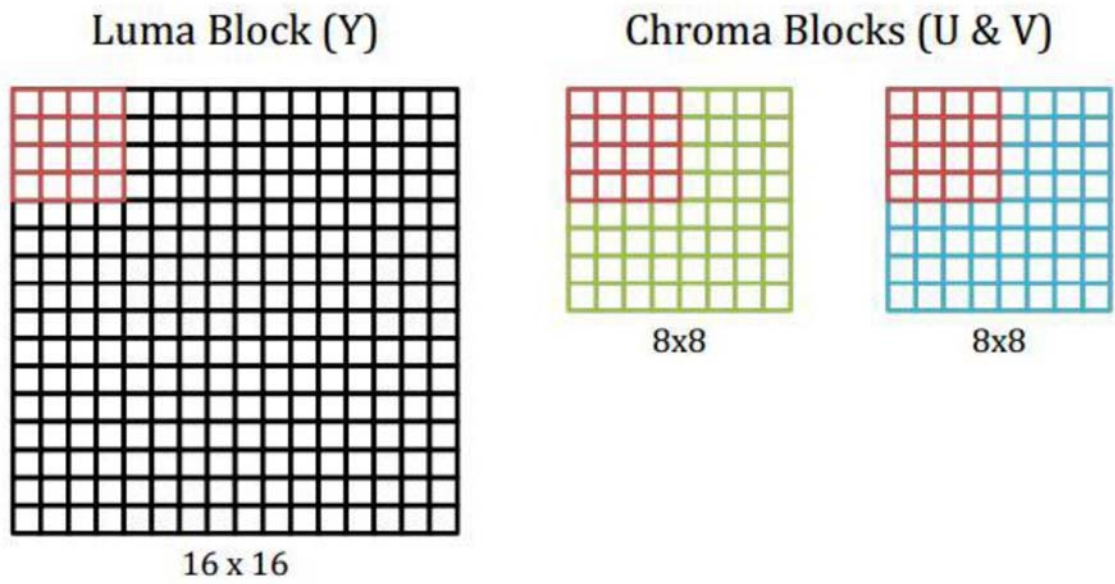
ζητήματα: την αυξημένη ανάλυση βίντεο και την αυξημένη χρήση των παράλληλων αρχιτεκτονικών επεξεργασίας [28].

Πώς μπορεί το HEVC να κωδικοποιεί αρχεία βίντεο δύο φορές πιο αποτελεσματικά απ' όσα τα προηγούμενα πρότυπα κωδικοποίησης [29]; Το μεγαλύτερο μέρος της ισχύος των προτύπων συμπίεσης βίντεο προέρχεται από μια τεχνική γνωστή ως motion compensated prediction. Τα blocks των pixels που κωδικοποιούνται κάνοντας αναφορά σε μια άλλη περιοχή στο ίδιο πλαίσιο (intra reference) ή σε άλλο πλαίσιο (inter reference). Ενώ το H.264/AVC ορίζει macroblocks έως και 16×16 pixels, το HEVC μπορεί να περιγράψει μια πολύ μεγαλύτερη ποικιλία μεγεθών blocks, έως και 64×64 pixels. Το HEVC επιτρέπει στα blocks να κωδικοποιηθούν σε διάφορα μεγέθη σε σχέση με το υπολειπόμενο σφάλμα. Κάθε μονάδα κωδικοποίησης ανώτατου επιπέδου (CTU) πρώτα κωδικοποιείται ως πρόβλεψη quad-tree, όπου σε κάθε βάθος ο κωδικοποιητής αποφασίζει για κωδικοποίηση με συγχώνευση/παράλειψη, inter ή intra κωδικοποίηση. Το υπόλοιπο από αυτές τις προβλέψεις στη συνέχεια κωδικοποιείται με ένα δεύτερο quad-tree το οποίο προαιρετικά μπορεί να έχει μεγαλύτερο βάθος από ότι η πρώτη πρόβλεψη quad-tree. Για παράδειγμα, αυτό επιτρέπει στο υπολειπόμενο σφάλμα από 32×32 inter coded κωδικοποίηση μονάδας (CU) να εκπροσωπηθεί από ένα μείγμα 16×16 , 8×8 και 4×4 μετασχηματισμών. Επίσης, το HEVC μπορεί να κωδικοποιήσει τα διανύσματα κίνησης με πολύ μεγαλύτερη ακρίβεια, δίνοντας μια καλύτερη πρόβλεψη των blocks με λιγότερο υπολειπόμενο σφάλμα. Υπάρχουν 35 κατευθύνσεις intra picture, σε σύγκριση με τις 9 μόνο κατευθύνσεις για το H.264/AVC. Επιπλέον, το HEVC περιλαμβάνει το μηχανισμό adaptive motion vector prediction, μια νέα μέθοδο για τη βελτίωση του inter prediction [8].

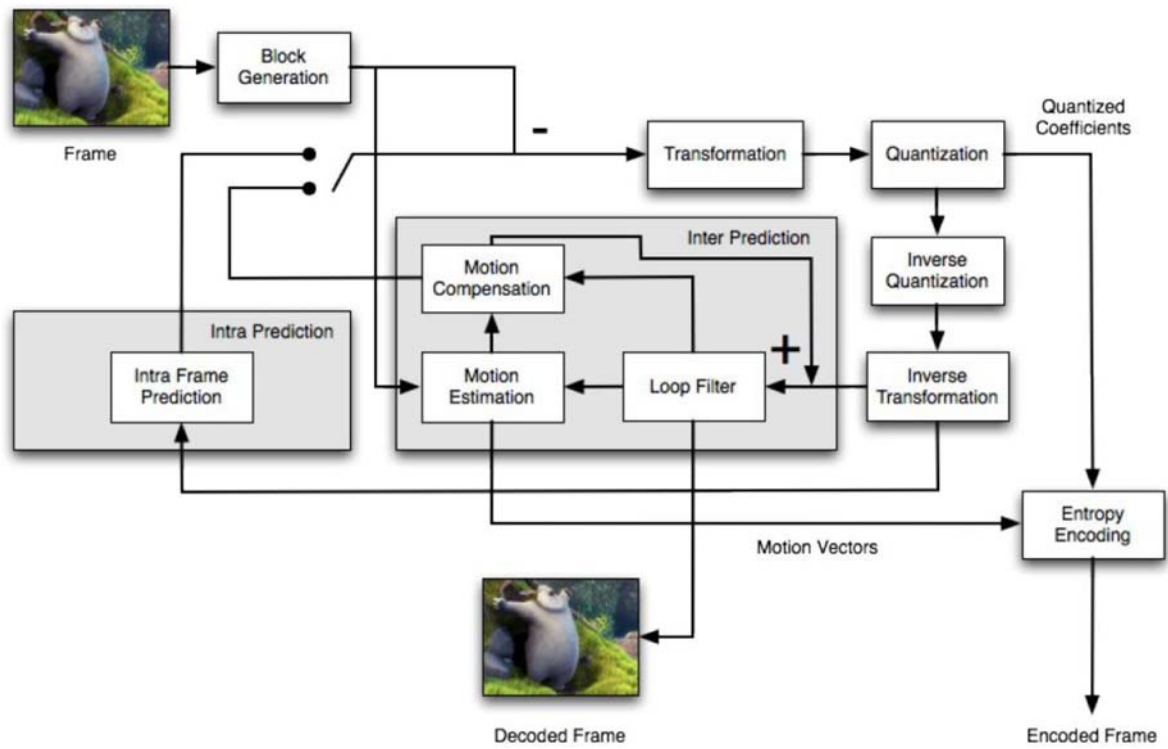
2.3.3 VP8

Το VP8 είναι μιας υψηλής απόδοσης τεχνολογία συμπίεσης βίντεο που αναπτύχθηκε από την εταιρία On2 Technologies. Η Google εξαγόρασε την On2 τον Φεβρουάριο του 2010. Το VP8 είναι ανοιχτού κώδικα και η υλοποίηση αναφοράς, που ονομάζεται libvpx, διατέθηκε το 2010 από την Google και έχει άδεια χρήσης τύπου BSD. Η επιλογή αυτού του τύπου άδειας χρήσης από την Google δεν ήταν τυχαία καθώς τον κάνει ιδανικό για εφαρμογές διαδικτύου. Για αυτό η σχεδίαση του είναι βελτιστοποιημένη για τέτοιου είδους εφαρμογές [30].

Το VP8 χρησιμοποιεί για την συνιστώσα της φωτεινότητας (Y) block μεγέθους 16×16 και για τις συνιστώσες της χρωματικότητας (U, V) block μεγέθους 8×8 .



Εικόνα 2.8: Luma και Chroma Blocks στον VP8 [30].



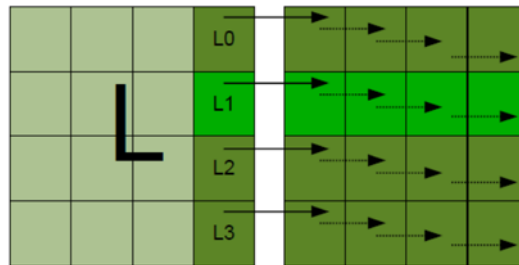
Εικόνα 2.9: Διάγραμμα Ροής στον VP8 [30].

Intra Frame Prediction:

Για τη φωτεινότητα (luma) έχουμε 4 επιλογές:

H PRED

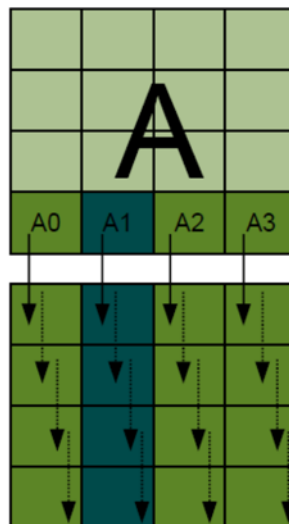
Συμπληρώνεται κάθε στήλη με ένα αντίγραφο της συνορεύουσας προς τα αριστερά στήλης.



Εικόνα 2.10: Οριζόντια Πρόβλεψη [30].

V PRED

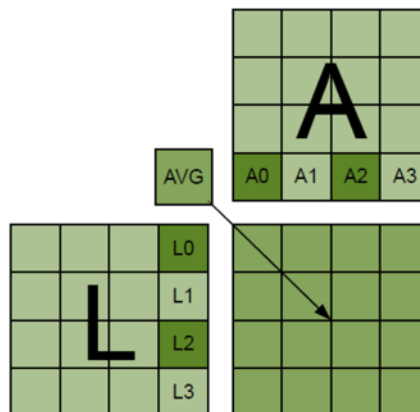
Συμπληρώνεται κάθε γραμμή με ένα αντίγραφο της συνορεύουσας προς τα πάνω γραμμής.



Εικόνα 2.11: Κάθετη Πρόβλεψη [30].

DC_PRED

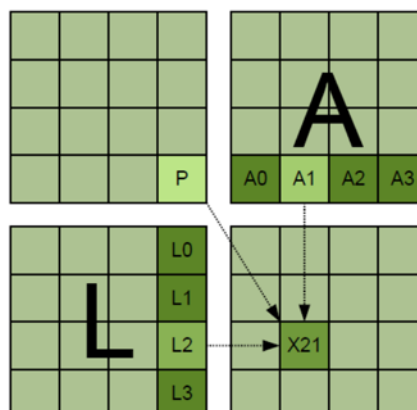
Χρησιμοποιείται ο μέσος όρος των pixel που βρίσκονται αριστερά και πάνω από το μπλοκ. Αν το block βρίσκεται στην ανώτερη γραμμή του καρέ, χρησιμοποιείται ο μέσος όρος των αριστερών pixels. Αν το block βρίσκεται στην αριστερότερη στήλη του καρέ χρησιμοποιείται ο μέσος όρος των ανώτερων pixels.



Εικόνα 2.12: DC Pred Πρόβλεψη [30].

TM_PRED

Κάθε 16x16 block χωρίζεται σε 16 4x4 block τα οποία προβλέπεται ανεξάρτητα, δηλαδή για καθένα από αυτά τα 4 μπορεί να χρησιμοποιηθεί οποιοδήποτε από τα διαθέσιμα B_PRED modes πρόβλεψης.



Εικόνα 2.13: TM Pred Πρόβλεψη [30].

Motion Estimation:

Υπολογίζονται μέσω εκτίμησης κίνησης τα διανύσματα κίνησης μόνο των block φωτεινότητας (Luma blocks). Τα διανύσματα κίνησης των block χρωματικότητας (Chroma blocks) παράγονται από τα block φωτεινότητας. Χρησιμοποιούνται διανύσματα κίνησης για block μεγέθους 16x16, 16x8, 8x16, 8x8 και 4x4. Έχουμε ανάλυση $\frac{1}{4}$ του pixel για φωτεινότητα και $\frac{1}{8}$ του pixel για χρωματικότητα.

Χρησιμοποιούνται οι ακόλουθοι τύποι διανυσμάτων κίνησης (Motion Vectors):

Τύπος Διανύσματος Κίνησης	Τι εκτελεί
MV NEAREST	Επαναχρησιμοποίηση μη-μηδενικού motion vector του τελευταίου αποκωδικοποιημένου block.
MV NEAR	Επαναχρησιμοποίηση μη-μηδενικού motion vector του προ-τελευταίου αποκωδικοποιημένου block.
MV ZERO	Το block βρίσκεται στην ίδια θέση.
MV NEW	Δημιουργία νέου Motion Vector.
MV SPLIT	Το block χωρίζεται σε sub-block με το κάθε ένα από αυτά να έχει το δικό του motion vector. Χρήσιμο για αντικείμενα εντός του block με διαφορετικά χαρακτηριστικά κίνησης.

Στην συνέχεια έχουμε την λειτουργία του Motion Compensation, δηλαδή την εφαρμογή των motion vectors στο προηγούμενο frame για την δημιουργία ενός «προβλέψιμου» καρέ. Αρκεί να μεταδοθεί η διαφορά μεταξύ «προβλέψιμου» και πραγματικού καρέ.

Inter Frame Prediction:

Χρησιμοποιώντας τα καρέ αναφοράς και τα motion vectors, προβλέπονται τα ενδιάμεσα (inter) καρέ. Σε αντίθεση με το H264, δεν έχουμε B frames.

Τρόπος ενδιάμεσης πρόβλεψης καρέ

Χρησιμοποιώντας κάποιο από τα ακόλουθα είδη καρέ:

Previous Frame: Είναι το πιο πρόσφατο πλήρως αποκωδικοποιημένο καρέ.

Alternate Reference Frame: Το εναλλακτικό αυτό frame μπορεί να είναι αόρατο. Μεταφέρει συμπληρωματική πληροφορία είτε για πρόβλεψη μείωσης θορύβου είτε για πρόβλεψη με κατεύθυνση στο παρελθόν ή στο μέλλον. Στην 2η περίπτωση, σε συνδυασμό με τα Golden Frames, αντισταθμίζει την έλλειψη B-frames (το Alternate Reference Frame μεταδίδεται με πληροφορία από κάποιο μελλοντικό frame).

Golden Reference Frame: Καρέ αναφοράς της εικόνας του τρέχοντος καρέ. Μπορεί να ενημερωθεί τμήμα του κάθε φορά. Χρήσιμο για ανάκτηση λαθών και για αλλαγή σκηνών.

Transformation:

Χρησιμοποιείται υβριδικός μετασχηματισμός που συνδυάζει τον διακριτό μετασχηματισμό συνημίτονου και το μετασχηματισμό Walsh-Hadamard.

Τέλος να επισημάνουμε ότι η κβαντοποίηση των συντελεστών του μετασχηματισμού DC είναι κλιμακωτή και βελτιστοποιημένη για ποιότητα μεταξύ 30dB to ~45dB SNR. Οι κβαντοποιημένοι συντελεστές διαβάζονται με zig-zag σειρά. Επίσης χρησιμοποιείται Adaptive Quantization με συνέπεια το καρέ να μπορεί να χωριστεί σε 4 τμήματα το πολύ και για το καθένα εξ αυτών να εφαρμοσθεί διαφορετικό σεντ παραμέτρων κβαντοποίησης.

2.3.4 VP9

Το VP9 είναι ο διάδοχος του VP8 (2010) και ένα μεγάλο κομμάτι στην εξέλιξή του έγινε μέσα από την φυσική εξέλιξη και πρόοδο. Η βελτίωση της επίδοσης ενός μεγάλου κομματιού του κώδικα έχει επιφέρει μεγάλες βελτιώσεις στον VP9. Ένα παράδειγμα είναι η επέκταση του block

πρόβλεψης το οποίο εφαρμόζεται στον VP9 είναι τα super blocks με μέγεθος 64x64 και επιτρέπεται να χωρίζεται με recursive decomposition σε 4x4 block.

Η πρόγνωση γίνεται μέσα από τέσσερις επιλογές. Στις τρεις από αυτές αναπαριστώνται ως τελικά σημεία έτσι ώστε να μην μπορούν να χωριστούν άλλο και περιλαμβάνουν, μεταχειρίζονται τον κύριο block ως ένα ενιαίο block ή ως δύο κάθετα τριγωνικά block ή ως δύο οριζόντια τριγωνικά block. Υπάρχουν συνολικά 13 διαφορετικά μεγέθη block για σκοπούς αναγνώρισης περισσότερων σημάτων πρόβλεψης: 64x64, 64x32, 32x64, 32x32, 32x16, 16x32, 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 και 4x4 [31].

Επίσης το VP9 υποστηρίζει 10 Intra prediction modes για τα block από 4x4 μέχρι 32x32. Αυτά είναι το DC PRED, TM PRED (True motion prediction), H PRED (Horizontal prediction), V PRED (Vertical prediction) και επιπλέον έξι λοξός modes. Αυτά τα modes είναι το D27, D153, D135, D117, D63, D45 τα οποία αντιστοιχούν στις εξής γωνίες 27,153,135,117,63 και 45.

Ο VP9 υποστηρίζει ένα σετ από 4 inter prediction modes για block μεγέθους από 4x4 μέχρι 64x64 pixels. Τα τέσσερα αυτά modes είναι: NEARESTMV, NEARMV, ZEROMV και NEWMV. Η περιγραφή του πως συμπεριφέρεται το κάθε mode είναι η ίδια με αυτή που αναφέραμε πιο πάνω για το VP8 [32].

2.3.5 AV-1

Ο νεώτερος (2018) και ίσως ο πιο σημαντικός κωδικοποιητής. Μια ματιά στην πληθώρα των εταιρειών που συμμετέχουν στην ανάπτυξη του, μας πείθει αμέσως.



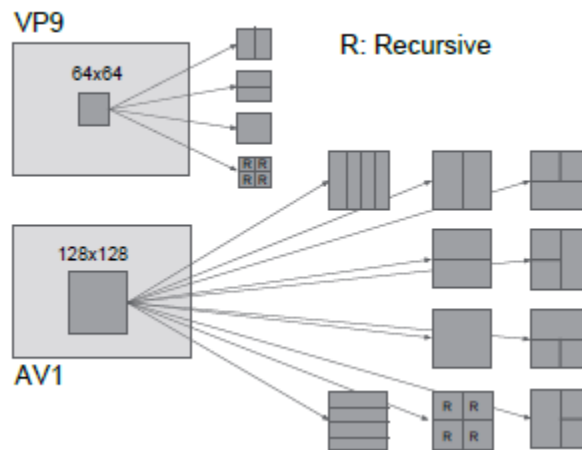
Εικόνα 2.14: Μέλη της AOMedia, δημιουργού του AV-1 [33].

Ας δούμε τις διαφορές που έχει στην κωδικοποίηση από τους VP9 και HEVC.

