



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Κωδικοποίηση και αποκωδικοποίηση με χρήση του
H.264/MPEG-4 για βίντεο με μία ή περισσότερες λήψεις**

Κωνσταντίνος Γ. Ζιάκας

Επιβλέπων: Εμμανουήλ Σαγκριώτης, Αναπληρωτής Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2011



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Κωδικοποίηση και αποκωδικοποίηση με χρήση του
H.264/MPEG-4 για βίντεο με μία ή περισσότερες λήψεις**

Κωνσταντίνος Γ. Ζιάκας

Επιβλέπων: Εμμανουήλ Σαγκριώτης, Αναπληρωτής Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2011

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κωδικοποίηση και αποκωδικοποίηση με χρήση του H.264/MPEG-4

Κωνσταντίνος Γ. Ζιάκας

M1076

ΕΠΙΒΛΕΠΩΝ:

Εμμανουήλ Σαγκριώτης, Αναπληρωτής Καθηγητής ΕΚΠΑ

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία περιγράφει την εξέλιξη του προτύπου H.264/MPEG-4 στον χρόνο. Δίνει ορισμούς και χρήσιμες εφαρμογές που αφορούν το H.264/MPEG-4.

Στο δεύτερο κεφάλαιο δίνονται κάποιες κωδικοποιήσεις και αποκωδικοποιήσεις βίντεο με μία λήψη (ουσιαστικά με μία κάμερα) με το πρότυπο H.264/MPEG-4 με την χρήση του lencod.exe, encoder.cfg, ldecod.exe, decoder.cfg.

Στο τρίτο κεφάλαιο δίνονται αντίστοιχες κωδικοποιήσεις και αποκωδικοποιήσεις αλλά με χρήση multiview βίντεο (δηλαδή με παραπάνω από μία κάμερες).

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Κωδικοποίηση με το H.264/MPEG-4.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: H.264/MPEG-4, multiview video, κωδικοποίηση και αποκωδικοποίηση βίντεο, βίντεο profiles, επίπεδα H.264.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	12
ΚΕΦΑΛΑΙΟ 1	
ΕΙΣΑΓΩΓΗ ΣΤΟ H.264/MPEG-4	13
1.1 Εισαγωγικές έννοιες για το H.264/MPEG-4	13
1.2 Επίπεδα του H.264 (H.264 Layers)	14
1.3 H.264 και MPEG-4 Ιστορική εξέλιξη	15
<i>1.3.1 Η δημιουργία του H.264/MPEG-4</i>	<i>15</i>
<i>1.3.2 Οι βελτιώσεις του H.264/MPEG-4</i>	<i>17</i>
1.4 Υποδειγματοληψία χρώματος (chroma subsampling)	18
1.4.1 4:4:4	19
1.4.2 4:2:2	20
1.4.3 4:1:1	21
1.4.4 4:2:0	22
1.5 Εκτίμηση κόστους και κίνησης (motion estimation και motion compensation).....	23
1.6 H.264/MPEG-4 Profiles	24
1.6.1 <i>Baseline Profile</i>	24
1.6.1.1 <i>Baseline Profile slices</i>	25
1.6.1.2 <i>Inter prediction/Intra prediction</i>	25
1.6.1.3 <i>CAVLC</i>	26
1.6.2 <i>Main Profile</i>	28
1.6.2.1 <i>B-slices</i>	28
1.6.2.2 <i>Πρόβλεψη με βάρη</i>	29
1.6.2.3 <i>Interlaced βίντεο</i>	30
1.6.2.4 <i>CABAC</i>	31
1.6.3 <i>Extended Profile</i>	32

1.6.3.1 <i>SP και Slices</i>	32
1.6.3.2 <i>Data partitioning slices</i>	33
1.6.3 <i>Υπόλοιπα Profiles</i>	34
1.7 Εργαλεία του H.264/MPEG-4	37

ΚΕΦΑΛΑΙΟ 2

ΥΛΟΠΟΙΗΣΗ ΠΑΡΑΔΕΙΓΜΑΤΩΝ ΓΙΑ ΤΟ H.264/MPEG-4 ΜΕ ΧΡΗΣΗ ΜΙΑΣ ΚΑΜΕΡΑΣ 39

2.1 Εισαγωγή	39
2.2 Χρήσιμα στοιχεία για την κωδικοποίηση και αποκωδικοποίηση	39
2.2.1 Όνομα βίντεο κατά την κωδικοποίηση και παραγόμενα αποτελέσματα	40
2.2.2 <i>Frames</i> κατά την κωδικοποίηση	40
2.2.3 <i>Profile</i> του βίντεο	41
2.2.4 Άλλες μεταβλητές κατά την κωδικοποίηση.....	41
2.2.5 Στοιχεία για την αποκωδικοποίηση.....	42
2.3 Αποτελέσματα κωδικοποιήσεων με χρήση μίας κάμερας.....	42
2.3.1 <i>Akiyo_cif</i>	43
2.3.2 <i>Coastguard_cif</i>	45
2.3.3 <i>Bowing_cif</i>	47
2.3.4 <i>Carphone_qcif</i>	49
2.3.5 <i>Door_Flowers</i>	51

ΚΕΦΑΛΑΙΟ 3

ΤΡΕΞΙΜΑΤΑ ΓΙΑ ΠΟΛΛΑΠΛΗΣ ΘΕΑΣΗΣ ΒΙΝΤΕΟ ΜΕ ΔΙΑΦΟΡΑ ΜΟΝΤΕΛΑ

ΑΝΑΦΟΡΑΣ..... 61

3.1 Εισαγωγή για Multiview	61
3.2 Χρήσιμοι ορισμοί	61
3.3 Σημαντικά ζητήματα για τα multiview βίντεο	62

3.4 Εφαρμογές των multiview βίντεο	63
3.4.1 3-D βίντεο.....	63
3.4.2 Βίντεο ελεύθερης θέασης	64
3.4.3 Απεικόνιση για μεγαλύτερη ευκρίνεια	64
3.5 Βασικά στοιχεία για την κωδικοποίηση και αποκωδικοποίηση multiview βίντεο	65
3.6 Μοντέλα αναφοράς.....	67
3.6.1 1 ^ο Μοντέλο αναφοράς	67
3.6.2 2 ^ο μοντέλο αναφοράς	69
3.6.3 3 ^ο μοντέλο αναφοράς	70
3.6.4 4 ^ο μοντέλο αναφοράς	71
3.7 Door_Flowers multiview	73
3.7.1 Door_Flowers με 1 ^ο Μοντέλο αναφοράς	74
3.7.2 Door_Flowers με 2 ^ο μοντέλο αναφοράς	77
3.7.3 Door_Flowers με 3 ^ο μοντέλο αναφοράς	80
3.7.4 Door_Flowers με 4 ^ο μοντέλο αναφοράς	83
3.8 Leaving_Laptop multiview	86
3.7.1 Leaving_Laptop με 1 ^ο Μοντέλο αναφοράς	87
3.7.2 Leaving_Laptop με 2 ^ο μοντέλο αναφοράς	90
3.7.3 Leaving_Laptop με 3 ^ο μοντέλο αναφοράς	93
3.7.4 Leaving_Laptop με 4 ^ο μοντέλο αναφοράς	96

ΚΕΦΑΛΑΙΟ 4

ΕΠΙΛΟΓΟΣ	99
ΑΝΑΦΟΡΕΣ	100

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1.1: Τυπικό διάγραμμα κωδικοποιητή για το πεδίο VCL του H.264/MPEG-4	96
Σχήμα 1.2: Η δημιουργία του H.264/MPEG-4.....	96
Σχήμα 1.3: Παράσταση μίας εικόνας και οι συνιστώσες Y,U και V κατά σειρά	96
Σχήμα 1.4: Υποδειγματοληψία χρώματος 4:4:4.....	96
Σχήμα 1.5: Υποδειγματοληψία χρώματος 4:2:2.....	96
Σχήμα 1.6: Υποδειγματοληψία χρώματος 4:1:1.....	96
Σχήμα 1.7: Υποδειγματοληψία χρώματος 4:2:0.....	96
Σχήμα 1.8: Κωδικοποίηση block μετασχηματισμένων συντελεστών.....	96
Σχήμα 1.9: Παράδειγμα πρόβλεψης σε ένα B macroblock	96
Σχήμα 1.10: Reordering σάρωση	96
Σχήμα 1.11: Παράδειγμα μίας ακολουθίας.....	96
Σχήμα 1.12: Σχέσεις μεταξύ Baseline, Main και Extended Profiles.....	96
Σχήμα 1.13: Hardware και Software που στηρίζεται στο H.264/MPEG-4	96
Σχήμα 2.1: Παρουσίαση αποτελεσμάτων για το SNR στο βίντεο akiyo_cif των συνιστωσών Y (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα).....	96
Σχήμα 2.2: Παρουσίαση αποτελεσμάτων για το SNR στο βίντεο container_cif των συνιστωσών Y (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα).....	96
Σχήμα 2.3: Παρουσίαση αποτελεσμάτων για το SNR στο βίντεο bowing_cif των συνιστωσών Y (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα).....	96
Σχήμα 2.4: Παρουσίαση αποτελεσμάτων για το SNR στο βίντεο carphone_qcif των συνιστωσών Y (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα).....	96
Σχήμα 2.5: Συνιστώσες SNR Y (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam01	96
Σχήμα 2.6: Συνιστώσες SNR Y (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam02	96
Σχήμα 2.7: Συνιστώσες SNR Y (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam03	96

Σχήμα 2.8: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam04	96
Σχήμα 2.9: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam05	96
Σχήμα 2.10: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam06	96
Σχήμα 2.11: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam07	96
Σχήμα 2.12: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam08	96
Σχήμα 2.13: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam09	96
Σχήμα 2.14: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam10	96
Σχήμα 2.15: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam11	96
Σχήμα 2.16: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam12	96
Σχήμα 2.17: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam13	96
Σχήμα 2.18: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam14	96
Σχήμα 2.19: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam15	96
Σχήμα 2.20: Συνιστώσες SNR Υ (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam16	96
Σχήμα 3.1: Παράδειγμα μοντέλου αναφοράς ανά 8 πλάνα (frames)	96
Σχήμα 3.2: 3-D γυαλιά απαραίτητα για να δει κάποιος ένα 3-D βίντεο	96
Σχήμα 3.3: 1 ^ο reference mode	96
Σχήμα 3.4: 2 ^ο reference mode	96