Coding Block Partition:

Το VP9 χρησιμοποιεί ένα δέντρο κατατμήσεων 4 κατευθύνσεων ξεκινώντας από το επίπεδο 64x64 μέχρι και το επίπεδο 4x4, με κάποιους πρόσθετους περιορισμούς για τα μπλοκ 8x8 και κάτω όπως φαίνεται στο σχήμα 3.11. Σημειώστε ότι για τα διαμερίσματα που χαρακτηρίζονται ως R αναφερόμαστε ως επαναλαμβανόμενα ως προς το ίδιο δένδρο κατατμήσεων που επαναλαμβάνεται σε χαμηλότερη κλίμακα μέχρι να φτάσουμε στο χαμηλότερο επίπεδο 4x4. Το AV1 όχι μόνο επεκτείνει το δέντρο κατατμήσεων σε δομή 10 διαδρομών όπως φαίνεται στο ίδιο σχήμα, αλλά αυξάνει και το μεγαλύτερο μέγεθος (αναφέρεται όπως ως superblock στο VP9 / AV1 parlance) για να ξεκινήσει από 128x128. Σημείωση ότι αυτό περιλαμβάνει 4:1/1:4

ορθογώνια διαμερίσματα που δεν υπήρχαν στο VP9. Κανένα από τα ορθογώνια χωρίσματα δεν μπορεί να υποδιαιρεθεί περαιτέρω. Επιπλέον, το AV1 προσθέτει περισσότερη ευελιξία στη χρήση διαμερισμάτων κάτω από το επίπεδο του 8x8, με την έννοια ότι τώρα η πρόβλεψη χρωμάτων σε επίπεδο 2x2 είναι δυνατή σε ορισμένες περιπτώσεις [34].



Εικόνα 2.15: Partition Tree σε VP9 και AV-1 [34].

Intra Prediction:

Το VP9 υποστηρίζει 10 intra prediction modes, μεταξύ των οποίων 8 ως προς την κατεύθυνση που αντιστοιχούν σε γωνίες από 45 έως 207 μοίρες και 2 μη κατευθυντικές τις: DC και True Motion (TM) Mode. Στον AV1 φτάνουμε στα 56 intra prediction modes κατεύθυνσης και προσθέτει και 3 μη κατεύθυνσης. Ενσωματώνει την νέα τεχνολογία Recursive Filtering Based Intra Predictor και πλέον βλέπει τα Luma Blocks σε δύο διαστάσεις με την βοήθεια της διεργασίας Markov. Προσθέτει επίσης την τεχνολογία Chroma Predicted from Luma και την Color Palette as Predictor που φτιάχνει μια εσωτερική διεργασία, ένα πρόσθετο εργαλείο, μια δική του παλέτα βασικών χρωμάτων ως ένα εργαλείο επιπλέον εσωτερικής κωδικοποίησης. Τέλος εισάγει την τεχνολογία Intra Block Copy που μπορεί και λειτουργεί ως Inter Prediction, μπορεί δηλαδή να ανατρέξει σε ήδη δομημένα Block μέσα στο ίδιο πλαίσιο.

Inter Prediction:

Το AV1 επεκτείνει τον αριθμό των αναφορών για κάθε πλαίσιο από 3 έως 7. Εκτός από το LAST του VP9 (πλησιέστερο παρελθόν), καρέ GOLDEN (μακρινό παρελθόν) και ALTREF (χρονικό

φιλτραρισμένο μέλλον), προσθέτουμε δύο κοντινά πλαίσια (LAST2 και LAST3) και δύο μελλοντικά πλαίσια (BWDREF και ALTREF2).

Δυναμική αναπαράσταση διανυσματικού και χρονικού εύρους κίνησης:

Η αποδοτική κωδικοποίηση διανύσματος κίνησης (MV) είναι ζωτικής σημασίας για έναν κωδικοποιητή βίντεο επειδή παίρνει ένα μεγάλο μέρος του κόστους του τέλους για τα inter πλαίσια. Για το σκοπό αυτό, το AV1 ενσωματώνει μια εξελιγμένη επιλογή αναφοράς MV για να αποκτήσετε καλές αναφορές MV για ένα δεδομένο μπλοκ με αναζήτηση τόσο των χωρικών όσο και των χρονικών υποψηφίων. Το AV1 όχι μόνο αναζητά βαθύτερα χωρική γειτονιά από την VP9 για την κατασκευή μιας υποψήφιας περιοχής χώρου, αλλά χρησιμοποιεί επίσης ένα μηχανισμό εκτίμησης χρονικού πεδίου κίνησης για να δημιουργούν χρονικούς υποψηφίους. Η διαδικασία εκτίμησης πεδίου κίνησης λειτουργεί σε τρία στάδια: motion vector buffering, motion trajectory creation, και motion vector projection.

Επικαλυπτόμενη αντιστάθμιση κίνησης μπλοκ (OBMC):

Το OBMC μπορεί να μειώσει σε μεγάλο βαθμό τα σφάλματα πρόβλεψης κοντά στις άκρες των μπλοκ με ομαλή λειτουργία συνδυάζοντας προβλέψεις που δημιουργήθηκαν από παρακείμενα διανύσματα κίνησης. Στο AV1, ένας αλγόριθμος αλληλοεπικάλυψης δύο όψεων έχει σχεδιαστεί για να καταστήσει το OBMC εύκολα προσαρμόσιμο στο προηγμένο πλαίσιο διαμέρισης. Προοδευτικά συνδυάζει την πρόβλεψη βάσει μπλοκ με τους δευτερεύοντες παράγοντες πρόβλεψης στην άνω άκρη και στη συνέχεια στα αριστερά, εφαρμόζοντας προκαθορισμένο 1-D φίλτρα εξομάλυνσης σε κάθετες και οριζόντιες διευθύνσεις. Οι δευτερεύοντες προγνωστικοί παράγοντες λειτουργούν μόνο σε περιορισμένες επικαλυπτόμενες περιοχές στο Block έτσι ώστε να μην μπερδεύονται μεταξύ τους στην ίδια πλευρά. Το AV1 OBMC είναι ενεργοποιημένο μόνο για Block χρησιμοποιώντας ένα ενιαίο πλαίσιο αναφοράς και λειτουργεί μόνο με την πρώτη πρόβλεψη οποιουδήποτε γείτονα με δύο πλαίσια αναφοράς, επομένως το εύρος ζώνης μνήμης στην χειρότερη περίπτωση θα είναι το ίδιο με έναν παραδοσιακό σύνθετο πρόγνωσης.

Οι Warped Motion Compensation και Advanced Compound Prediction φέρνουν ένα νέο πρόσθετο στο Inter Prediction, βοηθώντας στην ανάλυση της κίνησης της κάμερας και δημιουργώντας ένα νέο εργαλείο (συλλογή από compound prediction tools) που κάνει το Inter Prediction πιο ευέλικτο.

Transform Coding:

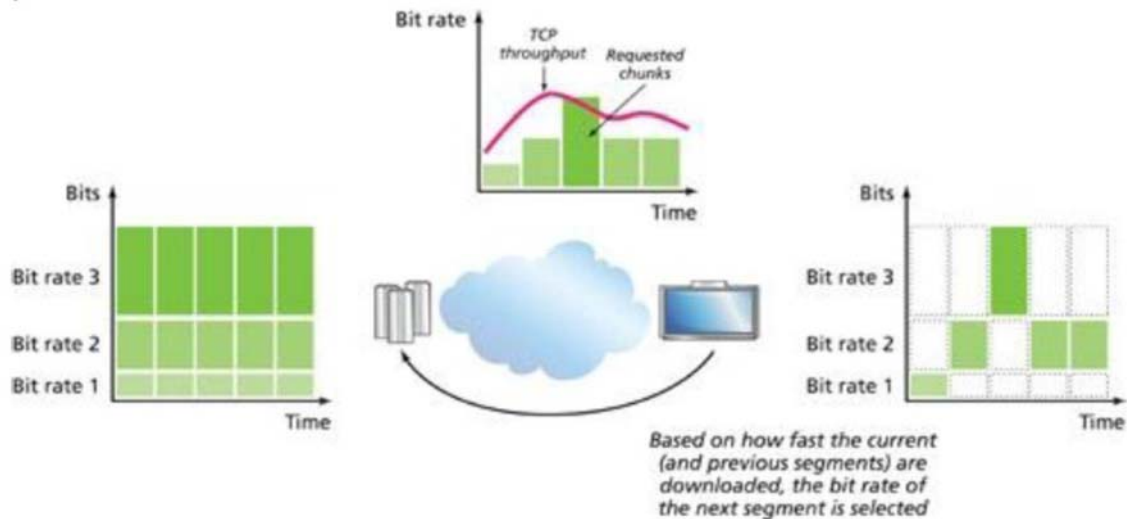
Περνάμε από τον σταθερού μεγέθους μετασχηματισμό του VP9 σε έναν ευέλικτο μετασχηματισμό πολλαπλών μεγεθών στα Inter Coding Luma Blocks, από 4x4 έως 64x64.

Τέλος εισάγει δυο νέες τεχνολογίες, τις Multi Symbol Entropy Coding που περνάει από έναν tree-based boolean non-adaptive binary arithmetic encode τρόπο που χρησιμοποιεί ο VP9 σε ένα δικό του τρόπο symbol-to-symbol adaptive multi-symbol arithmetic coder, και τέλος το In-Loop Filtering Tools and Post-processing που επιτρέπει στον AV-1 την εφαρμογή διαδοχικών εργαλείων φιλτραρίσματος εντός του βρόγχου σε ένα αποκωδικοποιημένο πλαίσιο.

2.4 HTTP Adaptive Streaming

Τα adaptive bitrate streaming συστήματα λειτουργούν ανιχνεύοντας το εύρος ζώνης του χρήστη καθώς επίσης και την χωρητικότητα της CPU του σε πραγματικό χρόνο έτσι ώστε να μπορούν να προσαρμόσουν ανάλογα την ποιότητα του βίντεο που θα μεταδοθεί. Επομένως το βίντεο κωδικοποιείται σε διαφορετικές μορφές και διαφορετικές ποιότητες. Τα HTTP adaptive streaming συστήματα μεταφέρουν τα δεδομένα τους μέσω του πρωτοκόλλου HTTP.

Ένα βίντεο χωρίζεται σε μικρά μη-επικαλυπτόμενα διαστήματα σε μια ακολουθία, μήκους μεταξύ ενός και δέκα δευτερολέπτων και κάθε χρονικό διάστημα κωδικοποιείται σε διαφορετικά bit rates. Ο χρήστης ενός HTTP adaptive streaming συστήματος ζητά τα κομμάτια αυτά μέσω του HTTP και συνήθως το πρώτο κομμάτι στέλνεται σε ένα χαμηλό bit rate. Εάν τα κομμάτια παραδίδονται συστηματικά σε χρόνο μικρότερο από το μήκος του διαστήματος, ο Αλγόριθμος Προσδιορισμού του ρυθμού (RDA) στον πελάτη, επιλέγει έναν υψηλότερο ρυθμό bit για το επόμενο κομμάτι [35] [36] [37].



Εικόνα 2.16: Μεθοδολογία HTTP Adaptive Streaming [19].

2.4.1 Το Πρωτόκολλο MPEG-DASH

Το MPEG-DASH (Dynamic Adaptive Streaming over HTTP) είναι ένα πρότυπο για adaptive streaming μέσω HTTP. Σε αυτό το πρότυπο το περιεχόμενο των multimedia συλλέγεται και αποθηκεύεται σε έναν HTTP server και μεταδίδεται μέσω HTTP. Το περιεχόμενο αυτό αποθηκεύεται σε δύο μέρη τα οποία έχουν ως εξής: το πρώτο μέρος είναι το Media Presentation Description (MPD) και το δεύτερο μέρος είναι τα segments. Το MPD περιγράφει το διαθέσιμο περιεχόμενο, τις διάφορες εναλλακτικές λύσεις όσο αφορά το βίντεο που θα γίνει stream, τις διευθύνσεις URL που βρίσκεται το συγκεκριμένο βίντεο καθώς επίσης και διάφορα άλλα χαρακτηριστικά. Όσο αφορά τα segments, περιέχουν τα πραγματικά multimedia bitstreams σε μορφή τεμαχίων, τα οποία μπορεί να βρίσκονται σε μονά ή πολλαπλά αρχεία [38].

Για να μπορεί να αναπαραχθεί το περιεχόμενο, το πρώτο και βασικό βήμα είναι ο χρήστης του DASH να αποκτήσει το MPD. Υπάρχουν πολλά είδη μεταφορών μέσω των οποίων το MPD μπορεί να παραδοθεί στον χρήστη, όπως για παράδειγμα μέσω HTTP, thumb drive και broadcast. Στη συνέχεια το MPD αναλύεται και ο χρήστης παίρνει πληροφορίες για την διαθεσιμότητα του media-content, τους τύπους των media, τα resolutions, το μέγιστο και ελάχιστο εύρος ζώνης, τις διάφορες εναλλακτικές κωδικοποιήσεις του περιεχομένου καθώς επίσης και διάφορα άλλα χαρακτηριστικά του περιεχομένου. Χρησιμοποιώντας αυτές τις

πληροφορίες, επιλέγεται η κατάλληλη εναλλακτική κωδικοποίηση και ξεκινά το streaming του περιεχομένου ανακτώντας τα τμήματα μέσω HTTP GET requests [39].

Ακολούθως, ο πελάτης ανακτά τα επόμενα τμήματα και παρακολουθεί ταυτόχρονα τις διακυμάνσεις στο εύρος ζώνης του δικτύου. Ανάλογα με αυτές τις διακυμάνσεις, αποφασίζεται πώς το βίντεο μπορεί να προσαρμοστεί στο διαθέσιμο εύρος ζώνης, ανακτώντας τμήματα διαφορετικών εναλλακτικών λύσεων όπως για παράδειγμα χαμηλότερων ή υψηλότερων bitrates [40] [41] [42].

2.4.1.1 Media Presentation Description

Το Dynamic HTTP streaming για να γίνει απαιτεί να υπάρχουν στον server διαθέσιμες εναλλακτικές λύσεις, για το περιεχόμενο, για διαφορετικά bit rates. Επίσης το περιεχόμενο μπορεί να αποτελείται από πολλά συστατικά, όπως για παράδειγμα βίντεο ή ήχος, καθένα από τα οποία μπορεί να έχει διαφορετικά χαρακτηριστικά. Στο MPEG-DASH τα χαρακτηριστικά αυτά περιγράφονται μέσω του MPD το οποίο είναι ένα XML έγγραφο.

Το MPD αποτελείται από μια ή περισσότερες περιόδους, όπου κάθε περίοδος είναι ένα χρονικό διάστημα κατά την διάρκεια του χρονικού άξονα. Κάθε περίοδος έχει τον δικό της χρόνο έναρξης, την διάρκεια της και αποτελείται από ένα ή περισσότερα adaptation sets. Το κάθε adaptation παρέχει πληροφορίες για ένα ή περισσότερα συστατικά και τις εναλλακτικές κωδικοποιήσεις για τα συστατικά αυτά. Κάθε adaptation set περιλαμβάνει συνήθως πολλαπλές αναπαραστάσεις.