Σχήμα 3.5: 3 ^ο reference mode	96
Σχήμα 3.6: 4 ^ο reference mode	96
Σχήμα 3.7: SNR για το Door Flowers για 8 βίντεο χρησιμοποιώντας το 1 ^ο reference mode. Με το μπλε το SNR της Υ συνιστώσας, με κόκκινο της U συνιστώσας και με πράσινο της V.....	96
Σχήμα 3.8: SNR για το Door Flowers για 8 βίντεο χρησιμοποιώντας το 2 ^ο reference mode. Με το μπλε το SNR της Υ συνιστώσας, με κόκκινο της U συνιστώσας και με πράσινο της V.....	96
Σχήμα 3.9: SNR για το Door Flowers για 8 βίντεο χρησιμοποιώντας το 3 ^ο reference mode. Με το μπλε το SNR της Υ συνιστώσας, με κόκκινο της U συνιστώσας και με πράσινο της V.....	96
Σχήμα 3.10: SNR για το Door Flowers για 8 βίντεο χρησιμοποιώντας το 4 ^ο reference mode. Με το μπλε το SNR της Υ συνιστώσας, με κόκκινο της U συνιστώσας και με πράσινο της V.....	96
Σχήμα 3.11: SNR για το Leaving_Laptop για 8 βίντεο χρησιμοποιώντας το 1 ^ο reference mode. Με το μπλε το SNR της Υ συνιστώσας, με κόκκινο της U συνιστώσας και με πράσινο της V	96
Σχήμα 3.12: SNR για το Leaving_Laptop για 8 βίντεο χρησιμοποιώντας το 2 ^ο reference mode. Με το μπλε το SNR της Υ συνιστώσας, με κόκκινο της U συνιστώσας και με πράσινο της V	96
Σχήμα 3.13: SNR για το Leaving_Laptop για 8 βίντεο χρησιμοποιώντας το 3 ^ο reference mode. Με το μπλε το SNR της Υ συνιστώσας, με κόκκινο της U συνιστώσας και με πράσινο της V	96
Σχήμα 3.14: SNR για το Leaving_Laptop για 8 βίντεο χρησιμοποιώντας το 4 ^ο reference mode. Με το μπλε το SNR της Υ συνιστώσας, με κόκκινο της U συνιστώσας και με πράσινο της V	96

ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία αποτελεί την διπλωματική μου εργασία, κατά την παραμονή μου στο τμήμα Πληροφορικής και Τηλεπικοινωνιών σαν μεταπτυχιακός φοιτητής της 5^{ης} κατεύθυνσης (Επεξεργασία Σήματος για Επικοινωνίες και Πολυμέσα).

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον αναπληρωτή καθηγητή Εμμανουήλ Σαγκριώτη για την όλη συνεργασία στον καιρό περάτωσης της διπλωματικής μου εργασίας. Επίσης θα ήθελα να ευχαριστήσω τους γονείς μου για την βοήθεια και την στήριξη που μου έχουν προσφέρει πολλά χρόνια.

Ο συγγραφέας,

Κωνσταντίνος Ζιάκας

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ ΣΤΟ H.264/MPEG-4

1.1. Εισαγωγικές έννοιες για το H.264/MPEG-4

Το H.264/MPEG-4 είναι ένα πρότυπο που χρησιμοποιείται για συμπίεση βίντεο, και είναι ένα από τα πιο χρήσιμα πρότυπα για εγγραφή, συμπίεση και διαμοιρασμό υψηλής ευκρίνειας βίντεο (high definition video). Η τελική εργασία για το H. 264/MPEG-4 είναι το Part 10 στις διάφορες εκδόσεις και ολοκληρώθηκε το 2003.

Το H.264/ MPEG-4 είναι ένα πρότυπο που βασίζεται σε προσανατολισμένη κίνηση (motion compensated) που αναπτύχθηκε από την ITU-T Video Coding Experts (VCEG) μαζί με την ISO/IEC Moving Pictures Experts Group (MPEG). Το αποτέλεσμα αυτής της κοινής σύμπραξης είναι γνωστό σαν Joint Video Team (JVT).

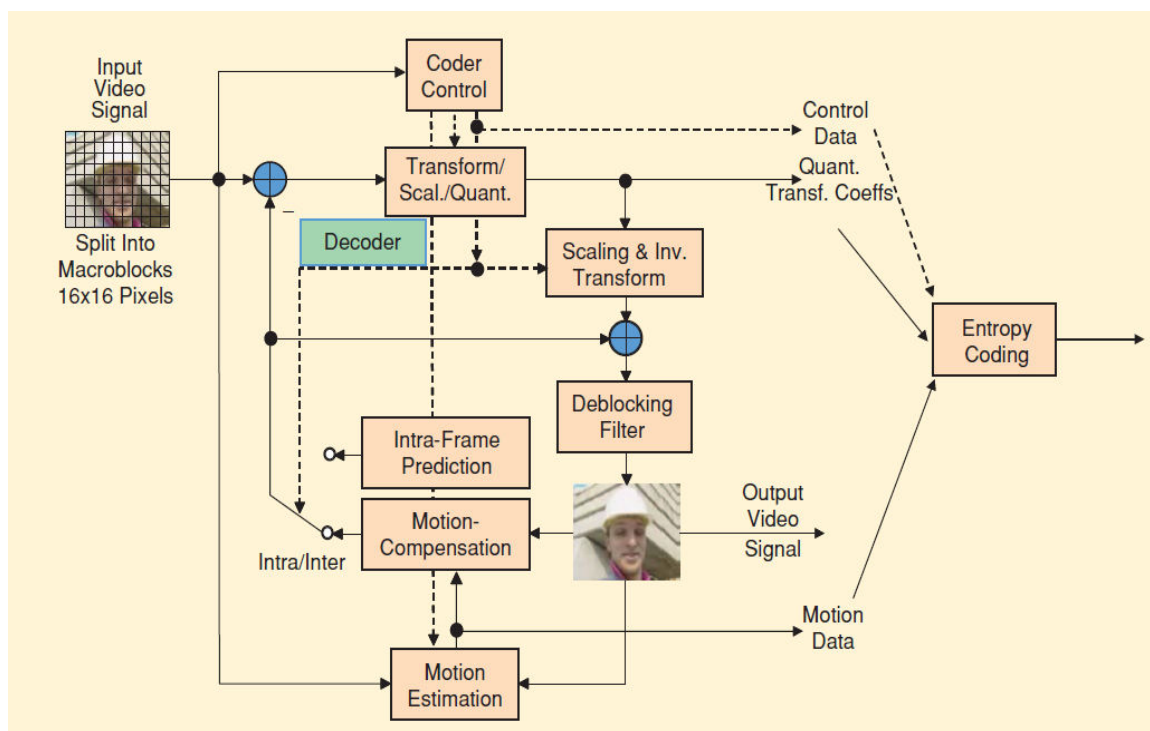
Βασικές χρήσεις που έχει το H.264/MPEG-4 είναι οι Blu-Ray δίσκοι αφού όλες οι συσκευές που μπορούν να «παίζουν» Blu-Ray δίσκοι θα πρέπει να μπορούν να αποκωδικοποιήσουν το H.264. Επίσης χρησιμοποιείται για μετάδοση σε βίντεο του διαδικτύου από site όπως Vimeo, Youtube, itunes Store και σε άλλα. Άλλες χρήσεις είναι για μετάδοση σήματος τηλεόρασης μέσω δορυφόρου, σε υπηρεσίες καλωδιακής τηλεόρασης και ίσως το πιο σημαντικό όλων για ζωντανή μετάδοση τηλεμετάδοσης (real time videoconferencing).

1.2. Επίπεδα του H264 (H264 Layers)

Το H264 αποτελείται από δύο επίπεδα: ένα επίπεδο αφαίρεσης δικτύου (network abstraction layer NAL) και από ένα επίπεδο κωδικοποίησης βίντεο (video coding layer VCL).

Το επίπεδο αφαίρεσης δικτύου (NAL) είναι ένα επίπεδο του H.264 Video Coding Standard. Ο κύριος στόχος αυτού του επιπέδου είναι να παρέχει μία «φιλική για το δίκτυο (network-friendly)» αναπαράσταση βίντεο τόσο για εφαρμογές που έχουν ομιλία όσο και για εφαρμογές που είναι για αποθήκευση ή απλή μετάδοση. Το επίπεδο NAL έχει πετύχει σημαντική βελτίωση στην αποδοτικότητα του ποσοστού παραμόρφωσης (rate-distortion efficiency) σε σχέση με ότι ίσχυε.

Οι μονάδες του NAL ταξινομούνται σε VCL και non-VCL NAL μονάδες. Οι VCL μονάδες περιέχουν τα δεδομένα που αντιπροσωπεύουν τις τιμές των δειγμάτων στις εικόνες και οι non-VCL NAL μονάδες περιλαμβάνουν οποιεσδήποτε συμπληρωματικές πληροφορίες καθώς και διάφορες συμπληρωματικές πληροφορίες όπως χρονισμό που μπορούν να ενισχύσουν την χρηστικότητα του αποκωδικοποιημένου σήματος βίντεο και δεν είναι απαραίτητες για την αποκωδικοποίηση των τιμών των δειγμάτων που έχουμε στην VCL μονάδα.



Σχήμα 1.1: Τυπικό διάγραμμα κωδικοποιητή για το πεδίο VCL του H.264/MPEG-4.

1.3. H.264 και MPEG-4 Ιστορική εξέλιξη

Η πρώτη έκδοση του H.264/MPEG-4 εμφανίστηκε το 2003. Σε αυτή την ενότητα θα παρουσιαστούν τόσο κάποιες εκδόσεις πριν από την εμφάνιση του H.264, όσο και το πως εξελίχθηκε το H.264 με κάποιες βελτιώσεις του.

1.3.1. Η δημιουργία του H.264/MPEG-4

- 1993: Έχουμε την έκδοση του MPEG-4. Πρώτα αποτελέσματα με το H-263.
- 1995: Το MPEG-4 ζητάει προτάσεις για κωδικοποιητές βίντεο. Το H-263 διαλέχτηκε σαν κωδικοποιητής βίντεο.
- 1998: Κλήση για προτάσεις για το H.26L.