Μια αναπαράσταση είναι ουσιαστικά μια κωδικοποιημένη εναλλακτική του ίδιου συστατικού. Η αναπαράσταση αυτή διαφέρει από τις άλλες ως προς το εύρος ζώνης, το resolution, τον αριθμό των καναλιών ή και άλλα πολλά χαρακτηριστικά. Κάθε αναπαράσταση αποτελείται από ένα ή πολλά segments. Το κάθε segment έχει ένα URL το οποίο είναι μια διευθυνσιοδοτημένη θέση στον server, το οποίο μπορεί να κατέβει μέσω HTTP GET request.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
-<!--
  MPD file Generated with GPAC version 0.5.1-DEV-rev5619 on 2017-05-17T09:48:26Z
-->
- <MPD minBufferTime="PT1.500000S" type="static" mediaPresentationDuration="PT0H0M8.33S" profiles="urn:mpeg:dash:profile:full:2011">
- <ProgramInformation moreInformationURL="http://emcvideoz.cs.ucy.ac.cy/dash.js-master/samples/dash-if-reference-player/dashed/SingleDashed/incident/">
- <Title>
  /home/ele2016/SingleDashed/incident_10c_720x576_30/incident_10c_720x576_30_2k.mpd generated by GPAC
- <Title>
- <ProgramInformation>
- <BaseURL>
  http://emcvideoz.cs.ucy.ac.cy/dash.js-master/samples/dash-if-reference-player/dashed/SingleDashed/incident/
- <BaseURL>
- <Period duration="PT0H0M8.33S">
- <AdaptationSet segmentAlignment="true" maxWidth="720" maxHeight="576" maxFrameRate="30" par="5:4" lang="und">
- <Representation id="1" mimeType="video/mp4" codecs="avc1.64001f" width="720" height="576" frameRate="30" sar="1:1" startWithSAP="1" bandwidth="1781282">
- <SegmentList timescale="15360" duration="42664">
  <Initialization sourceURL="incident_10c_2k_1440x1080init.mp4"/>
  <SegmentURL media="incident_10c_2k_1440x10801.m4s"/>
  <SegmentURL media="incident_10c_2k_1440x10802.m4s"/>
  <SegmentURL media="incident_10c_2k_1440x10803.m4s"/>
- <SegmentList>
- <Representation>
- <AdaptationSet>
- <Period>
- </MPD>
```

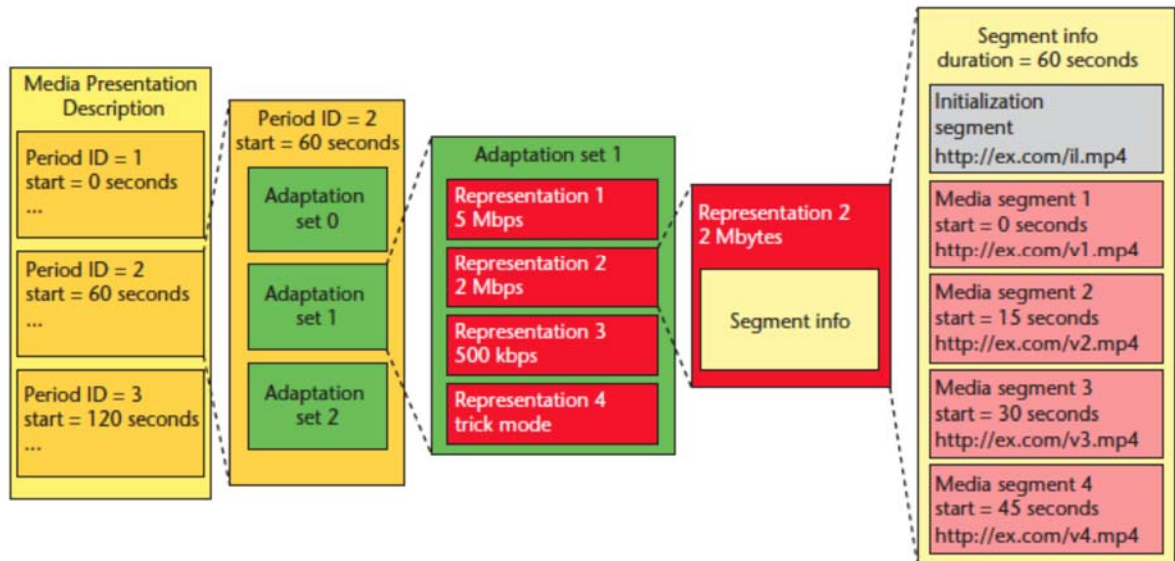
Εικόνα 2.17: XML αρχείο ενός Media Presentation Description [38].

2.4.1.2 Video Segments

Το περιεχόμενο είναι προσβάσιμο μέσω μιας συλλογής από segments. Ένα segment ορίζεται ως το σώμα της απόκρισης στο HTTP GET του DASH client. Ένα συστατικό είναι κωδικοποιημένο και μοιράζεται σε πολλά segments. Το πρώτο segment δεν περιλαμβάνει καθόλου πραγματικά δεδομένα για το πολυμέσο. Είναι συνήθως ένα segment αρχικοποίησης που περιλαμβάνει τις απαιτούμενες πληροφορίες για το πώς το ο αποκωδικοποιητής του DASH θα εκκίνηση στην μεριά του client.

Στη συνέχεια το stream διαιρείται σε ένα ή περισσότερα συνεχόμενα segments όπου το καθένα έχει ένα μοναδικό URL, ένα δείκτη, ώρα έναρξης και διάρκεια. Επίσης κάθε segment περιέχει τουλάχιστο ένα σημείο πρόσβασης το οποίο μπορεί να είναι τυχαίας προσπέλασης ή switch-to-point στο stream και η αποκωδικοποίηση μπορεί να αρχίσει χρησιμοποιώντας δεδομένα μόνο από αυτό το σημείο, προς τα εμπρός [38].

Για να είναι επιτρεπτό το κατέβασμα αυτών των segments σε πολλαπλά μέρη, καθορίζεται μια μέθοδος για σηματοδότηση των sub segments χρησιμοποιώντας ένα index box. Αυτό το index box περιγράφει τα sub segments καθώς και σημεία πρόσβασης του stream στο segment, σηματοδοτώντας της διάρκεια και το byte offset τους. Ο πελάτης DASH μπορεί να χρησιμοποιήσει την πληροφορία αυτή για να ζητήσει τα sub segments χρησιμοποιώντας HTTP GETS.



Εικόνα 2.18: MPEG-DASH Data Model [38].

2.4.2 Άλλα πρωτόκολλα υλοποίησης του HTTP Adaptive Streaming

Υπάρχουν βέβαια και άλλα πρωτόκολλα που μπορούν να υλοποιήσουν Adaptive Streaming μέσω του HTTP. Τα πιο γνωστά από αυτά είναι της Adobe, της Apple και της Microsoft που θα αναλύσουμε παρακάτω [19].

2.4.2.1 Adobe HTTP Dynamic Streaming

Το Adobe HTTP Dynamic Streaming είναι μια διαδικασία αποδοτικής μετάδοσης βίντεο streaming στον χρήστη με δυναμικό switching μέσω διαφορετικών streams που ποικίλουν ως προς την ποιότητα και το μέγεθος κατά την διάρκεια της αναπαραγωγής του βίντεο. Αυτό παρέχει στους χρήστες την καλύτερη δυνατή εμπειρία αναπαραγωγής την οποία μπορεί να υποστηρίξει το εύρος ζώνης τους και η μηχανή που χρησιμοποιούν. Ένας άλλος κύριος στόχος του είναι να κάνει την διαδικασία αναπαραγωγής ομαλή και απρόσκοπτη στους χρήστες, έτσι ώστε όταν είναι αναγκαία η προς τα κάτω ή προς τα πάνω κλιμάκωση της ποιότητας του stream να μην ενοχλείται η κανονική ροή της αναπαραγωγής [43].

2.4.2.2 Apple HTTP Adaptive Streaming

Το HTTP Live Streaming (HLS) είναι media streaming communication πρωτόκολλο βασισμένο στο HTTP το οποίο υλοποιείται από την Apple Inc. Το HLS υποστηρίζει τόσο την ζωντανή μετάδοση βίντεο όσο και βίντεο σε ζήτηση (on demand). Λειτουργεί, σπάζοντας τα streams ή τα συστατικά του βίντεο σε μικρά MPEG2-TS files με πολλαπλά bit rates και δίνει διάρκεια χρησιμοποιώντας ένα stream ή file τεμαχιστή. Την υλοποίηση ενός τέτοιου τεμαχιστή παρέχει η Apple. Ο τεμαχιστής είναι επίσης υπεύθυνος για την παραγωγή ενός συνόλου από index files με M3U8 format το οποίο λειτουργεί σαν playlist αρχείο για τα κομμάτια του βίντεο. Κάθε λίστα σχετίζεται με ένα δεδομένο bitrate και περιέχει τις διευθύνσεις URL στα κομμάτια με το συγκεκριμένο bitrate [44].

2.4.2.3 Microsoft Smooth Streaming

Το Microsoft Smooth Streaming είναι μια επέκταση του IIS Media Service το οποίο επιτρέπει να γίνεται adaptive streaming στον χρήστη μέσω HTTP. Εισάχθηκε για πρώτη φορά από την Microsoft τον Οκτώβριο του 2008 σαν κομμάτι της αρχιτεκτονικής Silverlight. Δυναμικά, παρακολουθεί το τοπικό εύρος ζώνης και την απόδοση του βίντεο rendering, και με αυτό τον τρόπο το Smooth Streaming είναι ικανό να βελτιστοποιεί το περιεχόμενο της αναπαραγωγής με την αλλαγή της ποιότητας του βίντεο σε πραγματικό χρόνο. Η Microsoft κατάφερε επιτυχώς την παράδοση τόσο της μετάδοσης βίντεο σε πραγματικό χρόνο όσο και την on-demand παράδοση 1080p HD video με το Smooth Streaming σε χρήστες του Silverlight [45].

Κεφάλαιο 3

Τεχνολογίες και Λογισμικό Υλοποίησης

Έχοντας αναλύσει τα πρότυπα κωδικοποίησης που χρησιμοποιήσαμε και τις λειτουργίες που προσφέρουν σε αυτό το κεφάλαιο θα περιγράψουμε τις παραμέτρους αξιολόγησης που θα χρησιμοποιηθούν στην διαδικασία της εξαγωγής των αποτελεσμάτων. Οι παράμετροι αυτοί έχουν σχέση με την ποιότητα των βίντεο καθώς επίσης και με το εύρος ζώνης που χρησιμοποιούν για την συγκεκριμένη ποιότητα.

3.1 Αντικειμενικές Τεχνικές Αξιολόγησης Ποιότητας Βίντεο

Στην παρούσα μεταπτυχιακή διατριβή, αυτό που προσπαθήσαμε ήταν να υλοποιήσουμε την έρευνα μας με τεχνικές για την αξιολόγηση της ποιότητας του βίντεο, οι οποίες θα ήταν πλήρως ελεύθερες ως προς την διάθεση τους και τον τρόπο υλοποίησης τους καθώς επίσης και ο τρόπος λειτουργίας και επεξεργασίας τους να είναι όσο το δυνατόν πιο απλός και εύχρηστος. Επίσης το

ίδιο σκεπτικό είχαμε και κατά την επιλογή του λογισμικού υλοποίησης. Ας τα δούμε όμως αναλυτικά.

3.1.1 PSNR

Ο PSNR (Peak Signal to Noise Ratio) είναι ένας αλγόριθμος ο οποίος εκφράζει την διαφορά σήματος μεταξύ της μέγιστης δυνατής τιμής και την δύναμη παραμόρφωσης του σήματος που επηρεάζει την ποιότητα του [46]. Χρησιμοποιείται δηλαδή ως ένα μέτρο εκτίμησης της επίδρασης των λαθών που συμβαίνουν σε ένα σύστημα μετάδοσης σήματος βίντεο ή ακόμη και κωδικοποίησης βίντεο. Το PSNR μπορεί να χρησιμοποιηθεί για τη μέτρηση της παραμόρφωσης μίας εικόνας ή πλαισίου λόγω των λαθών στην μετάδοση και κωδικοποίηση, σε σύγκριση με την αρχική εικόνα ή πλαίσιο. Πως ορίζεται και πως υπολογίζεται όμως;

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$
$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

Εικόνα 3.1: Τρόπος υπολογισμού και Ορισμός του PSNR (Peak Signal to Noise Ratio) [46].

Ο δεύτερος τύπος της εικόνας 4.1 ορίζει το PSNR. Ως MAX_i είναι η μέγιστη τιμή του pixel μίας εικόνας. Όταν το κάθε pixel αναπαριστάται με 8-bits τότε η τιμή αυτή είναι 256. Ο όρος PSNR εκφράζεται σε dB (decibel). Για να υπολογίσουμε το PSNR αρχικά υπολογίζουμε το Μέσο Τετραγωνικό Σφάλμα (MSE - Mean Squared Error) [47] ανάμεσα στην αρχική και την τελική μας εικόνα σύμφωνα με τον πρώτο τύπο της εικόνας 4.1, όπου για κάθε πλαίσιο έχουμε το m να είναι το πλάτος της εικόνας, το n να είναι το ύψος της εικόνας, το i είναι η στήλη που βρίσκεται το pixel, το j είναι η γραμμή που βρίσκεται το pixel και με I(i, j) και K(i, j) το αρχικό και το παραμορφωμένο βίντεο αντίστοιχα.

Όσο πιο μεγάλο είναι το PSNR τόσο μικρότερη παραμόρφωση υπάρχει στο βίντεο. Επιπλέον το PSNR είναι αλγόριθμος πλήρους αναφοράς δηλαδή απαιτείται και το αρχικό και το τελικό βίντεο και η εφαρμογή προϋποθέτει ότι οι δοκιμές αναφέρονται στο ίδιο περιεχόμενο και οι κωδικοποιήσεις έγιναν με τον ίδιο αλγόριθμο συμπίεσης.

3.2 Εύρος Ζώνης και Ποιότητα Βίντεο

Ένα από τα βασικά ερωτήματα που καλούμαστε να απαντήσουμε σε αυτή την μεταπτυχιακή διατριβή είναι το ποιος κωδικοποιητής μπορεί να προσφέρει την καλύτερη ποιότητα στο βίντεο μας, σε σχέση με το εύρος ζώνης.

3.2.1 PSNR και Bit rate

Το bit rate αναφέρεται στην ταχύτητα μεταφοράς δεδομένων μέσα από ένα κανάλι. Η μονάδα μέτρησης του bit rate είναι σε bits ανά δευτερόλεπτο (bits per second). Στα πλαίσια της μεταπτυχιακής διατριβής χρησιμοποιήσαμε το bit rate σε σχέση με το PSNR (Peak Signal to Noise Ratio). Αυτή η σχέση μας δίνει την δυνατότητα να παρατηρήσουμε το πως σύμφωνα με την ποιότητα ενός βίντεο, αλλάζει το bit rate. Το ιδανικότερο βίντεο είναι αυτό που θα έχει υψηλό PSNR (δηλαδή χαμηλότερη παραμόρφωση) και χαμηλό bit rate. Έτσι με βάση βίντεο σε διαφορετικές αναλύσεις, με διαφορετικές ζητούμενες τιμές Bitrate και με όλους τους πιθανούς κωδικοποιητές ερευνήσαμε για τα καλύτερα αποτελέσματα.

3.3 Εφαρμογές Ανοικτού Κώδικα

Οι Εφαρμογές Ανοικτού Κώδικα είναι το λογισμικό που μπορεί ελεύθερα να χρησιμοποιηθεί, να αντιγραφεί, να διανεμηθεί και να τροποποιηθεί. Είναι ένας εναλλακτικός τρόπος ανάπτυξης και χρήσης λογισμικού που βασίζεται στην ελεύθερη διάθεση του πηγαίου κώδικα. Το ελεύθερο λογισμικό στηρίζεται σε κοινότητα χρηστών και προγραμματιστών, οι οποίοι συνεργάζονται για τη συνεχή βελτίωση του λογισμικού. Η χρήση εφαρμογών ανοικτού κώδικα και λογισμικού προσφέρει πολλά πλεονεκτήματα σε σχέση με τις εφαρμογές κλειστού κώδικα. Το πιο σημαντικό πλεονέκτημα είναι ότι αυξάνεται η αξιοπιστία αφού η εφαρμογή ή το λογισμικό που χρησιμοποιούμε είναι είδη εφαρμοσμένο σε κάποια συστήματα που δουλεύουν, επομένως όλα τα προβλήματα έχουν είδη διορθωθεί. Ακόμα, το σύστημα που χρησιμοποιεί εφαρμογές ανοικτού κώδικα δεν υλοποιείται από το μηδέν, επομένως τόσο ο χρόνος ανάπτυξης όσο και ο χρόνος επικύρωσης μπορεί να μειωθεί. Τέλος, υπάρχει μικρότερη αβεβαιότητα στο κόστος ανάπτυξης, δεδομένου ότι το λογισμικό επαναχρησιμοποιείται και δεν αναπτύχθηκε από το μηδέν.