- 1999: Το MPEG-4 Visual εκδόθηκε. Έχουμε την έκδοση επίσης και του Initial Test Model (TMI) του H.26L.
- 2000: Το MPEG καλεί για προτάσεις για προηγμένους κωδικοποιητές βίντεο.
- 2001: Έχουμε την δεύτερη έκδοση του MPEG-4 Visual. Το H.26L κρατήθηκε σαν βάση για το MPEG-4.
- 2002: Έχουμε διάφορες τροποποιήσεις με τις οποίες εκδίδεται το MPEG-4 Visual Edition 2.
- 2003: Έχουμε την έκδοση του H.264/MPEG-4 Part 10.

Table 4.2 MPEG-4 and H.264 development history

1993	MPEG-4 project launched. Early results of H. 263 project produced.
1995	MPEG-4 call for proposals including efficient video coding and content-based functionalities. H.263 chosen as core video coding tool
1998	Call for proposals for H.26L.
1999	MPEG-4 Visual standard published. Initial Test Model (TMI) of H.26L defined.
2000	MPEG call for proposals for advanced video coding tools.
2001	Edition 2 of the MPEG-4 Visual standard published. H.26L adopted as basis for proposed MPEG-4 Part 10. JVT formed.
2002	Amendments 1 and 2 (Studio and Streaming Video profiles) to MPEG-4 Visual Edition 2 published. H.264 technical content frozen.
2003	H.264/MPEG-4 Part 10 ('Advanced Video Coding') published.

Σχήμα 1.2: Η δημιουργία του H.264/MPEG-4.

1.3.2. Οι βελτιώσεις του H.264/MPEG-4

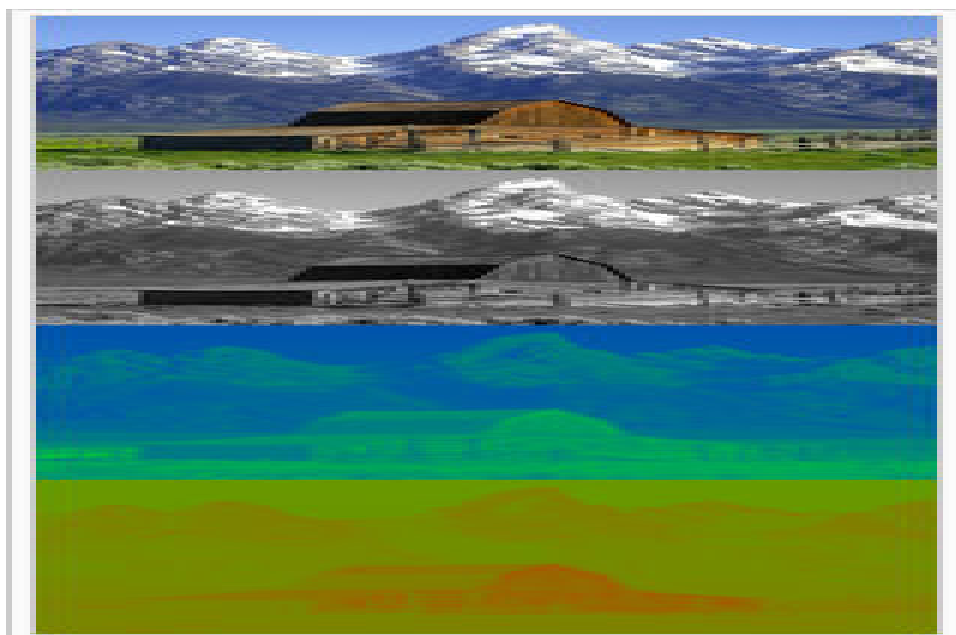
Μετά από το 2003 το H.264/MPEG-4 συνέχισε να βελτιώνεται. Ας δούμε αυτές τις βελτιώσεις:

- Μάιος 2003: Πρώτη έκδοση του H.264/MPEG-4 με τα Baseline, Main και Extended Profiles να περιλαμβάνονται.
- Μάιος 2004: Νέα έκδοση με διορθώσεις πολλών μικρών λαθών.
- Μάρτιος 2005: Η πρώτη μεγάλη εξέλιξη που περιλαμβάνει τα High, High 10, High 4:2:2 και High 4:4:4 Profiles.
- Σεπτέμβριος 2005: Νέα έκδοση με διορθώσεις πολλών μικρών λαθών.
- Ιούνιος 2006: Εξέλιξη με την απομάκρυνση του High 4:4:4 Profile. Την ίδια περίοδο επίσης είχαμε την ένταξη στο H.264/MPEG-4 περισσότερων μεθόδων υποδειγματοληψίας χρώματος.
- Απρίλιος 2007: Εξέλιξη με την ένταξη Intra-Profiles: High 10 Intra, High 4:2:2 Intra, High 4:4:4 Intra και CAVLC 4:4:4 Intra Profiles.
- Νοέμβριος 2007: Μεγάλη εξέλιξη στο H.264/MPEG-4 με την ένταξη scalable profiles όπως τα Scalable Baseline, Scalable High και Scalable High Intra Profiles.
- Ιανουάριος 2009: Νέα έκδοση με διορθώσεις πολλών μικρών λαθών.
- Μάρτιος 2009: Μεγάλες εξελίξεις για το H.264/MPEG-4. Αρχικά έχουμε ένταξη νέου Profile του Constrained Baseline Profile και έπειτα με την ένταξη του multiview video coding (MVC).
- Μάρτιος 2010: Έχουμε την ένταξη νέου profile αποκλειστικά για το MVC του Stereo High Profile.

Όλα αυτά τα Profiles θα τα δούμε αναλυτικότερα σε μετέπειτα ενότητα αυτού του κεφαλαίου.

1.4. Υποδειγματοληψία χρώματος (Chroma subsampling)

Σε αυτό το σημείο είναι ενδιαφέρον να δούμε κάποια πράγματα για την υποδειγματοληψία χρώματος. Κατά την παράσταση της εικόνας έχουμε 3 συνιστώσες: τις Y, U και V (παρουσιάζονται και σαν Y, C_b, C_r). Η βασική συνιστώσα είναι η Y (luminance) την οποία και αντιλαμβάνεται το ανθρώπινο μάτι. Σε αυτή έχουμε και το μεγαλύτερο ποσοστό πληροφορίας που θέλουμε για την εικόνα. Στις άλλες δύο έχουμε ένα ποσοστό πληροφορίας που αφορά την «χρωματικότητα» (chrominance) της εικόνας. Για το ποσοστό πληροφορίας που παίρνουμε από κάθε συνιστώσα μπορούμε να κοιτάξουμε το σχήμα 1.3.



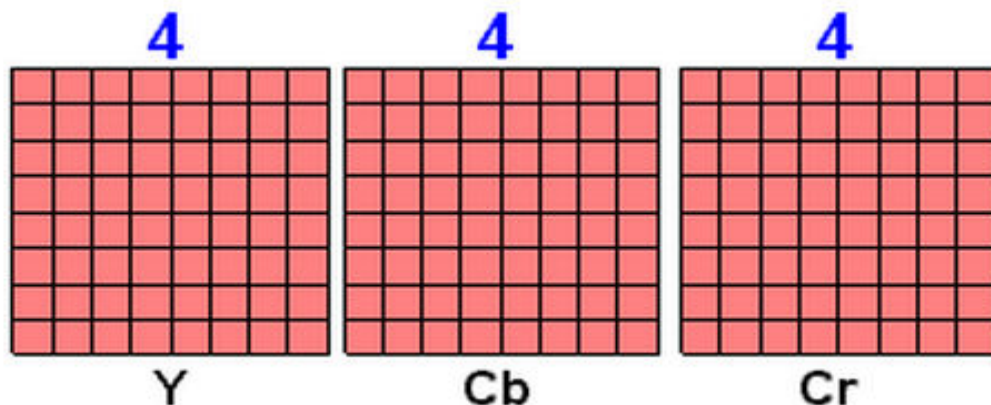
Σχήμα 1.3: Παράσταση μίας εικόνας και οι συνιστώσες Y, U και V κατά σειρά.

Για να θεωρήσουμε υποδειγματοληψία χρώματος ουσιαστικά θα πρέπει να σκεφτόμαστε τις Y, U και V σαν ξεχωριστούς πίνακες με τιμές από τα pixels της εικόνας. Κατά την υποδειγματοληψία παίρνουμε κάποιες από αυτές τις τιμές ανάλογα με το είδος υποδειγματοληψίας που έχουμε διαλέξει (από ότι θα

δούμε και αργότερα στα βίντεο που έχουν κωδικοποιηθεί έχει επιλεγεί 4:2:0 υποδειγματοληψία). Καλό θα είναι να λαμβάνεται υπόψη ότι η συνιστώσα της φωτεινότητας είναι αυτή που λαμβάνει υπόψη το ανθρώπινο μάτι, οπότε η υποδειγματοληψία δεν είναι πάντα ενδεδειγμένη για αυτή και μόνο αυτή την συνιστώσα. Ας δούμε τώρα κάποια από τα πιο βασικά είδη υποδειγματοληψίας (για καλή κατανόηση του κάθε είδους υποδειγματοληψίας χρώματος έχουμε και εικόνες για μερικά από αυτά):

1.4.1. 4:4:4

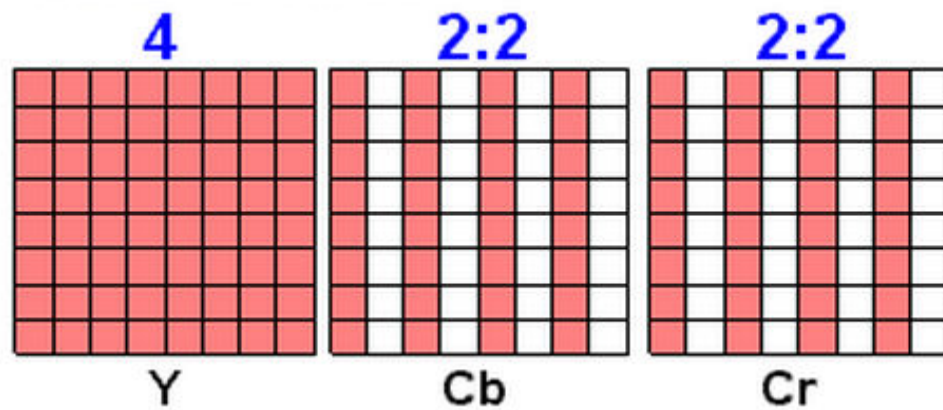
Σε αυτό το είδος υποδειγματοληψίας χρώματος για κάθε συνιστώσα έχουμε το ίδιο ποσοστό υποδειγματοληψίας. Χρησιμοποιείται σε σαρωτές υψηλής ευκρίνειας καθώς και στον κινηματογράφο κατά το μοντάζ.