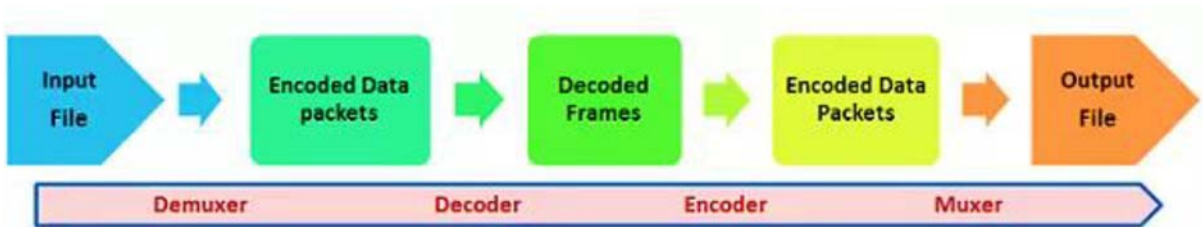
Οι εφαρμογές ανοικτού κώδικα που χρησιμοποιήθηκαν για την δημιουργία του συστήματος αυτού, είναι:

3.3.1 Η Πλατφόρμα FFmpeg

Για την επεξεργασία και συμπίεση των πλαισίων χρησιμοποιήθηκε η πλατφόρμα επεξεργασίας εικόνας και ήχου FFmpeg. Η πλατφόρμα FFmpeg είναι ένα ελεύθερο εργαλείο που προσφέρει ποικίλες επιλογές για πολλές μορφές αρχείων βίντεο και ήχου. Μπορεί να κωδικοποιήσει και να αποκωδικοποιήσει σχεδόν οποιουδήποτε τύπου αρχεία σε πολύ ικανοποιητικούς χρόνους, προσφέροντας ταυτόχρονα αρκετά μεγάλους λόγους συμπίεσης. Είναι υλοποιημένο σε γλώσσα C. Οι πιο βασικές βιβλιοθήκες που περιλαμβάνει είναι η libavcodec, μια βιβλιοθήκη μέσω της οποίας γίνεται κωδικοποίηση ήχου και εικόνας και η οποία χρησιμοποιείται και σε διάφορες άλλες εφαρμογές και η βιβλιοθήκη libavformat η οποία χρησιμοποιείται για πολυπλεξία και αποπολυπλεξία ήχου και βίντεο [36]. Περιέχει επίσης τις βιβλιοθήκες libavutil, libavfilter, libavdevice, libswscale και libswresample. Εκτός από τις βιβλιοθήκες, ο FFmpeg παρέχει και διάφορα εργαλεία όπως ο ffmpeg, ο ffserver, ο ffplay και ο ffprobe. Τα εργαλεία αυτά χρησιμοποιούνται ούτως ώστε να γίνεται επιτυχώς κωδικοποίηση, αποκωδικοποίηση, πολυπλεξία, αποπολυπλεξία, streaming καθώς επίσης και φιλτράρισμα. Οι χρήστες του FFmpeg μπορούν να χρησιμοποιήσουν το εργαλείο αυτό μέσω του command-line για να κωδικοποιήσουν και να αποκωδικοποιήσουν αρχεία ήχου και βίντεο σε διάφορες μορφές [48] Τέλος, οι εξωτερικές βιβλιοθήκες ανοικτού κώδικα που χρησιμοποιήθηκαν για τις πέντε κωδικοποιήσεις είναι:

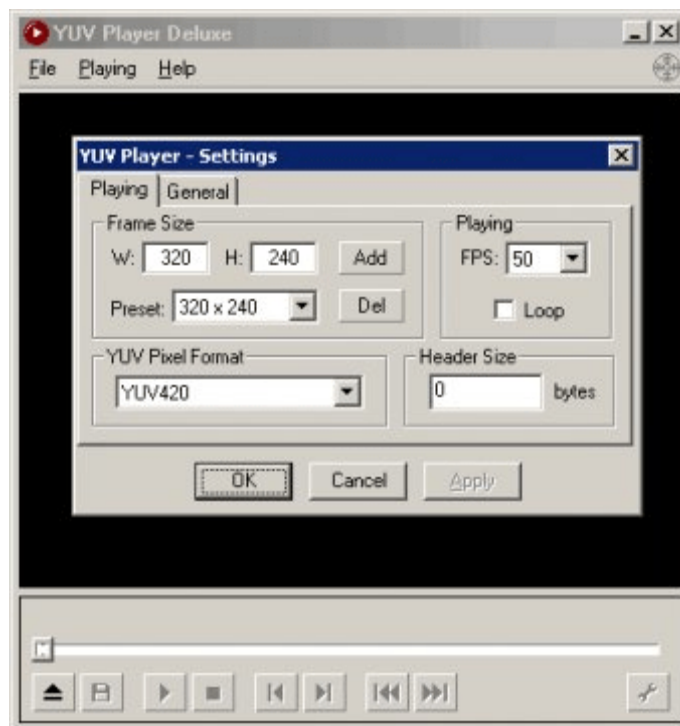
Εξωτερικές Βιβλιοθήκες Ανοικτού Κώδικα	
Κωδικοποιητής	Όνομα βιβλιοθήκης
H264	libx264
HEVC	libx265
VP8	libvpx
VP9	libvpx-vp9
AV-1	libaom

Πίνακας 3.1: Οι εξωτερικές βιβλιοθήκες ανοικτού κώδικα που χρησιμοποιήθηκαν στον FFmpeg.



Εικόνα 3.2: Λογική σειρά διεργασιών που εκτελεί ο FFmpeg [49].

3.3.2 YUV Player



Εικόνα 3.3: Το γραφικό περιβάλλον του YUV Player.

Το πρόγραμμα YUV Player, χρησιμοποιήθηκε για να αναπαράγει τα βίντεο τα οποία είναι σε ασυμπίεστη μορφή, δηλαδή τα .yuv αρχεία βίντεο. Ο όρος YUV αναφέρεται σε μια οικογένεια χρωματικών χώρων, οι οποίοι κωδικοποιούν την φωτεινότητα ανεξάρτητα από το χρώμα. Όπως και το RGB, το YUV χρησιμοποιεί τρεις τιμές για να αναπαραστήσει κάθε χρώμα, το Y, το U και το V. Η συνιστώσα Y αναπαριστά την φωτεινότητα του χρώματος και οι συνιστώσες U και V αναπαριστούν το χρώμα [50].

Κεφάλαιο 4

Μεθοδολογία Υλοποίησης

Στο κεφάλαιο αυτό θα παραθέσουμε αναλυτικά τις κινήσεις που κάναμε από την αρχή, δηλαδή από την επιλογή των ασυμπίεστων βίντεο (.yuv) μέχρι και την τελική εκτίμηση των αποτελεσμάτων της κωδικοποίησης μέσω των πέντε κωδικοποιητών.

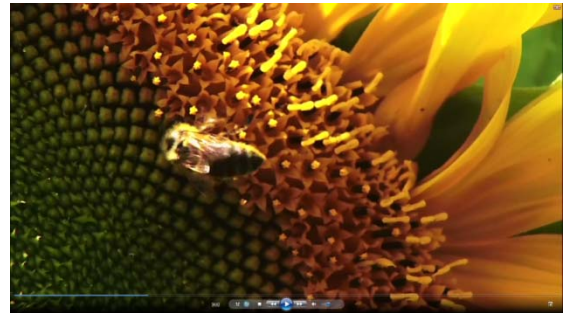
4.1 Βάση Δεδομένων – Πρότυπα Ασυμπίεστα Βίντεο

Το πρώτο μας βήμα ήταν η επιλογή των ασυμπίεστων βίντεο. Το Research Database της Netflix ήταν η καλύτερη επιλογή [51]. Αιτηθήκαμε την πρόσβαση μας στα πρότυπα και ασυμπίεστα βίντεο για σκοπούς έρευνας και το αίτημα μας ικανοποιήθηκε. Έτσι διαλέξαμε 6 βίντεο με ανάλυση 768x432 στα 25 frames per second. Επίσης προμηθευτήκαμε 3 ασυμπίεστα βίντεο με ανάλυση 1440x1080 στα 25 frames per second επίσης, από ερευνητικό πρόγραμμα του Dr. Αντρέα Παναγίδη που υλοποίησε σε συνεργασία με τον τομέα Υγείας του Πανεπιστημίου Κύπρου [52]. Διαλέξαμε δυο σετ βίντεο με διαφορετική ανάλυση για να δούμε πως συμπεριφέρονται οι διαφορετικές κωδικοποιήσεις σε διαφορετικές αναλύσεις [53]. Επιλέξαμε επίσης να κωδικοποιήσουμε τα βίντεο και σε διαφορετικά bit rates. Έτσι για μεν τα βίντεο με

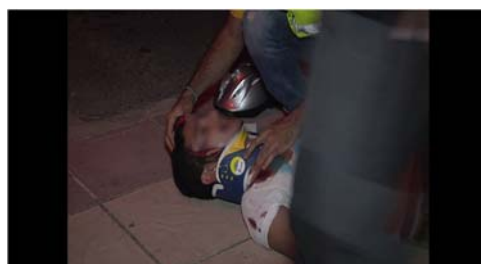
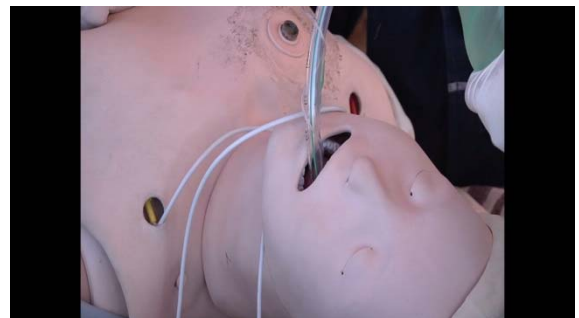
ανάλυση 768x432 επιλέξαμε ως στόχο τα ακόλουθα 8 bit rates 100,200,240,375,550,750,1000,1250 Kbps και για δε τα βίντεο 1440x1080 επιλέξαμε ως στόχο τα 8 bit rates 100,200,240,375,1000,2300,4300,8000 Kbps. Οι αναλύσεις των βίντεο μπορούν να φτάσουν στις ακόλουθες διαστάσεις βάση τιμής Bit Rate:

Bit Rate (Kbps)	Ανάλυση Εικόνας (Pixels)
100	265x144
200	320x180
240	384x216
375	384x216
550	512x288
750	640x360
1000	768x432
1250	768x432
2300	1280x720
4300	1920x1080
8000	3840x2160

Πίνακας 4.1: Δυνατή Διάσταση Εικόνας Βάση Τιμής Bit Rate [54].



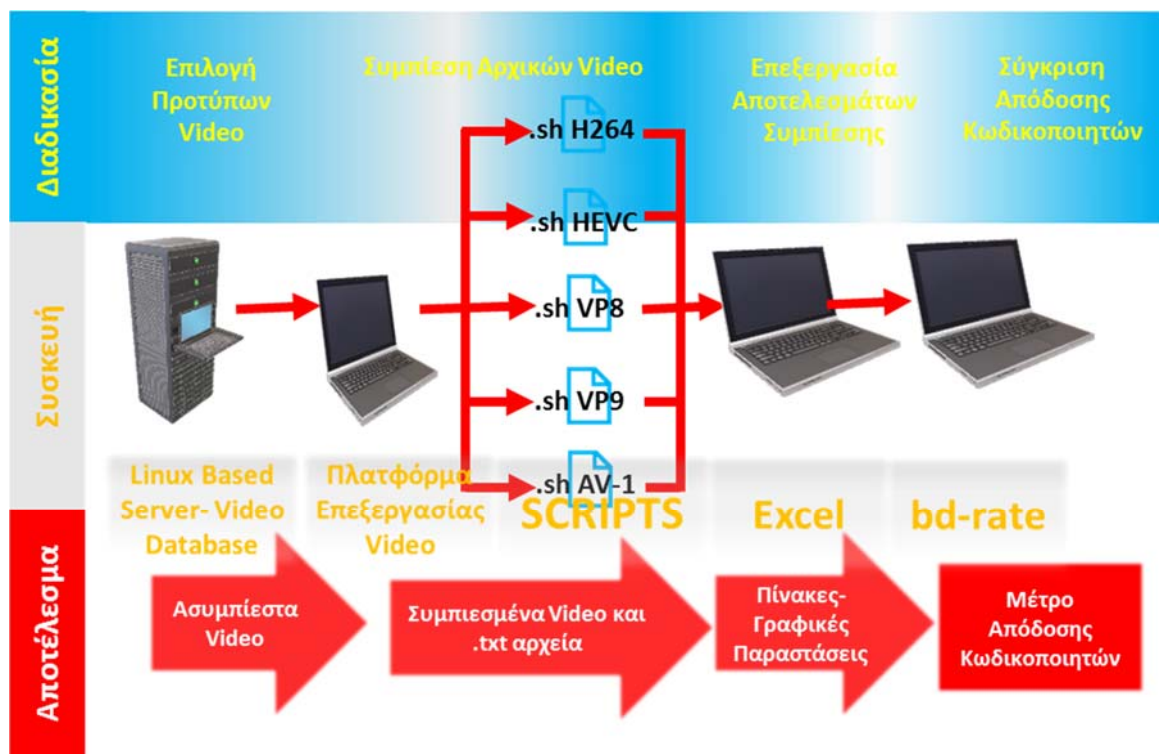
Εικόνα 4.1: Προεπισκόπηση των ασυμπιεστων βίντεο με διάσταση 768x432 [51].



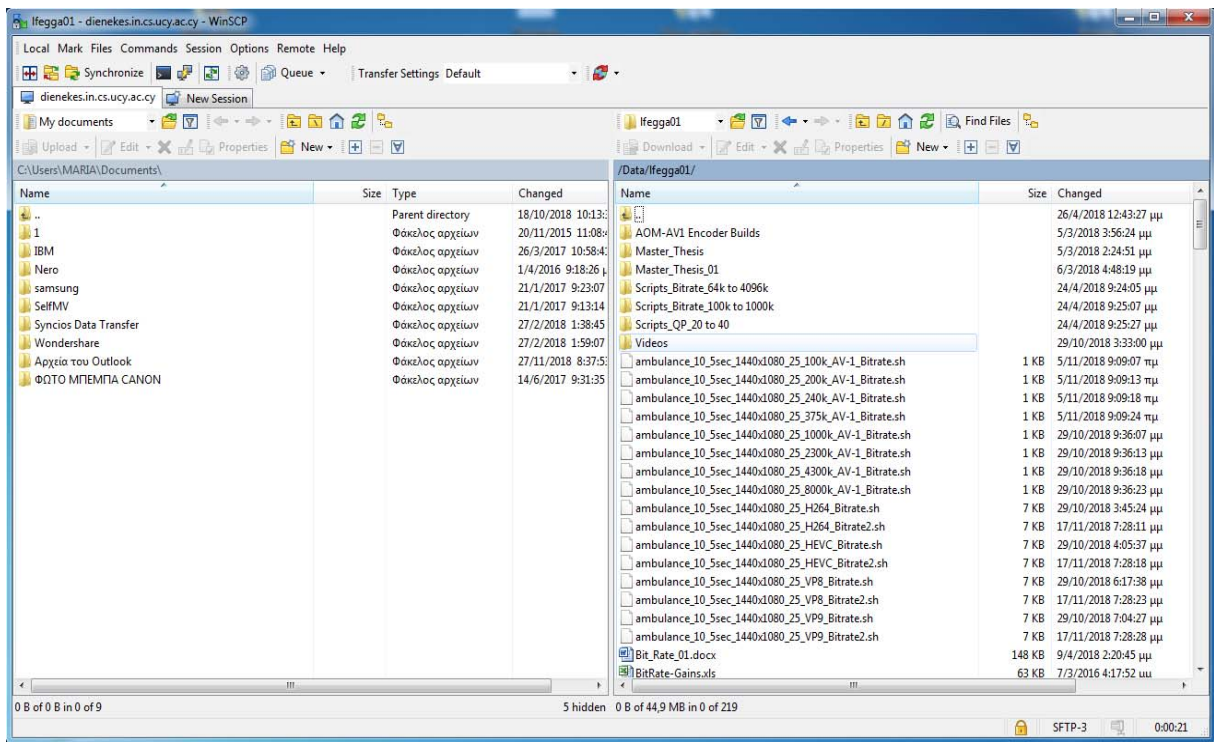
Εικόνα 4.2: Προεπισκόπηση των ασυμπιεστων βίντεο με διάσταση 1440x1080 [52].

4.2 Μεθοδολογία Σύγκρισης Προτύπων Ψηφιακής Κωδικοποίησης, Εξαγωγή και Καταγραφή Αποτελεσμάτων

Η υλοποίηση της όλης διαδικασίας έγινε σε Linux Based Server του πανεπιστημίου όπου είχε εγκατασταθεί η πλατφόρμα Ffmpeg καθώς και οι εξωτερικές βιβλιοθήκες ανοικτού κώδικα, υπεύθυνες για την κωδικοποίηση με τους διαφορετικούς κωδικοποιητές, σύμφωνα με το παρακάτω διάγραμμα.



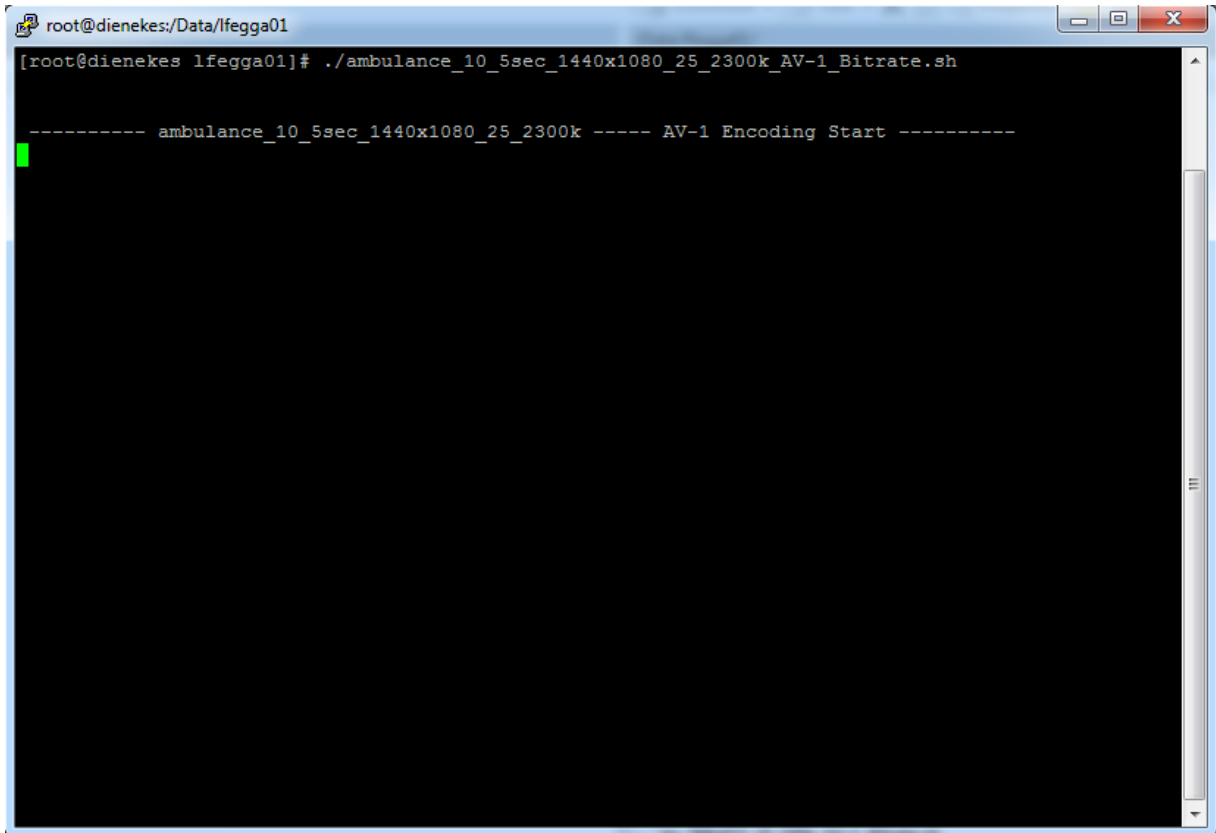
Εικόνα 4.3: Διαγραμματική Ανάλυση της Διαδικασίας.



Εικόνα 4.4: Εικόνα του Server dienekes.in.cs.ucy.ac.cy.

Αφού έχουμε ήδη αποθηκεύσει τα πρότυπα και ασυμπίεστα βίντεο στον Server μας, στην συνέχεια συντάξαμε scripts (.sh) με την μέθοδο των ***two pass*** (πλην της περίπτωσης του AV-1) όπου αναλύαμε όλες τις παραμέτρους υλοποίησης της συμπίεσης μέσω των κωδικοποιητών. Στο επόμενο κεφάλαιο θα αναλύσουμε την μεθοδολογία Two Pass τις εντολές των scripts.

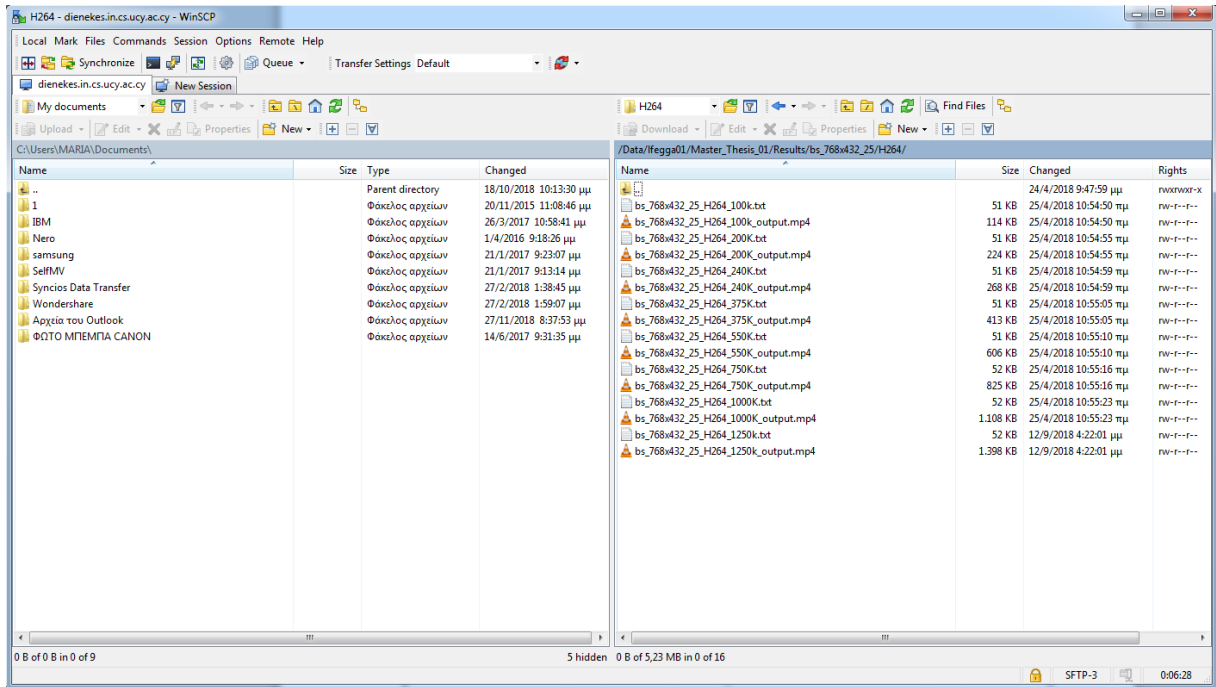
Σε περιβάλλον γραμμής εντολών, εκτελέσαμε τα scripts που δημιουργήθηκαν με την γλώσσα προγραμματισμού Bash.

A terminal window titled 'root@dienekes:/Data/lfegga01' is shown. The prompt is '[root@dienekes lfegga01]#'. The user has entered the command './ambulance_10_5sec_1440x1080_25_2300k_AV-1_Bitrates.sh'. The output of the script is '----- ambulance_10_5sec_1440x1080_25_2300k ----- AV-1 Encoding Start -----'. A green cursor is visible on the line following the output.

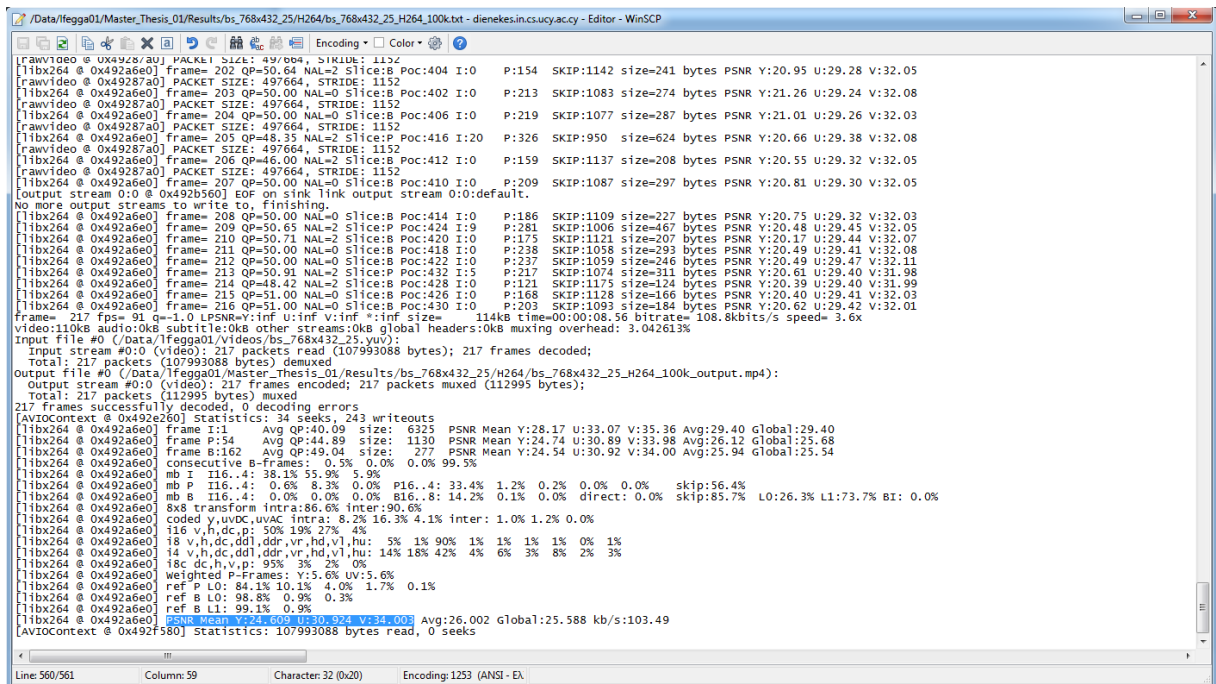
```
root@dienekes:/Data/lfegga01
[root@dienekes lfegga01]# ./ambulance_10_5sec_1440x1080_25_2300k_AV-1_Bitrates.sh
----- ambulance_10_5sec_1440x1080_25_2300k ----- AV-1 Encoding Start -----
```

Εικόνα 4.5: Εικόνα του Server dienekes.in.cs.ucy.ac.cy, κατά την εκτέλεση script file (.sh) σε περιβάλλον γραμμής εντολών.

Ακολουθως εξήχθησαν τα αποτελέσματα, τα οποία είναι ένα αρχείο .txt με τις λεπτομέρειες εκτέλεσης καθώς και τις τιμές των αποτελεσμάτων, και ένα αρχείο συμπιεσμένου βίντεο για κάθε περίπτωση κωδικοποίησης.

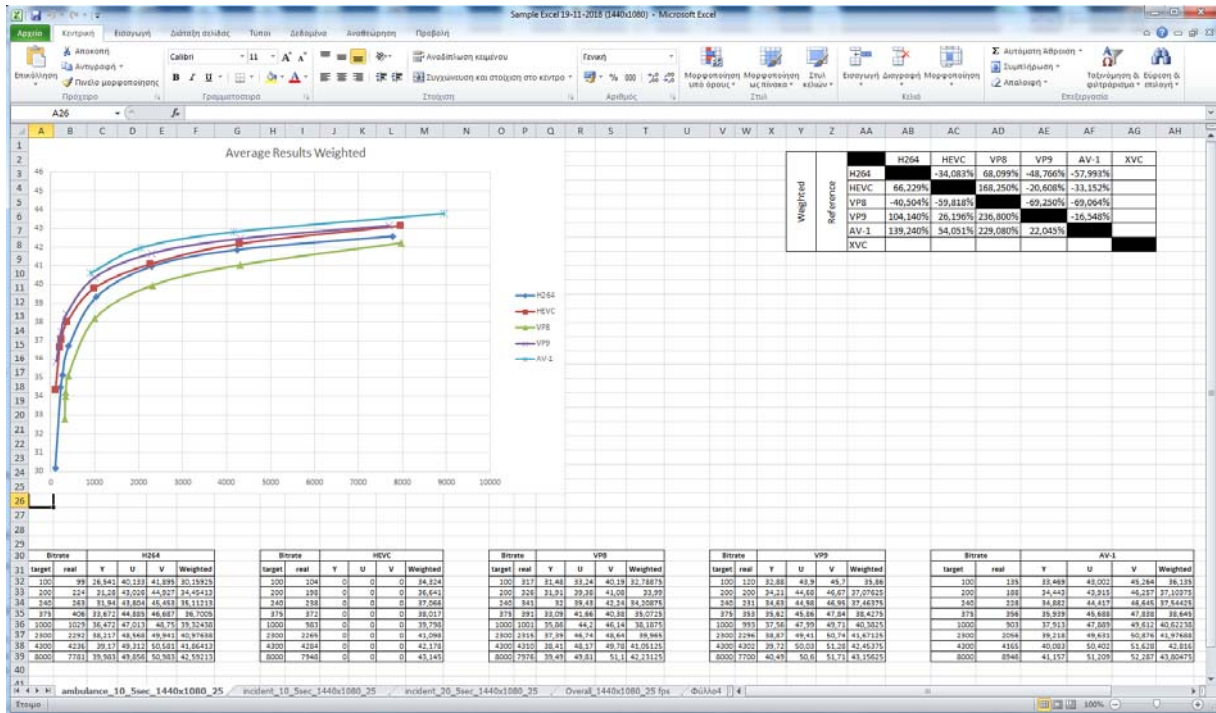


Εικόνα 4.6: Εικόνα από τα αποτελέσματα που εξήχθησαν μετά την κωδικοποίηση για το βίντεο bs_768x432_25 με κωδικοποίηση H264.



Εικόνα 4.7: Παράδειγμα εξαχθέντος αρχείου .txt με σκιασμένες τις τιμές PSNR για τα Y,U,V.

Στην συνέχεια και μέσω εντολών της Ffmpeg βρήκαμε το Real Bitrate της κωδικοποίησης, για κάθε βίντεο, για κάθε κωδικοποίηση και για όλες τις τιμές Bitrate. Τα αποτελέσματα τόσο των τιμών PSNR που υπολογίζονται μέσω των τιμών PSNR των Y,U,V, με την πράξη $PSNR = ((6*Y)+U+V)/8$ [55], όσο και των Real Bitrate τα εισαγάγαμε σε ένα αρχείο EXCEL.



Εικόνα 4.8: Αποτελέσματα ταξινομημένα σε αρχείο Excel.

Βάσει των αποτελεσμάτων δημιουργήσαμε τις γραφικές παραστάσεις και με βάση τις πραγματικές τιμές Bitrate και του υπολογισμένου PSNR υπολογίσαμε τις επί τις εκατό διαφορές στην ποιότητα των συμπίεσεων σε σχέση με το bitrate μεταξύ των διαφορετικών κωδικοποιήσεων για το ίδιο βίντεο, με την βοήθεια του αλγόριθμου **BD-PSNR Model** [56].

	H264	HEVC	VP8	VP9	AV-1
H264		-34,08%	68,09%	-48,76%	-57,99%
HEVC	66,22%		168,25%	-20,60%	-33,15%
VP8	-40,50%	-59,81%		-69,25%	-69,06%
VP9	104,14%	26,19%	236,80%		-16,54%
AV-1	139,24%	54,05%	229,08%	22,04%	

Πίνακας 4.2: Παράδειγμα αποτελεσμάτων με τις επί τις εκατό διαφορές στην ποιότητα σε σχέση με το Bitrate μεταξύ των διαφορετικών κωδικοποιητών.

Τέλος εξαγάγαμε τους συνολικούς μέσους όρους των διαφορών των κωδικοποιήσεων για όλα τα βίντεο και για τις δυο κατηγορίες βίντεο που είχαμε στην διάθεση μας.

Overall_1440x1080_25_fps_3_samples							
Weighted Reference		H264	HEVC	VP8	VP9	AV-1	
	H264			-37,25%	55,74%	-55,37%	-62,01%
	HEVC	72,98%			155,33%	-27,51%	-36,45%
	VP8	-34,84%	-58,31%			-69,82%	-71,76%
	VP9	143,27%	39,02%	245,00%			-12,65%
	AV-1	169,42%	59,92%	259,11%	16,73%		

Πίνακας 4.3 Μέσος όρος για τα αποτελέσματα των διαφορών των κωδικοποιήσεων στην περίπτωση των βίντεο με ανάλυση 1440x1080.

4.3 Εντολές στην Πλατφόρμα FFmpeg

Για να επεξεργαστούμε τα ασυμπίεστα βίντεο πρέπει να διαλέξουμε καταρχάς, ποια μέθοδο θα χρησιμοποιήσουμε. Η επιλογή μας θα εξαρτάται από το ποια παράμετρος είναι η ζητούμενη. Ο FFmpeg μας προσφέρει δύο μεθόδους γενικής χρήσης. Στην πρώτη μέθοδο **Constant Rate Factor (CRF)** ζητούμενο είναι η όσο το δυνατόν καλύτερη ποιότητα της εικόνας, αδιαφορώντας για το τελικό μέγεθος του αρχείου που θα δημιουργηθεί. Αυτό παρέχει μέγιστη απόδοση συμπίεσης με μία μόνο διέλευση. Ρυθμίζοντας τον λεγόμενο κβαντιστή για κάθε καρτέ, παίρνει το bit rate που χρειάζεται για να διατηρήσει το απαιτούμενο επίπεδο ποιότητας. Το μειονέκτημα είναι ότι δεν μπορούμε να τον χρησιμοποιήσουμε για να πάρουμε ένα συγκεκριμένο μέγεθος αρχείου ή να μην υπερβούμε ένα συγκεκριμένο μέγεθος ή bit rate, πράγμα που σημαίνει ότι αυτή η μέθοδος δεν συνιστάται για την κωδικοποίηση βίντεο για streaming. Η δεύτερη μέθοδος, που είναι και αυτή που χρησιμοποιήσαμε είναι η **Two Pass Coding** [48]. Σε αυτή την μέθοδο στοχεύουμε σε ένα συγκεκριμένο μέγεθος αρχείου ή κατ'επέκταση σε ένα συγκεκριμένο στόχο Bit Rate τον οποίο και δηλώνουμε μέσα στο κώδικα. Δεν είναι ζητούμενο η όσο το δυνατόν καλύτερη ποιότητα της εικόνας, αλλά ένα συγκεκριμένο Bit Rate. Η μέθοδος αυτή ταιριάζει στην περίπτωση μας, αφού συνιστάται για κωδικοποίηση βίντεο για streaming. Στην περίπτωση του AV-1 χρησιμοποιήσαμε μια ξεχωριστή μέθοδο που είναι μόνο εδώ συμβατή, όπου με ένα μόνο πέρασμα, υλοποιούμε την **Two Pass Coding**.

Οι εντολές που ακολουθήσαμε σε κάθε περίπτωση μέσα στην πλατφόρμα FFmpeg είναι οι ακόλουθες:

Εντολές που χρησιμοποιήσαμε στα SCRIPTS (.sh)

Περίπτωση 1^η H264

1^ο Πέρασμα