Σχήμα 1.4: Υποδειγματοληψία χρώματος 4:4:4.

1.4.2. 4:2:2

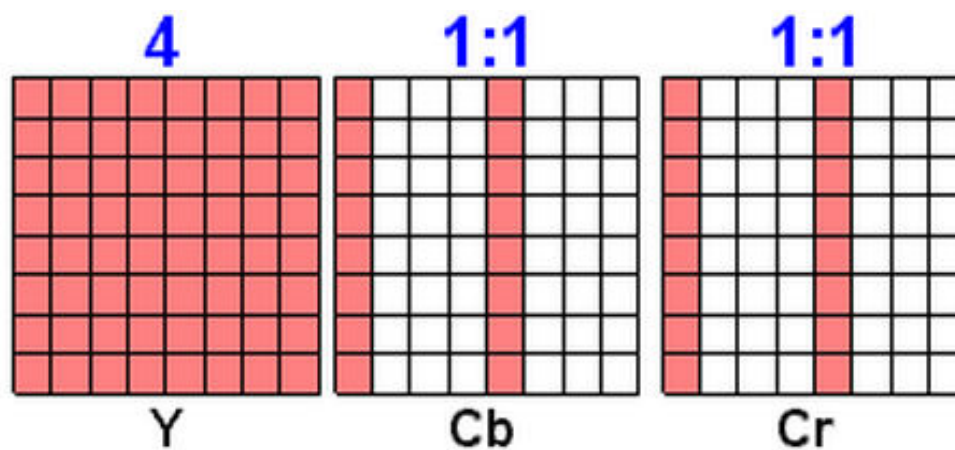
Σε αυτό το είδος υποδειγματοληψίας χρώματος, ενώ η Υ συνιστώσα μένει αναλλοίωτη οι άλλες δύο κωδικοποιούνται στο μισό από ότι ήταν αρχικά οριζόντια. Αυτό το είδος υποδειγματοληψίας χρώματος είναι ιδιαίτερα διαδεδομένο στην εγγραφή μεγάλης ευκρίνειας βίντεο.



Σχήμα 1.5: Υποδειγματοληψία χρώματος 4:2:2.

1.4.3. 4:1:1

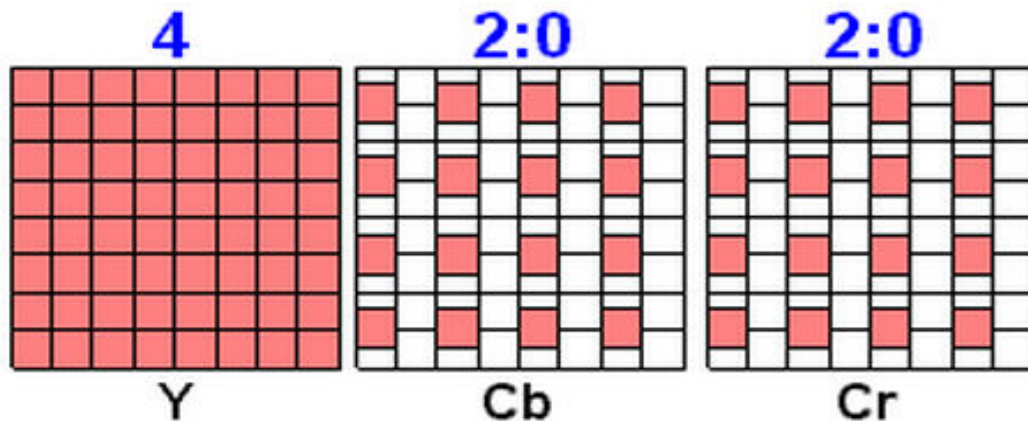
Σε αυτό το είδος υποδειγματοληψίας χρώματος η Υ συνιστώσα μένει αναλοιώτη και πάλι, αλλά οι άλλες δύο οι U και οι V κωδικοποιούνται στο $\frac{1}{4}$ σε σχέση με το αρχικό οριζόντια. Χρησιμοποιείται σε DV, DVCAM και DVCPRO πρότυπα.



Σχήμα 1.6: Υποδειγματοληψία χρώματος 4:1:1.

1.4.4. 4:2:0

Αυτό το είδος υποδειγματοληψίας χρώματος είναι και αυτό που χρησιμοποιείται κατά βάση στα βίντεο που κωδικοποιήσαμε. Σε αυτό το είδος υποδειγματοληψίας η Υ συνιστώσα μένει αμετάβλητη αλλά οι άλλες κωδικοποιούνται τόσο οριζόντια όσο και κάθετα.



Σχήμα 1.7: Υποδειγματοληψία χρώματος 4:2:0.

Το 4:2:0 χρησιμοποιείται σε αρκετές εφαρμογές όπως στο PAL DV και DVCAM, στο HDV, στο AVCHD και AVC-Intra 50 και άλλα.