```
ffmpeg -v 50 -y -r 25 -s:v 1440x1080 -pix_fmt yuv420p -i /Data/lfegga01/Videos/ambulance_10_5sec_1440x1080_25.yuv -c:v libx264 -b:v 1000k -pass 1 -an -f mp4 /dev/null > /Data/lfegga01/Master_Thesis_01/Results/ambulance_10_5sec_1440x1080_25/H264/ambulance_10_5sec_1440x1080_25_H264_1000k_temp.txt 2>&1
```

Εικόνα 4.9: Εντολή κωδικοποίησης με FFmpeg (H264 – 1^ο Πέρασμα).

Οι παράμετροι που χρησιμοποιήσαμε για τον κώδικα είναι οι εξής:

Εντολή	Τι εκτελεί
-v 50	Παράμετρος που δείχνει στον FFmpeg πόσα και ποια στοιχεία ζητάμε να εξαχθούν στο αρχείο .txt
-y -r 25	Παράμετρος που δηλώνει τα αρχικά Frames Per Second του βίντεο
-s:v 1440x1080	Δηλώνει τις διαστάσεις (πλάτος x ύψος) του βίντεο
-pix_fmt yuv420p	Καθορίζουμε το pixel format του αρχικού βίντεο
-i /Data/lfegga01/Videos/ambulance_10_5sec_1440x1080_25.yuv	Καθορίζουμε το μονοπάτι (path) που βρίσκεται το βίντεο προς επεξεργασία
-c:v libx264	Με την παράμετρο -c:v καθορίζουμε το πρότυπο με το οποίο θέλουμε να κωδικοποιήσουμε και καλούμε την ανάλογη εξωτερική βιβλιοθήκη που προσθέσαμε στην πλατφόρμα FFmpeg
-b:v 1000k	Με την παράμετρο -b:v καθορίζουμε το target bitrate
-pass 1 -an -f mp4 /dev/null >	Δηλώνουμε ότι είμαστε στο πρώτο πέρασμα και το αποτέλεσμα (.mp4) αυτού θα χρησιμοποιηθεί στην επεξεργασία του στο δεύτερο πέρασμα. Το -f καθορίζει τον τύπο του αρχείου που θα εξαχθεί (στην περίπτωση μας .mp4)
/Data/lfegga01/Master_Thesis_01/Results/ambulance_10_5sec_1440x1080_25/H264/ambulance_10_5sec_1440x1080_25_H264_1000k_temp.txt 2>&1	Δημιουργεί temp αρχείο .txt στο path που δηλώνουμε.

Πίνακας 4.4: Εντολές στο 1^ο Πέρασμα

2^ο Πέρασμα

```
ffmpeg -v 50 -r 25 -s:v 1440x1080 -pix_fmt yuv420p -i  
/Data/lfegga01/Videos/ambulance_10_5sec_1440x1080_25.yuv -c:v  
libx264 -b:v 1000k -tune psnr -psnr -pass 2 -f mp4  
/Data/lfegga01/Master_Thesis_01/Results/ambulance_10_5sec_1440x1080_  
25/H264/ambulance_10_5sec_1440x1080_25_H264_1000k_output.mp4 >  
/Data/lfegga01/Master_Thesis_01/Results/ambulance_10_5sec_1440x1080_  
25/H264/ambulance_10_5sec_1440x1080_25_H264_1000k.txt 2>&1
```

Εικόνα 4.10: Εντολή κωδικοποίησης με FFmpeg (H264 – 2^ο Πέρασμα).

Οι διαφορές από το πρώτο πέρασμα είναι:

Εντολή	Τι εκτελεί
<code>-tune psnr -psnr</code>	Καθορίζει την μετρική όπου θα αποθηκευθεί κατά την κωδικοποίηση
<code>/Data/lfegga01/Master_Thesis_01/Results/ambulance_10_5sec_1440x1080_25/H264/ambulance_10_5sec_1440x1080_25_H264_1000k_output.mp4>/Data/lfegga01/Master_Thesis_01/Results/ambulance_10_5sec_1440x1080_25/H264/ambulance_10_5sec_1440x1080_25_H264_1000k.txt 2>&1</code>	Γίνεται ανακατεύθυνση της εξόδου σε ένα αρχείο .txt για την συγκέντρωση των αποτελεσμάτων

Πίνακας 4.5: Εντολές στο 2^ο Πέρασμα.

Και για τους κωδικοποιητές HEVC, VP8 και VP9 χρησιμοποιούμε ακριβώς τον ίδιο κώδικα με μόνη αλλαγή στην παράμετρο `-c:v` όπου εκεί καλούμε την εξωτερική βιβλιοθήκη που αντιστοιχεί σε κάθε κωδικοποιητή. Επίσης στις περιπτώσεις των VP8 και VP9 τα αρχεία βίντεο που προκύπτουν, έχουν την κατάληξη `.webm`

Περίπτωση 2^η AV-1

```
/usr/share/aom/aom_build/aomenc  
/Data/lfegga01/Videos/Pack4/ambulance_10_5sec_1440x1080_25.yuv -v --  
good --cpu-used=2 --width=1440 --height=1080 --target-bitrate=8000 -  
psnr -o  
/Data/lfegga01/Master_Thesis_01/Results/ambulance_10_5sec_1440x1080_  
25/AV-1/ambulance_10_5sec_1440x1080_25_AV-1_8000k.webm >  
/Data/lfegga01/Master_Thesis_01/Results/ambulance_10_5sec_1440x1080_  
25/AV-1/ambulance_10_5sec_1440x1080_25_AV-1_8000k.txt 2>&1
```

Εικόνα 4.11: Εντολή κωδικοποίησης με FFmpeg (AV-1).

Εντολή	Τι εκτελεί
<code>/usr/share/aom/aom_build/aomenc</code>	Δηλώνουμε που βρίσκεται η εξωτερική βιβλιοθήκη που θα χρησιμοποιηθεί για την κωδικοποίηση σύμφωνα με τον AV-1
<code>/Data/lfegga01/Videos/Pack1/ambulance_10_5sec_1440x1080_25.yuv</code>	Δηλώνουμε που βρίσκεται το βίντεο προς επεξεργασία.
<pre> -v --good --cpu-used=2 --width=1440 --height=1080 --target- bitrate=100 --psnr -o /Data/lfegga01/Master_Thesis_01/Results/ambulance_10_5sec_144 0x1080_25/AV-1/ambulance_10_5sec_1440x1080_25_AV- 1_100k.webm > /Data/lfegga01/Master_Thesis_01/Results/ambulance_10_5sec_144 0x1080_25/AV-1/ambulance_10_5sec_1440x1080_25_AV-1_100k.txt 2>&1 </pre>	Δίνουμε τις διαστάσεις του αρχικού βίντεο, το Bitrate που επιθυμούμε, παραμέτρους σχετικά με το πόσο θέλουμε να χρησιμοποιήσουμε την CPU μας στην διάρκεια της επεξεργασίας καθώς και το που θα αποθηκευθούν τα αρχεία .webm και .txt που θα δημιουργηθούν.

Πίνακας 4.6: Εντολές στο Script του AV-1.

Κεφάλαιο 5

Ανάλυση Αποτελεσμάτων

5.1 Ανάλυση Αποτελεσμάτων Α' Φάσης

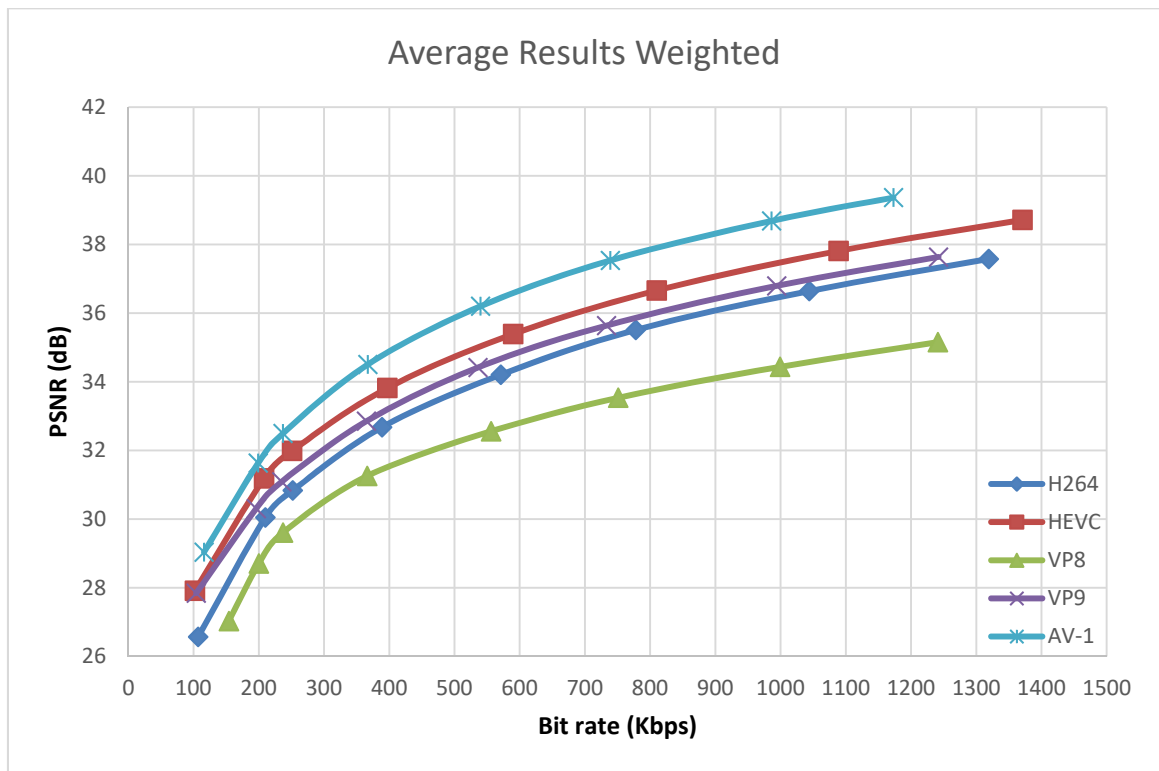
Στην προσπάθειά μας να συγκρίνουμε την απόδοση των πέντε κωδικοποιητών εικόνας (H264, HEVC, VP8, VP9, AV-1), ερευνούμε και επεξεργαζόμαστε σε πρώτη φάση, έξι πρότυπα βίντεο (ασυμπιεστά .yuv) σε ανάλυση 768x432. Προσπαθώντας να θέσουμε κάποια ποιοτικά και ποσοτικά χαρακτηριστικά του σεναρίου του κάθε βίντεο, δημιουργήσαμε τον πίνακα 5.1.

Στον πίνακα 5.1 θέσαμε κάποιες παραμέτρους που μπορούν να χαρακτηρίσουν την ροή (ή σενάριο) του κάθε βίντεο. Αυτές οι παράμετροι είναι η «ταχύτητα της κίνησης της κάμερας» σε σχέση με σταθερό σημείο και θέτουμε τιμές από μηδέν αν η κάμερα είναι ακίνητη, έως δέκα αν κινείται με πολύ μεγάλη ταχύτητα. Στην συνέχεια δημιουργήσαμε τον παράγοντα «κίνηση μέσα στα καρέ των αντικειμένων που καταγράφονται στο βίντεο» και πάλι η αριθμητική κλίμακα βαθμολόγησης είναι από το μηδέν, αν τα αντικείμενα του βίντεο είναι σταθερά έως και δέκα, αν κινούνται με μεγάλη ταχύτητα. Και πάλι στην ίδια αριθμητική κλίμακα βαθμολογήσαμε το πλήθος των χρωμάτων που παρατηρούμε. Τιμή μηδέν για ελάχιστα διαφορετικά χρώματα και τιμή δέκα για πάρα πολλά διαφορετικά χρώματα. Η επόμενη παράμετρος που θέσαμε είναι αυτή

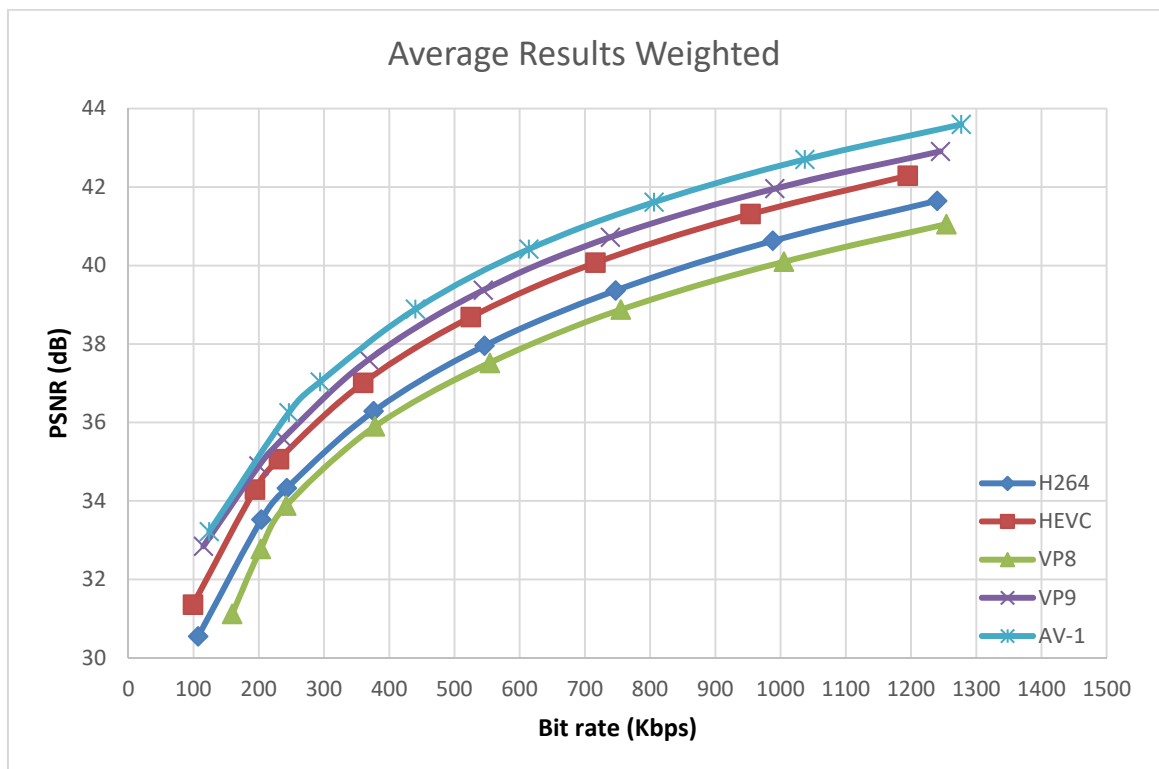
της «κατάληψης όλων των χρωμάτων σε όλο το καρέ». Με αυτή την παράμετρο προσπαθούμε μέσω της αριθμητικής κλίμακας να πούμε αν υπάρχουν όλα τα χρώματα σε όλες τις περιοχές της εικόνας. Ακολουθώς αξιολογούμαι με βοήθo την αριθμητική κλίμακα από το μηδέν έως το δέκα το γεγονός της εναλλαγής των χρωμάτων στα pixels. Αν δηλαδή σε γειτονικά pixels έχουμε διαφορετικά χρώματα. Τέλος με την τελευταία παράμετρο «Ταχύτητα εναλλαγής χρωμάτων στα pixel» αξιολογούμε αν αλλάζει γρήγορα η εικόνα και υπάρχουν συνεχόμενα frames με μεγάλη διαφορά. Μέσα από αυτές τις παραμέτρους που θέσαμε προσπαθήσαμε να συνδέσουμε τα σενάρια κίνησης που υπάρχουν στα βίντεο με τις έννοιες του χωρικού και χρονικού πλεονασμού, έτσι ώστε να γίνουν πιο εύκολα κατανοητά τα αποτελέσματα και οι διαφορές στην απόδοση των κωδικοποιητών που περιγράφονται στα γραφήματα που ακολουθούν μετά τον πίνακα 5.1.

Χαρακτηριστικά / Όνομα βίντεο	bs	pa	rh	sf	st	tr
Κίνηση Κάμερας (0 Σταθερή - 10 Πολύ Γρήγορα)	5	2	0	1	2	4
Κίνηση Καρέ - Αντικειμένου του βίντεο (0 Σταθερό - 10 Πολύ Γρήγορο)	2	5	4	1	2	4
Πλήθος Χρωμάτων (0 Λίγα - 10 Πάρα Πολλά)	2	5	4	4	2	3
Κατάληψη Όλων Των Χρωμάτων Σε Όλο Το Καρέ (0 Μικρή - 10 Πολύ Μεγάλη)	1	4	3	1	2	2
Εναλλαγή Χρωμάτων Στα Pixel (0 Μικρή - 10 Πολύ Μεγάλη)	2	4	3	2	1	2
Ταχύτητα Εναλλαγής Χρωμάτων Στα Pixel (0 Μικρή - 10 Πολύ Μεγάλη)	3	4	4	2	1	4

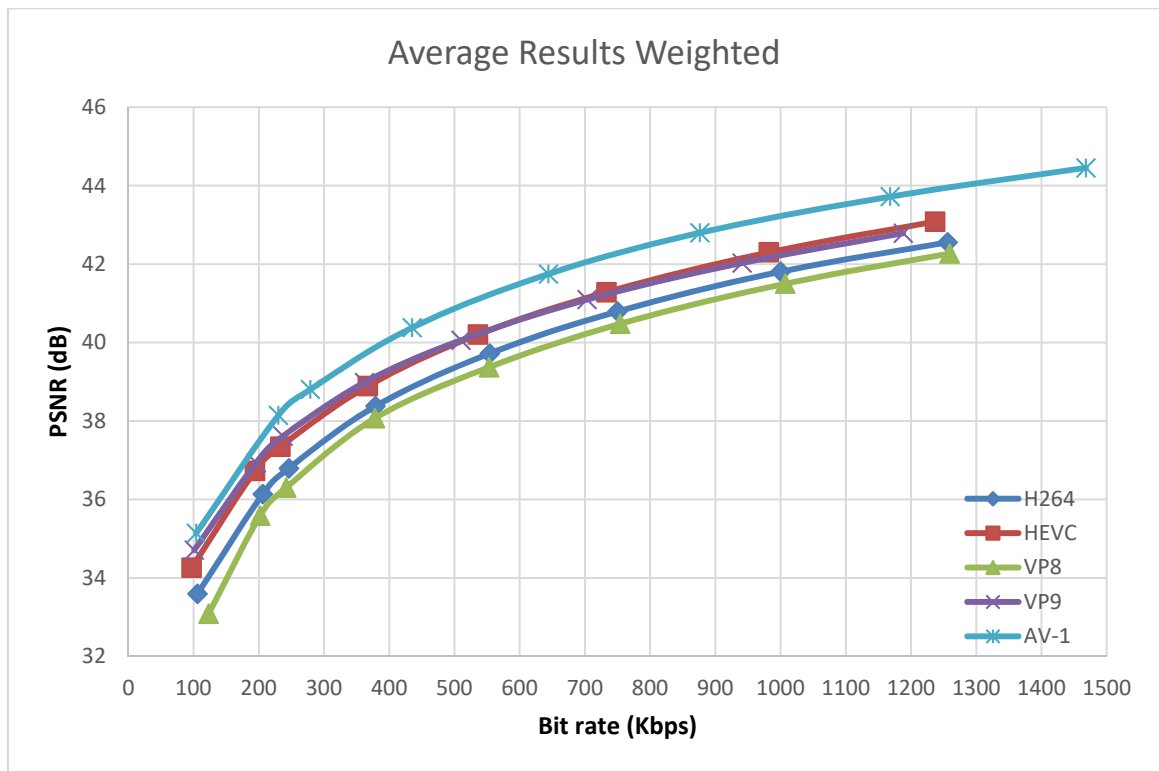
Πίνακας 5.1: Αποτύπωση Ποιοτικών Χαρακτηριστικών των Βίντεο με ανάλυση 768x432.



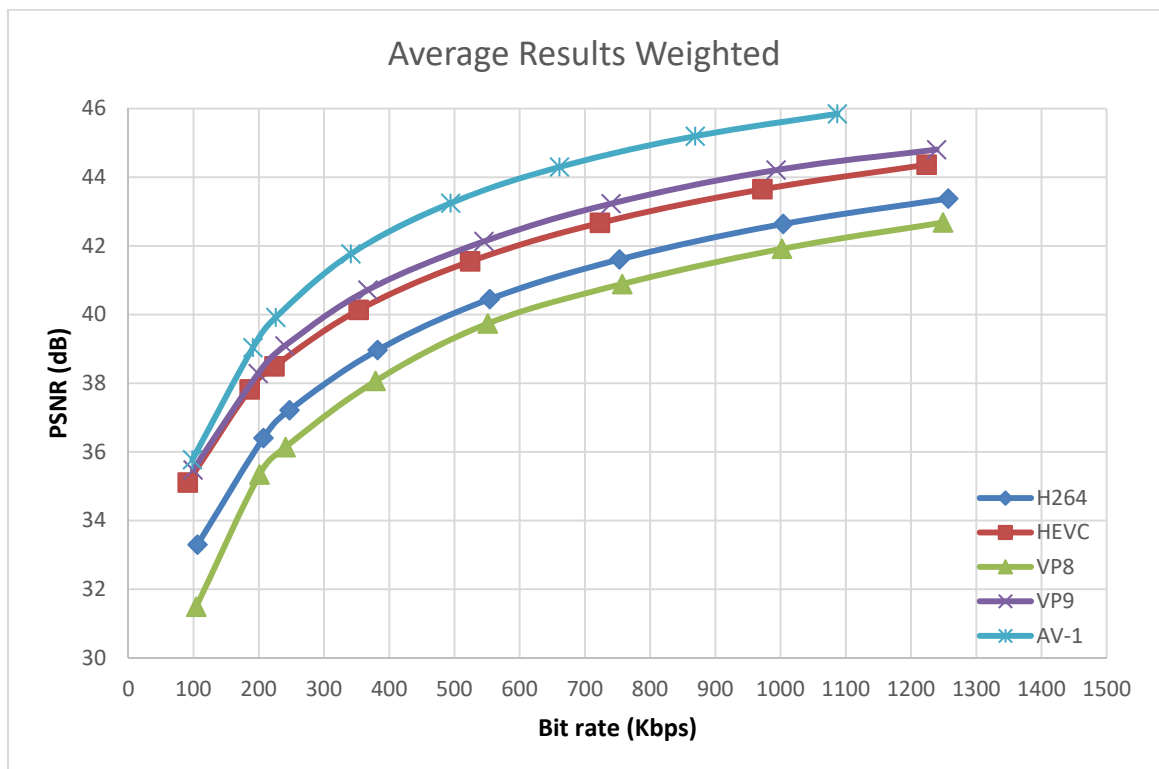
Γράφημα 5.1: Γράφημα Αποτελεσμάτων PSNR προς Bit rate του βίντεο bs_768x432.



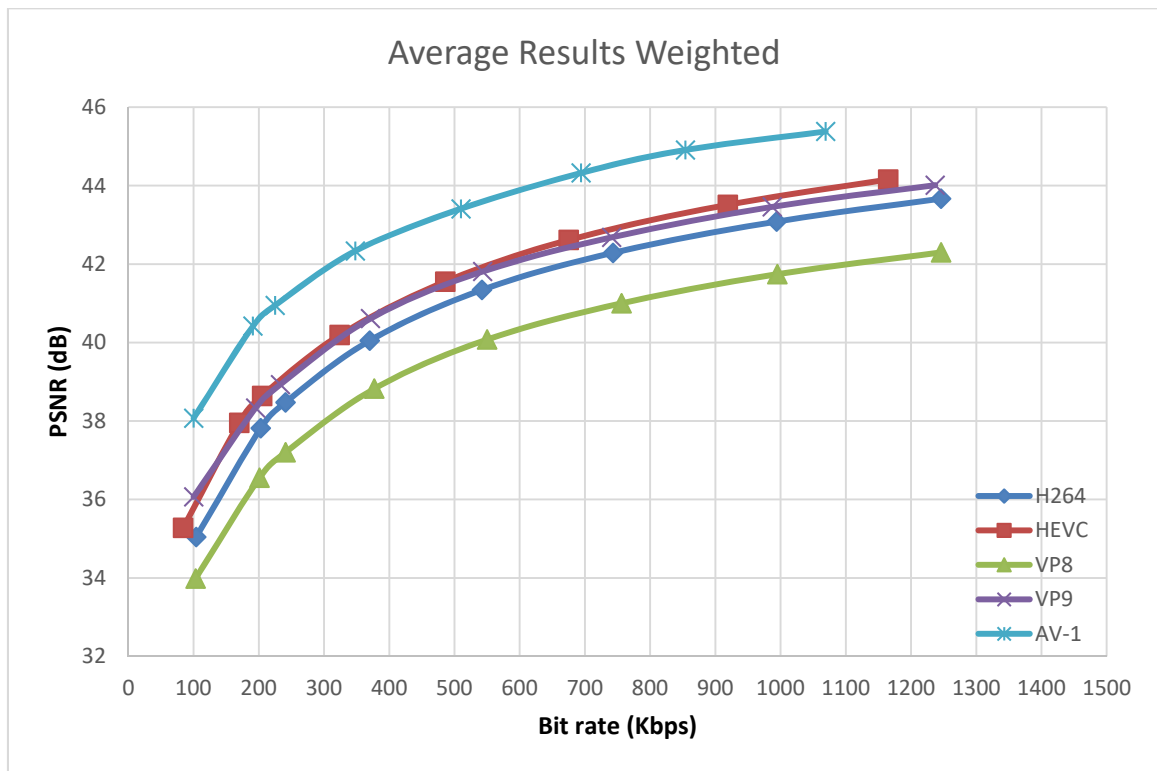
Γράφημα 5.2: Γράφημα Αποτελεσμάτων PSNR προς Bit rate του βίντεο ra_768x432.



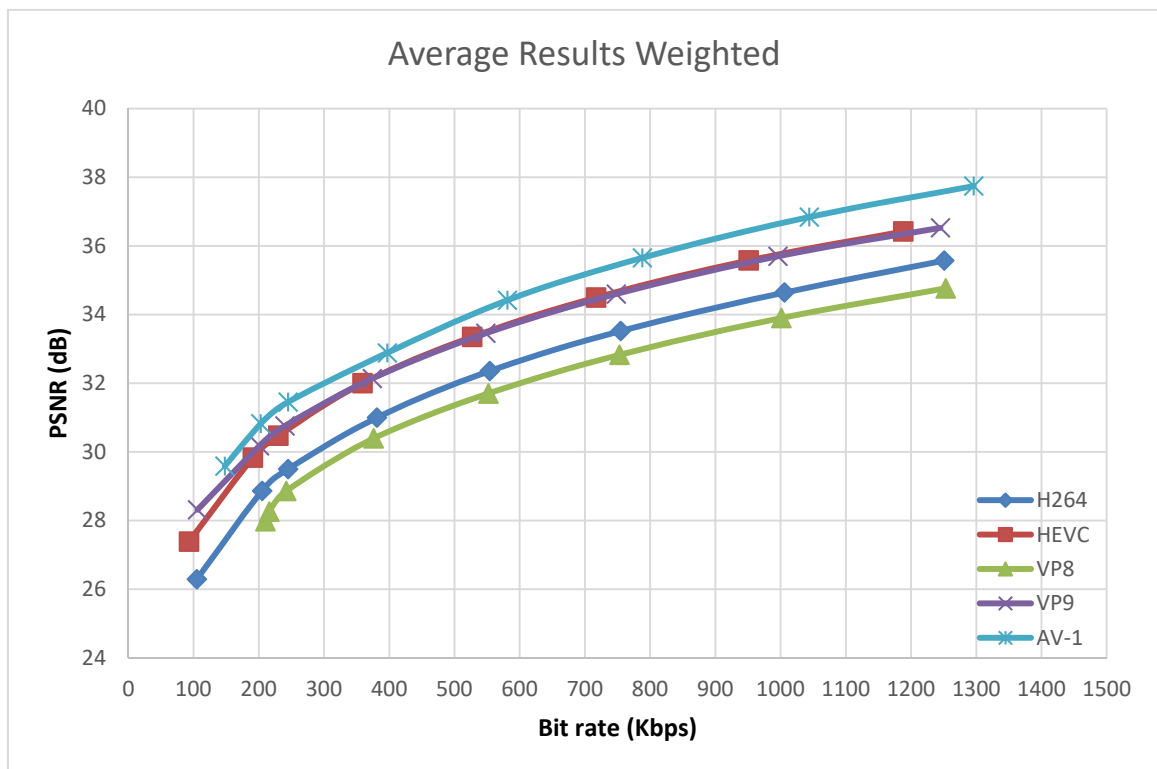
Γράφημα 5.3: Γράφημα Αποτελεσμάτων PSNR προς Bit rate του βίντεο rh_768x432.



Γράφημα 5.4: Γράφημα Αποτελεσμάτων PSNR προς Bit rate του βίντεο sf_768x432.



Γράφημα 5.5: Γράφημα Αποτελεσμάτων PSNR προς Bit rate του βίντεο st_768x432.



Γράφημα 5.6: Γράφημα Αποτελεσμάτων PSNR προς Bit rate του βίντεο tr_768x432.

Αυτά που έχουμε να σχολιάσουμε μετά και την παρατήρηση των βίντεο, την βαθμολόγηση των παραμέτρων του πίνακα 5.1 και των γραφημάτων με τις καμπύλες αποτελεσμάτων, είναι τα εξής:

Στο πρώτο μας βίντεο βλέπουμε την κάμερα να κινείται αριστερόστροφα με μέτρια ταχύτητα και η εικόνα μας να είναι κάποια δέντρα και το θρόισμα των φύλλων τους. Στο γράφημα 5.1 βλέπουμε να υπερέχει ο AV-1 με το μεγαλύτερο PSNR στο μικρότερο Bit rate . Ακολουθεί με αμέσως μεγαλύτερη τιμή PSNR ο HEVC με την μεγαλύτερη όμως τιμή Bit rate. Ακολουθούν VP9, H264 και VP8. Θα μπορούσαμε να την χαρακτηρίσουμε ως μια κανονική κατάταξη κωδικοποιητών, σύμφωνα και με τα αποτελέσματα των μέσων τιμών όλων των αποτελεσμάτων. Στο δεύτερο βίντεο βλέπουμε μια αργά κινούμενη προς τα αριστερά κάμερα, αλλά και μια εικόνα ενός πλήθους σε συνεχή κίνηση. Εναλλαγή χρωμάτων στα Pixel και μεγάλο πλήθος χρωμάτων σε αυτή την περίπτωση. Στο γράφημα 5.2 βλέπουμε και πάλι τον AV-1 να είναι ο αποδοτικότερος, ακολουθούμενος από τον VP9 και από κοντά ο HEVC. Μετά έρχονται οι H264 και VP8. Στο τρίτο βίντεο σταθερή η κάμερα μας σε αυτή την περίπτωση που καταγραφεί την κίνηση των αυτοκινήτων σε έναν δρόμο. Αρκετά χρώματα με γρήγορη εναλλαγή στα Pixels. Στο γράφημα 5.3 βλέπουμε τον AV-1 να είναι σταθερά πρώτος με διαφορά, ακολουθεί ο VP9 με τον HEVC σε απόσταση αναπνοής και έπονται H264 και VP8.

Στο τέταρτο βίντεο η κάμερα, ακολουθεί με πολύ μικρή ταχύτητα μια μέλισσα, που τρέφεται στον ανθό ενός λουλουδιού. Αρκετά χρώματα (κυρίως λόγω αλλαγής του φωτός), μικρή και αργή η εναλλαγή τους στα Pixels. Στο γράφημα 5.4 βλέπουμε τον AV-1 να είναι και πάλι πρώτος σε απόδοση, με σχετική διαφορά, ακολουθούμενος από VP9 και HEVC με μικρή μεταξύ τους διαφορά, και τέλος οι H264 και VP8. Πέμπτο βίντεο και έχουμε την κάμερα μας να κάνει αργά Zoom Out σε μια σταθερή εικόνα. Λίγα τα χρώματα, μικρή η εναλλαγή τους. Στο γράφημα 5.5 βλέπουμε τον AV-1 να είναι και πάλι ο πιο αποδοτικός και έπονται HEVC, VP9 και ακολουθούν H264 και VP8. Στο τελευταίο βίντεο και η κάμερα και το αντικείμενο κινούνται γρήγορα. Τα χρώματα δεν είναι πολλά, η ταχύτητα εναλλαγής τους στα Pixels είναι αρκετά γρήγορη. Στο γράφημα 5.6 βλέπουμε τον AV-1 να είναι πάλι πρώτος σε απόδοση και σε αυτό το βίντεο, ακολουθούμενος από τον VP9 και τον HEVC. Τελευταίοι όπως πάντα οι H264 και VP8.

Ακολούθως υλοποιήσαμε την σύγκριση των αποτελεσμάτων απόδοσης των πέντε κωδικοποιητών, βάσει του αλγόριθμου BD Rate/PSNR και τα συγκεντρωτικά αποτελέσματα φαίνονται στον πίνακα 5.2.

Overall_768x432_25_fps_6_samples							
Weighted Reference		H264	HEVC	VP8	VP9	AV-1	
	H264			-23,31%	27,82%	-23,47%	-42,68%
	HEVC	31,24%			68,21%	-0,50%	-26,08%
	VP8	-20,36%	-39,04%			-39,75%	-53,83%
	VP9	32,84%	1,40%	68,93%		-25,69%	
	AV-1	79,12%	38,50%	135,52%	39,07%		

Πίνακας 5.2 : Συγκεντρωτικός Πίνακας Σύγκρισης Απόδοσης Κωδικοποιητών των Βίντεο 768x432 (σε Bit rate από 100 έως 1250 Kbps).

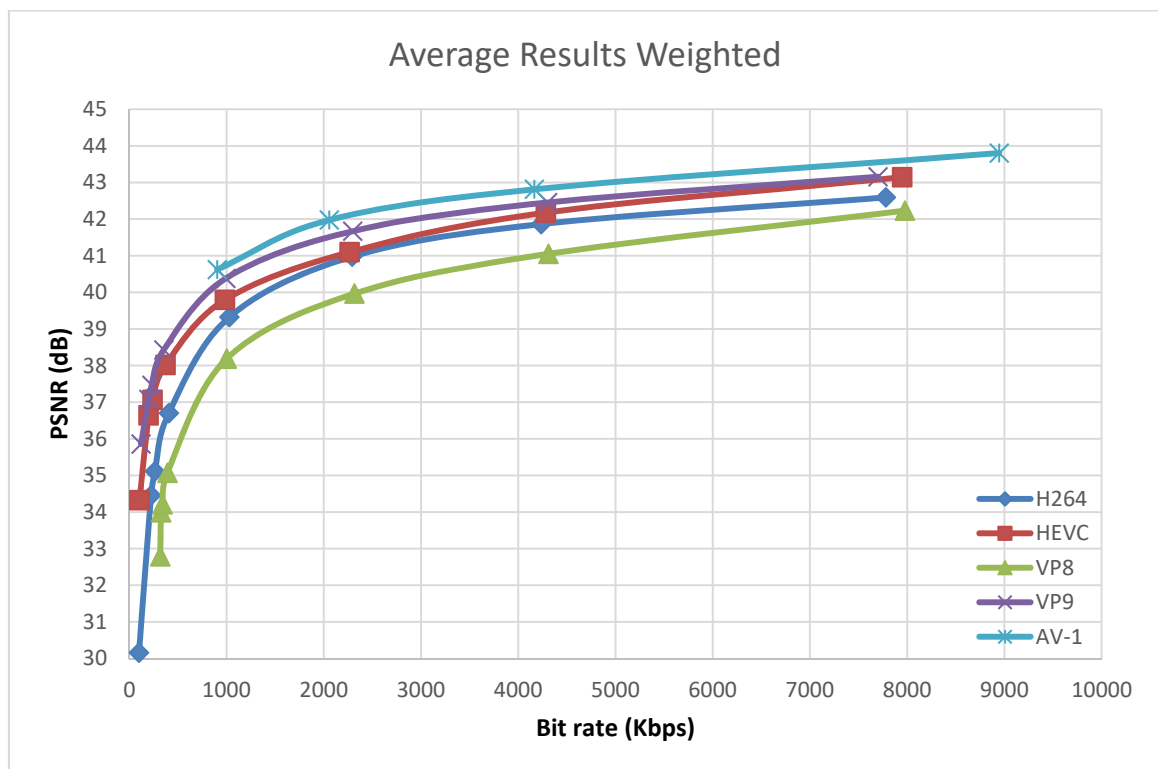
Στον πίνακα 5.2 με τα συγκεντρωτικά αποτελέσματα της απόδοσης των κωδικοποιητών παρατηρούμε την υπεροχή του AV-1 έναντι των HEVC και VP9 με τιμές 38,507% και 39,074% αντίστοιχα. Υπερτερεί σε ακόμη μεγαλύτερο ποσοστό από τον H.264, που φτάνει το 79,123% και με διπλάσιο από αυτό σχεδόν ποσοστό, από τον VP8. Οι HEVC και VP9 βρίσκονται μαζί με παραπλήσιες τιμές στην μεταξύ τους σύγκριση (μόλις 1,402% προηγείται ο VP9), ενώ ο H.264 έπεται των τιμών τους κατά 31,248% και 32,848% αντίστοιχα. Τέλος σε ποσοστό 27,826% πίσω από τον H.264 βρίσκουμε την απόδοση του VP8 στην μεταξύ τους σύγκριση.

5.2 Ανάλυση Αποτελεσμάτων Β' Φάσης

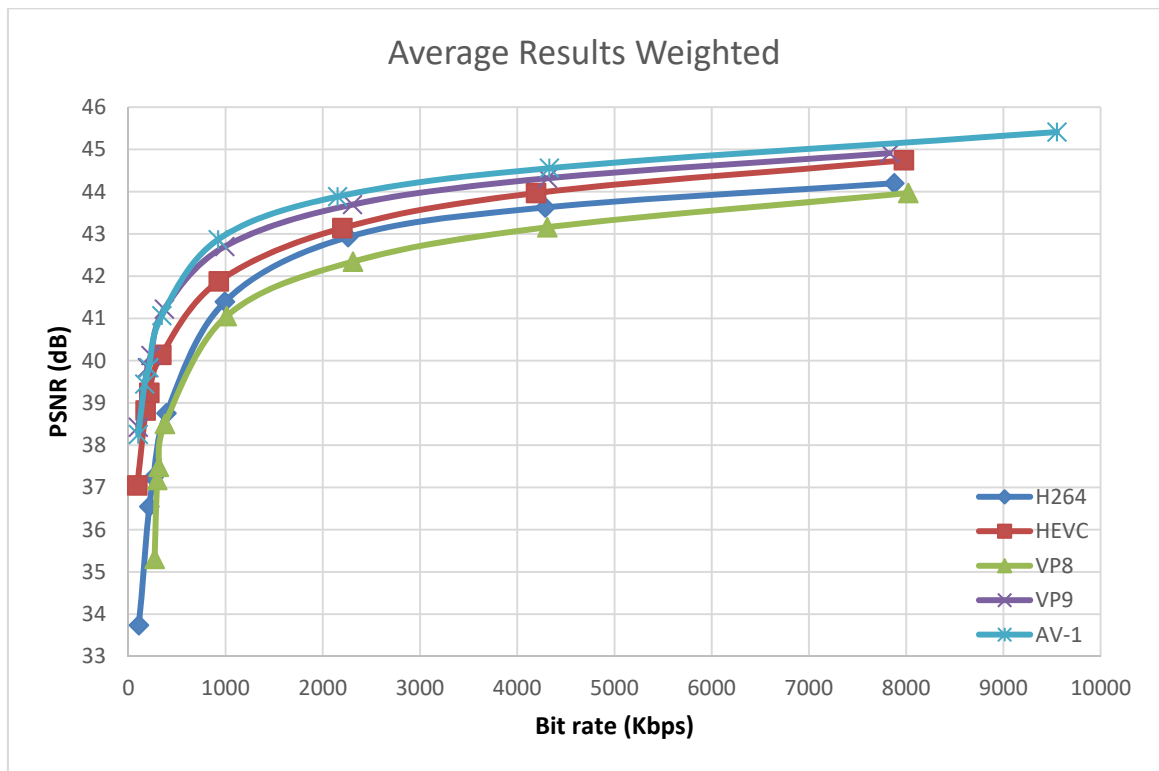
Στην προσπάθεια μας να συγκρίνουμε την απόδοση των πέντε κωδικοποιητών εικόνας (H264, HEVC, VP8, VP9, AV-1), ερευνούμε και επεξεργαζόμαστε σε δεύτερη φάση, τρία πρότυπα βίντεο (ασυμπιεστα .yuv) σε ανάλυση 1440x1080. Προσπαθώντας να θέσουμε κάποια ποιοτικά και ποσοτικά χαρακτηριστικά του σεναρίου του κάθε βίντεο, δημιουργήσαμε τον ακόλουθο πίνακα 5.3. Αναλύουμε τα βίντεο και βαθμολογούμε μέσω του πίνακα 5.3 τις παραμέτρους που έχουμε θέσει, όπως και στην Α' Φάση. Ομοίως δημιουργήσαμε και εδώ τα γραφήματα των αποτελεσμάτων κάθε βίντεο ξεχωριστά και τα παρουσιάζουμε ακολούθως του πίνακα 5.3.

Χαρακτηριστικά / Όνομα βίντεο	amb_10	inc_10	inc-20
Κίνηση Κάμερας (0 Σταθερή - 10 Πολύ Γρήγορα)	5	0	0
Κίνηση Καρέ - Αντικειμένου του βίντεο (0 Σταθερό - 10 Πολύ Γρήγορο)	4	1	2
Πλήθος Χρωμάτων (0 Λίγα - 10 Πάρα Πολλά)	3	2	2
Κατάληψη Όλων Των Χρωμάτων Σε Όλο Το Καρέ (0 Μικρή - 10 Πολύ Μεγάλη)	3	2	1
Εναλλαγή Χρωμάτων Στα Pixel (0 Μικρή - 10 Πολύ Μεγάλη)	2	1	1
Ταχύτητα Εναλλαγής Χρωμάτων Στα Pixel (0 Μικρή - 10 Πολύ Μεγάλη)	3	1	2

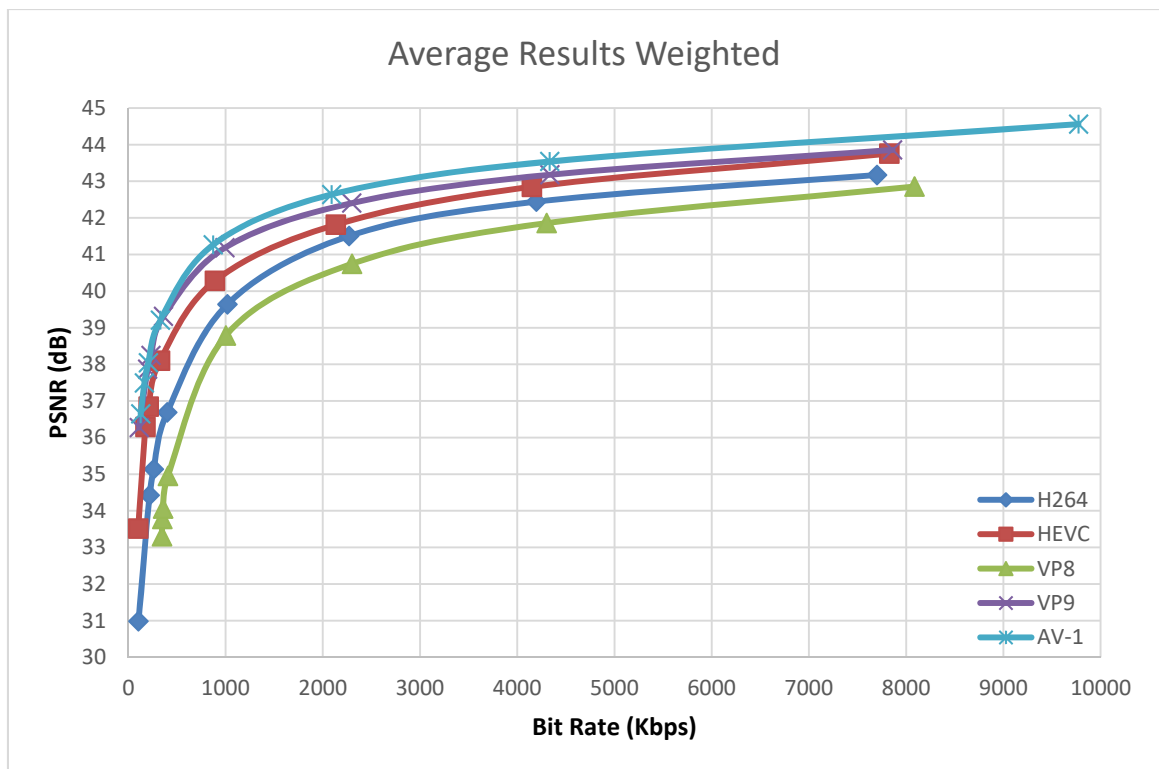
Πίνακας 5.3: Αποτύπωση Ποιοτικών Χαρακτηριστικών των Βίντεο με ανάλυση 1440x1080.



Γράφημα 5.7: Γράφημα Αποτελεσμάτων PSNR προς Bit rate του βίντεο ambulance_10_1440x1080.



Γράφημα 5.8: Γράφημα Αποτελεσμάτων PSNR προς Bit rate του βίντεο incident_10_1440x1080.



Γράφημα 5.9: Γράφημα Αποτελεσμάτων PSNR προς Bit rate του βίντεο incident_20_1440x1080.

Στο δεύτερο σετ βίντεο που είχαμε στην διάθεση μας, αυτά με ανάλυση 1440x1080 παρατηρούμε τα εξής:

Στο πρώτο βίντεο βλέπουμε την κάμερα να κινείται αριστερόστροφα με μέτρια ταχύτητα και η εικόνα μας να είναι ένα κινούμενο ασθενοφόρο πάνω σε μια επαρχιακή οδό. Στο γράφημα 5.7 με την απόδοση των κωδικοποιητών βλέπουμε να υπερέχει ο AV-1 με το μεγαλύτερο PSNR με μια μεγάλη τιμή Bit rate όμως. Ακολουθεί με αμέσως μεγαλύτερη τιμή PSNR ο VP9 και με την μικρότερη τιμή Bit rate. Ακολουθούν HEVC, H264 και VP8. Στο δεύτερο βίντεο βλέπουμε την κάμερα να είναι σταθερή και η εικόνα μας να είναι ένας διασώστης να υλοποιεί την κίνηση KARPA σε μια κούκλα που παίζει το ρόλο του τραυματία. Στο γράφημα 5.8 βλέπουμε να υπερέχει ο AV-1 με το μεγαλύτερο PSNR σε μια πολύ μεγάλη τιμή Bit rate. Ακολουθεί, από πολύ κοντά, με αμέσως μεγαλύτερη τιμή PSNR ο VP9 και με την μικρότερη τιμή Bit rate. Ακολουθούν HEVC και οι H264 και VP8 με πολύ μεγάλη διαφορά. Τέλος στο τρίτο βίντεο βλέπουμε την κάμερα να είναι σταθερή και η εικόνα μας να είναι ένας διασώστης να δίνει πρώτες βοήθειες σε έναν τραυματία από τροχαίο ατύχημα. Στο γράφημα 5.9 βλέπουμε να υπερέχει ο AV-1 με το μεγαλύτερο PSNR με μια πάρα πολύ μεγάλη τιμή Bit rate. Ακολουθεί με αμέσως μεγαλύτερη τιμή PSNR ο VP9 με την μικρότερη τιμή Bit rate. Ακολουθούν HEVC, H264 και VP8. Στον πίνακα 5.4 μπορούμε να δούμε τα συγκεντρωτικά αποτελέσματα σύγκρισης απόδοσης των κωδικοποιητών, τα οποία εξήχθησαν μέσω του αλγόριθμου BD-Rate/PSNR, από το δεύτερο σετ βίντεο.

Overall_1440x1080_25_fps_3_samples						
		H264	HEVC	VP8	VP9	AV-1
Weighted Reference	H264		-37,25%	55,74%	-55,37%	-62,01%
	HEVC	72,98%		155,33%	-27,51%	-36,45%
	VP8	-34,84%	-58,31%		-69,82%	-71,76%
	VP9	143,27%	39,02%	245,00%		-12,65%
	AV-1	169,42%	59,92%	259,11%	16,73%	

Πίνακας 5.4 : Συγκεντρωτικός Πίνακας Σύγκρισης Απόδοσης Κωδικοποιητών των Βίντεο 1440x1080 (σε Bit rate από 100 έως 8000 Kbps).

Αυτό που καταγράφεται στον πίνακα 5.4 είναι μια ακόμα μεγαλύτερη διαφορά του AV-1 από τον HEVC, η οποία φτάνει περίπου να διπλασιαστεί σε σχέση με τα βίντεο ανάλυσης 768x432. Έτσι βλέπουμε μια διαφορά της τάξης του 59,922% μεταξύ AV-1 και HEVC. Η διαφορά του AV-1 με τον VP9 βρίσκεται στο 16,737%, ενώ η απόσταση με τους H.264 και VP8 βλέπουμε να αποκτά ακόμα μεγαλύτερη διαφορά, με τιμές 169,427% και 259,110% αντίστοιχα. Στην μεταξύ των HEVC και VP9 σύγκριση, βρίσκουμε νικητή καθαρά τον VP9 σε ποσοστό 39,028%. Ο H.264 έπεται των VP9 και HEVC σε ποσοστό 143,270% και 72,983% αντίστοιχα. Στο τέλος βρίσκουμε τον VP8 να υπολείπεται του H.264 κατά 55,745%.

5.3 Συνολική Αποτίμηση

Συμπερασματικά από όλα τα βίντεο μας με ανάλυση 768x432, βλέπουμε μια καθολική υπεροχή του AV-1 (ανεξαρτήτως σεναρίου βίντεο). Υπερτερεί του HEVC σε ποσοστό 38,507%, του VP9 κατά 39,074%, του H.264 κατά 79,123% και του VP8 κατά 135,522%. Στους υπόλοιπους κωδικοποιητές βλέπουμε μια μάχη μεταξύ των HEVC και VP9. Με όσο λιγότερα χρώματα και χωρίς να υπάρχουν γρήγορες εναλλαγές της εικόνας, βλέπουμε τον HEVC να υπερέχει με μικρή διαφορά του VP9. Σε αντίθετη περίπτωση υπερισχύει ο VP9. Το συνολικό αποτέλεσμα δίνει για πολύ λίγο νικητή τον VP9 σε ποσοστό 1,402%, πράγμα που το θεωρούμε συγκυριακό βάσει των επιλεχθέντων βίντεο. Ο H264 υστερεί κατά 32,848% και 31,248% έναντι των VP9 και HEVC αντίστοιχα και τελευταίος ο VP8 με τον H.264 να υπερτερεί κατά 27,826% στην μεταξύ τους σύγκριση.

Εκτιμώντας τα αποτελέσματα από όλα τα βίντεο μας με ανάλυση 1440x1080, βλέπουμε και πάλι μια καθολική υπεροχή του AV-1 ανεξαρτήτως σεναρίου βίντεο. Στις περιπτώσεις που υπάρχει κίνηση στο βίντεο η διαφορά του AV-1 με τον VP9 μεγαλώνει, ενώ όσο έχουμε σταθερή εικόνα βλέπουμε τον VP9 να πλησιάζει στις τιμές PSNR τον AV-1. Κάτι που στην πρώτη φάση και σε ανάλυση video 768x432, συνέβαινε ακριβώς το αντίθετο. Στους υπόλοιπους κωδικοποιητές βλέπουμε τις τιμές PSNR του HEVC να έχουν μεγαλύτερη απόκλιση από τους δυο επικρατέστερους κωδικοποιητές, από ότι στην περίπτωση που το βίντεο μας είχε ανάλυση 768x432. Το ίδιο συμβαίνει και στους VP8 και H264 καθώς επίσης και στις περιπτώσεις που η εικόνα μας είναι σταθερή να έχουν ακόμα χειρότερα αποτελέσματα σε σχέση με τους AV-1 και VP9. Έτσι ο AV-1 είναι ο καθολικός νικητής με ποσοστό 59,922% έναντι του HEVC, 16,737% έναντι του VP9, 169,427% έναντι του H.264 και 259,110% έναντι του VP8. Ο VP9 υπερέχει του HEVC σε ποσοστό 39,028% τον οποίο HEVC ακολουθεί ο H.264 με ποσοστό 72.983% στην

μεταξύ τους σύγκριση. Τελευταίος κατατάσσεται ο VP8, αφού χάνει με ποσοστό 55,745% από τον H.264.

Ένα ακόμα στοιχείο που θα έπρεπε να αναφέρουμε, είναι ο χρόνος που χρειάζονται οι κωδικοποιητές ώστε να ολοκληρώσουν την επεξεργασία του βίντεο. Έτσι για κάθε βίντεο βλέπουμε τον H264 να χρειάζεται ελάχιστα λεπτά για να ολοκληρώσει και λίγο χρόνο παραπάνω για τον VP8. Σχεδόν διπλάσιο ή και περισσότερο χρόνο χρειάζονται οι HEVC και VP9. Στην περίπτωση του AV-1 έχουμε τρομερά μεγαλύτερους χρόνους. Έτσι για ένα βίντεο με ανάλυση 768x432 και για ζητούμενο Bit Rate 100 Kbps χρειάζεται περισσότερες από 12 ώρες και αν φτάσουμε σε ανάλυση 1440x1080 και ζητούμενο Bit Rate 8000Kbps να χρειάζεται σχεδόν ολόκληρες τρεις ημέρες!

Τελειώνοντας να πούμε ότι θεωρούμε τα αποτελέσματα μας αξιόλογα και αξιόπιστα, αφού είναι σύμφωνα με την εξαγγελία και τις εργαστηριακές έρευνες της AOMedia που κάνουν λόγο για υπεροχή του AV-1 έναντι του HEVC κατά τουλάχιστον 30%. Έχουμε απάντηση και μάλιστα σωστή, στο αρχικό ερώτημα που κλιθήκαμε να απαντήσουμε.

Κεφάλαιο 6

Επίλογος

6.1 Γενικότερα Συμπεράσματα

Στην παρούσα μεταπτυχιακή διατριβή προσπαθήσαμε να αναλύσουμε τις αρχές και τις μεθοδολογίες που χρησιμοποιούνται για την κωδικοποίηση των βίντεο. Εξηγήσαμε ποιες είναι και την χρησιμότητα τους, τις παραμέτρους που λαμβάνονται υπόψη για την δημιουργία του αλγορίθμου σε κάθε μια περίπτωση ξεχωριστά για κάθε έναν κωδικοποιητή. Επιλέξαμε τα κατάλληλα πρότυπα βίντεο για την επεξεργασία τους.

Τα βίντεο που χρησιμοποιήθηκαν τα κατατάξαμε σε δύο κατηγορίες. Η πρώτη περιλαμβάνει πρότυπα και ασυμπίεστα βίντεο, ανάλυσης 768x432 pixels, με 25 Frames Per Second, τα οποία μας παραχωρήθηκαν από τον τομέα έρευνας της Αμερικάνικης, παγκοσμίου βεληνεκούς εταιρείας στον τομέα της συνδρομητικής καλωδιακής τηλεόρασης, Netflix. Η δεύτερη κατηγορία περιλαμβάνει πρότυπα και ασυμπίεστα βίντεο, ανάλυσης 1440x1080 με 25 Frames Per Second, που προέρχονται από το εργαστήριο Ηλεκτρονικής Υγείας του Πανεπιστημίου Κύπρου.

Μέσω ενός Linux Based Server και λογισμικού ανοικτού κώδικα προσπαθήσαμε και καταφέραμε να κωδικοποιήσουμε τα βίντεο σε πλήθος διαφορετικών ζητούμενων τιμών Bit rate και με τους πέντε κωδικοποιητές που χρησιμοποιούνται στις μέρες μας. Αυτοί είναι ο H.264 και ο HEVC της Joint Video Team, οι VP8 και VP9 της On2 και της Google καθώς και ο τελευταίος, με χρονική σειρά εμφάνισης, AV-1 της AOMedia.

Τα αποτελέσματα της κωδικοποίησης των βίντεο συλλέχτηκαν, αναλύθηκαν και επεξεργάστηκαν με αντικειμενικές τεχνικές αξιολόγησης ποιότητας ψηφιακού βίντεο και ακολούθως αποτυπώθηκαν μέσω γραφημάτων. Ακολούθως και με την βοήθεια του αλγόριθμου σύγκρισης κωδικοποιητών BD-Rate/PSNR, υλοποιήσαμε σύγκριση όλων των αποτελεσμάτων για κάθε βίντεο και για όλα τα πιθανά ζεύγη κωδικοποιητών.

Στο πρώτο σετ βίντεο, αυτά με ανάλυση 768x432, βρίσκουμε καθολικό νικητή τον AV-1 με ποσοστά 38,507% από τον HEVC, 39,074% από τον VP9, 79,123% από τον H.264 και 135,422% από τον VP8. Ο VP9 προηγείται βραχεία κεφαλή μόλις 1,402% από τον HEVC στην μεταξύ τους σύγκριση, ενώ ο H.264 έπεται του HEVC κατά 31,248% και ο VP8 έπεται του H.264 κατά 27,826%.

Στο δεύτερο σετ με βίντεο, αυτά με ανάλυση 1440x1080, βρίσκουμε και πάλι νικητή σε όλες τις συγκρίσεις τον AV-1 με ποσοστά 16,737% έναντι του VP9, 59,922% έναντι του HEVC και 169,427% και 259,110% αντίστοιχα έναντι των H.264 και VP8. Ο VP9 κατατάσσεται δεύτερος με διαφορά 39,028% από τον τρίτο HEVC. Ακολουθεί ο H.264 που υστερεί του HEVC κατά 72,983% και τελευταίος σε απόδοση ο VP8 που στην σύγκριση με τον H.264 υστερεί κατά 55,745%.

Τα αποτελέσματα της σύγκρισης καταγράφηκαν, αξιολογήθηκαν και μας οδήγησαν σε ασφαλή απάντηση του αρχικού ερευνητικού μας ερωτήματος, επιβεβαιώνοντας την υπεροχή του AV-1 κωδικοποιητή έναντι του HEVC κατά τουλάχιστον 30%, καθώς και των υπολοίπων κωδικοποιητών.

Εν κατακλείδι, αν σκεφτούμε ότι σήμερα ο κωδικοποιητής που επικρατεί στις περισσότερες εφαρμογές όπου υπάρχει κωδικοποίηση εικόνας είναι ο H.264, καταλαβαίνουμε τα κέρδη που θα έχουμε μελλοντικά στην απρόσκοπτη μετάδοση με μικρότερη καθυστέρηση, εικόνας με καλύτερη ποιότητα, όταν υιοθετηθεί η κωδικοποίηση εικόνας μέσω του AV-1.

6.2 Μελλοντική Εργασία

Στις μέρες μας η αναζήτηση του καλύτερου και πιο αποδοτικού κωδικοποιητή είναι σε εξέλιξη. Ο νεοφερμένος AV-1 υπόσχεται πολλά. Οι εταιρείες που βοηθούν στην ανάπτυξη του είναι οι κολοσσοί της πληροφορικής και της τηλεόρασης. Οι εξελίξεις καταϊγιστικές. Αρκεί να σκεφτούμε ότι η πρώτη σταθερή έκδοση εξωτερικής βιβλιοθήκης για τον AV-1 δόθηκε στους ερευνητές, το δεύτερο τρίμηνο του 2018! Νεότεροι κωδικοποιητές θα εμφανισθούν και θα μπουν και αυτοί στην διαδικασία της συγκριτικής έρευνας. Ήδη έχει κάνει την εμφάνιση του ο XVC που υπόσχεται ακόμα καλύτερα αποτελέσματα από τον AV-1. Με την είσοδο τους στην καθημερινότητα μας, κάνουν την ζωή μας ακόμα πιο εύκολη, ακόμα καλύτερη και αυτό είναι το ζητούμενο.

Η ενασχόληση μας με το αντικείμενο της κωδικοποίησης της εικόνας μας έκανε ξεκάθαρα τα πράγματα που συναντούμε στην καθημερινότητα μας και δεν γνωρίζαμε πως δουλεύουν. Έννοιες και αντικείμενα που αφορούν και σχετίζονται με την εικόνα και την μετάδοση της, μας είναι πλέον γνώριμα. Πραγματικά είναι μια γνώση που μας έλυσε πολλές απορίες και μας έκανε να δούμε από μια άλλη οπτική γωνιά, αυτή του ερευνητή, αυτό το πολύπλοκο αλλά συνάμα και τόσο γοητευτικού αντικειμένου.

Η εργασία μας και ο τρόπος που αναλύθηκε στην παρούσα μεταπτυχιακή διατριβή, νομίζουμε ότι μπορεί να θέσει μια εύχρηστη βάση για τους μελλοντικούς ερευνητές επί του αντικειμένου. Οι δικές μας σκέψεις για το μέλλον, είναι η δημιουργία μιας ακόμα μεγαλύτερης σε έκταση έρευνας, που θα περιλαμβάνει ακόμα περισσότερα βίντεο με ακόμα μεγαλύτερες αναλύσεις και με περισσότερες τεχνικές αξιολόγησης. Η είσοδος του κωδικοποιητή XVC στο προσκήνιο της συμπίεσης της εικόνας ανοίγει το δρόμο για μια νέα ερευνητική διαδικασία σύγκρισης του με τον AV-1. Περαιτέρω, σημαντική κρίνεται η εφαρμογή της γνώσης που αποκτήθηκε για την προσαρμοστική μετάδοση βίντεο χρησιμοποιώντας το πρότυπο MPEG-DASH και της δυνατότητας που υπόσχεται ως προς την κατασκευή MPD αρχείων που θα περιέχουν και θα υποστηρίζουν κωδικοποιήσεις από διαφορετικά πρότυπα συμπίεσης ψηφιακού βίντεο.

Βιβλιογραφία

- [1] Cisco, Cisco Visual Networking Index: Forecast and Methodology, 2016-2021, 2017, White Paper C11-481360-01.
- [2] Thomas Stockhammer, "MPEG's Dynamic Adaptive Streaming over HTTP (DASH)," in *Webinar at EBU*, 2011.
- [3] (2018, November) Wikipedia. [Online].
https://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP
- [4] Victor Laskin. (2014, September) Victor Laskin's Blog. [Online]. <http://vitiy.info/is-4k-tv-effective-for-software-development/>
- [5] Philip A. Laplante, *Multimedia Image*, 2nd ed., LLC, Ed. Pennsylvania , USA: Taylor & Francis Group, 2012.
- [6] A.C.Bovik, *The Essential Guide to Video Processing*. Austin, Texas, 2009.
- [7] Stefano Soatto Georgios Georgiadis, "Exploiting Temporal Redundancy of Visual Structures for Video Compression," in *2015 Data Compression Conference, Snowbird, UT, USA* , 2015.
- [8] Γεώργιος Μινόπουλος, Τεχνολογίες Ανοικτού Κώδικα για τη Μετάδοση Βίντεο σε Ασύρματα Δίκτυα 4ης Γενιάς Χρησιμοποιώντας το Πρότυπο Κωδικοποίησης HEVC, 2016, Μεταπτυχιακή Διατριβή Ανοικτό Πανεπιστήμιο Κύπρου.
- [9] Ιάκωβος Ροδοθέου, Αξιολόγηση Προτύπων Συμπίεσης και Μετρικών Ποιότητας Βίντεο Χρησιμοποιώντας Λογισμικό Ανοικτού Κώδικα, 2017, Διπλωματική Εργασία Πανεπιστήμιο Κύπρου.
- [10] T.Sikora, "Digital Video Coding Standards and Their," in *Signal Processing For Multimedia.*, 1999, pp. 225-231.
- [11] B.A.Forouzan, *TCP/IP Protocol Suite*, 4th ed. New York, USA: The McGraw-Hill Companies, 2010.
- [12] Sachin Agarwal. (2010, Σεπτέμβριος) <https://sites.google.com/site/sachinkagarwal/home>. [Online]. <https://sites.google.com/site/sachinkagarwal/home/code-snippets/video-quality-analysis--part-1?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>
- [13] Μιχαήλ Ελευθεριάδης, Προσαρμογή Κωδικοποίησης για την Ασύρματη Μετάδοση Βίντεο σε Πραγματικό Χρόνο, 2017, Μεταπτυχιακή Διατριβή Ανοικτό Πανεπιστήμιο Κύπρου.
- [14] Tm Broadcast International. (2018, Μαρτίου) <http://tmbroadcast.com/>. [Online].

<http://tmbroadcast.com/index.php/bitmovin-video-streaming-tech-nab/>

- [15] Karl Soule. (2010, June) [Online].
http://blogs.adobe.com/VideoRoad/2010/06/what_is_yuv.html
- [16] Z. Antoniou, Y. Mylonas, M. S. Pattichis, A. Pitsillides, and C. S. Pattichis A. Panayides, "High-Resolution, Low-delay, and Error-resilient Medical Ultrasound Video Communication Using H.264/AVC Over Mobile WiMAX Networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 619-628, May 2013.
- [17] M. S. Pattichis, C. S. Pattichis, C. P. Loizou, M. Pantziaris, A. Pitsillides A. Panayides, "Robust and efficient ultrasound video coding in noisy channels using H.264," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 5143-5146.
- [18] A. S. Panayides, E. Christoforou, A. Jossif, S. Voskarides, C. Novales, L. Nouaille, C. S. Pattichis, P. Vieyres S. Avgousti, "Cardiac Ultrasonography over 4G Wireless Networks using a Tele-operated Robot," *IET Healthcare Technology Letters*, vol. 3, no. 3, pp. 212-217, September 2016.
- [19] Νάταλυ Ανδρέου, Προσαρμοστική Μετάδοση Ιατρικού Βίντεο σε Πραγματικό Χρόνο με την Χρήση του Προτύπου MPEG-DASH, 2016, Διπλωματική Εργασία Πανεπιστήμιο Κύπρου.
- [20] Nick Kounis, *Digital Quanta of Noise Shared with the World.*, 2015.
- [21] I.E.G. Richardson, *H.264 and MPEG-4 Video Compression*. Aberdeen, UK: The Robert Gordon University, 2003.
- [22] Heinrich Hertz Institute (HHI), Gary J. Sullivan, Microsoft Corporation Detlev Marpe and Thomas Wiegand, "The H.264/MPEG4 Advanced Video," *IEEE Communications Magazine*, August 2006.
- [23] Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand Garry J. Sullivan, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
- [24] A. Panayides, Z. Antoniou, M.S. Pattichis, C. S. Pattichis, and A. G. Constantinides, "High efficiency video coding for ultrasound video communication in m-health systems," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2012, pp. 2170-2173.
- [25] A. G. Constantinides, M. S. Pattichis, E. Kyriacou, C. S. Pattichis A. S. Panayides, "Adaptive real-time HEVC encoding of emergency scenery video," in *2014 4th International Conference on Wireless Mobile Communication and Healthcare - Transforming Healthcare Through Innovations in Mobile and Wireless Technologies (MOBIHEALTH)*, 2014, p. 217.
- [26] C. P. Loizou, M.S. Pattichis, E. Kyriacou, C. N. Shizas, A. N. Nicolaidis, C. S. Pattichis A. Panayides, "Ultrasound video despeckle filtering for High Efficiency Video Coding in m-health systems," in *2013 Constantinides International Workshop on Signal Processing (CIWSP*

2013), 2013, pp. 1-4.

- [27] Z. Antoniou, M.S. Pattichis, C. S. Pattichis A. Panayides, "The use of H.264/AVC and the emerging high efficiency video coding (HEVC) standard for developing wireless ultrasound video telemedicine systems," in *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2012, pp. 337-341.
- [28] M. S. Pattichis, C. P. Loizou, M. Pantziaris, A. G. Constantinides, C. S. Pattichis A. S. Panayides, "An Effective Ultrasound Video Communication System Using Despeckle Filtering and HEVC," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 668-676, March 2015.
- [29] M.S. Pattichis, C. S. Pattichis A. Panayides, "HEVC encoding for reproducible medical ultrasound video diagnosis," in *2013 Asilomar Conference on Signals, Systems and Computers*, 2013, pp. 1117-1121.
- [30] Δαμιανός Καλανίδης, Συγκριτική μελέτη και υλοποίηση κωδικοποιητών βίντεο με τα στάνταρ H.264 και VP8, 2013, Διπλωματική Εργασία Πανεπιστήμιο Θεσσαλίας.
- [31] Debargha Mukherjee, "A Technical Overview of VP9--the Latest Open-Source Video Codec," *SMPTE Motion Imaging Journal*, vol. 124, no. 1, pp. 44-54, February 2015.
- [32] O.P.T.I.C Maxim P. Sharabayko, "Intra Compression Efficiency in VP9 and HEVC," *Applied Mathematical Sciences*, vol. 6, no. 137, pp. 6812-6818, November 2013.
- [33] AOMedia. [Online]. <https://aomedia.org/membership/members/>
- [34] Debargha Murherjee, Jingning Han, Adrian Grange, Yaowu Xu, Zoe Liu, Sarah Parker, Cheng Chen, Hui Su, Urvang Joshi, Ching-Han Chiang, Yunqing Wang, Paul Wilkins, Jim Bankoski, Luc Trudeau, Nathan Egge, Jean-Marc Valin, Thomas Davies. Yue Chen, "An Overview of Core Coding Tools in the AV1 Video Codec," in *2018 Picture Coding Symposium (PCS)*, San Francisco, CA, USA, 2018.
- [35] 4G Americas. [Online]. http://anisimoff.org/eng/white_papers/supporting_wireless_video_growth_and_trends.html
- [36] Sebastian Egger, Martin Slanina, Thomas Zinner, Tobias Hobfeld and Phuoc Tran-Gia Michael Seufert, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communication Surveys & Tutorials*, vol. 17, no. 1, 2015.
- [37] Grenville Armitage and Philip Branch Jonathan Kua, "A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming over HTTP," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 10, 201x.
- [38] Anthony Vetro, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *Industry and Standards*, October 2011.
- [39] A. S. Panayides, M. Pantzaris, A. G. Constantinides, C. S. Pattichis and M. S. Pattichis Z. C. Antoniou, "Real-Time Adaptation to Time-Varying Constraints for Medical Video Communications," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 4, pp. 1177-

1188, July 2018.

- [40] M.S Pattichis and C.S. Pattichis A.S Panayides, "Mobile Health System Use Diagnostically Driven Medical Video Technologies," *IEEE Signal Processing Magazine*, vol. 30, no. 6, pp. 163-172, November 2013.
- [41] A.S.PAnayides, "Remote Diagnosis Leveraging the MPEG-DASH Standard," in *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Jeju Island, Korea, 2017.
- [42] M. Pattichis, C. Pattichis, C. Loizou, M. Pantziaris and A. Pitsillides A. Panayides, "Atherosclerotic plaque ultrasound video encoding, wireless transmission, and quality assessment using H.264," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 3, pp. 387-397, 2011.
- [43] Adobe. [Online]. <https://www.adobe.com/products/hds-dynamic-streaming.html>
- [44] (2018, August) <https://www.dacast.com/>. [Online]. <https://www.dacast.com/blog/hls-streaming-protocol/>
- [45] (2018) <http://www.iis.net>. [Online]. <https://www.iis.net/downloads/microsoft/smooth-streaming>
- [46] H.Q.Thu and M.Ghanbari, "Scope of validity of PSNR in image/video quality assesment," *IEEE*, vol. 44, pp. 800-801, 2008.
- [47] Mauro Barni, *Document and Image Compression.*: CRC Press, 2006.
- [48] FFmpeg. [Online]. <https://trac.ffmpeg.org/wiki/Encode/H.264>
- [49] FFmpeg A complete, cross-platform solution to record, convert and stream audio and video.. [Online]. <https://www.ffmpeg.org/>
- [50] YUV Player. [Online]. <http://www.yuvplayer.com/>
- [51] Netflix. [Online]. <https://research.netflix.com/>
- [52] A.S Panayides, Pattichis C.S, Stavros Stavrou Antoniou Zinonas, "Adaptive Emergency Scenery Video Communications using HEVC for Responsive Decision Support in Disaster Incidents," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society.*, Milano, 2015.
- [53] A. S. Panayides, M. S. Pattichis, S. Stavrou, E. Kyriacou, A. Spanias, A. G. Constantinides, C. S. Pattichis Z. Antoniou, "Adaptive emergency scenery video communications using HEVC for responsive decision support in disaster incidents," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 173-176.
- [54] Shiba Kuanar. [Online]. <https://slideplayer.com/slide/6022158/>
- [55] Adobe. [Online]. https://www.adobe.com/devnet/adobe-media-server/articles/dynstream_live/popup.html

- [56] G. Bjøntegaar, "Improvements of the BD-PSNR model," in *34th VCEG Meeting*, Berlin, 2008.
- [57] (2018, June) wikipedia. [Online]. https://en.wikipedia.org/wiki/Uncompressed_video
- [58] I.E.Richardson, *The H.264 advanced video compression standard*. United Kingdom: John Wiley & Sons Ltd, 2010.
- [59] Oleg G. Ponomarev, Roman I. Chernyak Maxim P. Sharabayko, "Intra compression efficiency in VP9 and HEVC," *Applied Mathematical Sciences*, vol. 7, no. 137-140, pp. 6803-6824, 2013.
- [60] Dominik Hübner and Sven Pfleiderer Thomas Maier. (2011) The VP8 Video Codec Multimedia Codecs SS 2011. [Online]. <http://www.slideshare.net/pfleidi/the-vp8-video-codec>
- [61] Kishore Kumar C, Praveen Kumar Reddy M Reuban Gnana Asir, "MPEG-DASH Enhanced Multimedia Streaming," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 3, March 2014.