Υπάρχουν και άλλα είδη υποδειγματοληψίας χρώματος ίσως λίγο λιγότερο διαδεδομένα όπως το 4:1:0 ή το 3:1:1 που παρουσιάζει ενδιαφέρον λόγω της υποδειγματοληψίας που γίνεται και στην συνιστώσα της φωτεινότητας. Αυτό το είδος χρησιμοποιείται σε εγγραφείς υψηλής ευκρίνειας.

1.5. Εκτίμηση κίνησης και κόστους (motion estimation and motion compensation)

Το motion compensation και motion estimation είναι ουσιαστικά δύο τεχνικές που εφαρμόστηκαν στους κωδικοποιητές του H.264/MPEG-4 που και οι δύο είχαν ως αποτέλεσμα την βελτίωση της κωδικοποίησης του προτύπου. Ας δούμε συνοπτικά μερικά πράγματα για το καθένα:

- Motion compensation είναι μία αλγοριθμική τεχνική που εφαρμόστηκε στην κωδικοποίηση δεδομένων για συμπίεση βίντεο. Το motion compensation ουσιαστικά περιγράφει μία εικόνα σαν μετασχηματισμό από το πλάνο αναφοράς (reference frame) στο τωρινό πλάνο (current frame). Όταν οι επόμενες εικόνες θα μπορούν να συντίθενται από εικόνες που ήδη έχουν συγκριθεί και βρεθεί ότι έχουν ελάχιστες διαφορές θα μπορούμε να έχουμε πολύ καλύτερα αποτελέσματα στην συμπίεση μας. Αυτός είναι ο ρόλος του motion compensation στο H.264/MPEG-4.
- Το motion estimation είναι η διαδικασία στην οποία θα καθορίζονται διανύσματα κίνησης από ένα προηγούμενο πλαίσιο με το τωρινό. Αυτό το διάνυσμα κίνησης ουσιαστικά θα μας δείχνει τις διαφορές μεταξύ πλαισίων και έτσι μπορούμε να κρατάμε μόνο την διαφορά και όχι όλο το πλαίσιο. Συνεπώς μειώνεται πολύ ο χώρος που χρειαζόμαστε για αποθήκευση.

Οπότε σαν αποτέλεσμα τόσο του motion compensation όσο και του motion estimation είναι η καλύτερη αποθήκευση και συμπίεση της κωδικοποιημένης ακολουθίας βίντεο που έχουμε.

1.6. H.264/MPEG-4 Profiles

Το H.264 έχει καθορίσει κάποια Profiles, τα οποία ανάλογα με την επιλογή τους, επιτελούν και διάφορες συναρτήσεις κατά την εκτέλεση τους. Τα πιο σημαντικά Profiles είναι το Baseline, το Extended και τα Main Profile. Στο Baseline και στο Extended Profile ιδιαίτερο ενδιαφέρον έχουν οι συναρτήσεις CAVLC και CABAC που χρησιμοποιούνται αντίστοιχα. Σε αυτή την ενότητα θα παρουσιάσουμε και τα 17 Profiles που μπορούν να χρησιμοποιηθούν από το H.264.

1.6.1. Baseline Profile

Το baseline profile υποστηρίζει κωδικοποιημένες σειρές που αποτελούνται από I και P κομμάτια (I και P slices). Τα I κομμάτια περιέχουν intra-coded macroblocks (MB), στα οποία περιοχές φωτεινότητας 16X16 ή 4X4 και περιοχές χρώματος 8X8 κωδικοποιούνται από προηγούμενα δείγματα του ίδιου κομματιού. Τα P κομμάτια περιέχουν intra-coded, inter-coded macroblocks ή macroblock που έχουν προσπεραστεί. Τα inter-coded MB σε ένα P κομμάτι έχουν προβλεφθεί από έναν αριθμό προηγούμενων κωδικοποιημένων εικόνων, έχοντας χρησιμοποιήσει MC (motion compensation).

Μετά την πρόβλεψη τα υπολοιπόμενα στοιχεία από κάθε MB μετατρέπονται βασιζόμενα σε ένα DCT μετασχηματισμό (integer transform) και μετά κβαντίζονται. Οι κβαντισμένοι συντελεστές αναδιατάσσονται και τα στοιχεία σύνταξης κωδικοποιούνται «εντροπικά». Αυτή η εντροπική κωδικοποίηση για τους συντελεστές στο baseline profile, γίνεται χρησιμοποιώντας CAVLC. Όλα τα άλλα στοιχεία σύνταξης που δεν έχουν υποστεί την ίδια διαδικασία κωδικοποιούνται με σταθερού μήκους κωδικοποίηση. Οι κβαντισμένοι συντελεστές κλιμακώνονται, αντιστρέφονται, αναδομούνται και έπειτα αποθηκεύονται για μελλοντική χρήση από intra-coded ή inter-coded MB.

1.6.1.1. Baseline profile slices

Έχει ιδιαίτερο ενδιαφέρον να δούμε τι κομμάτια χρησιμοποιούνται στο baseline profile και αντίστοιχα στα επόμενα 2 βασικά profile για το H.264. Στο baseline profile έχουμε μόνο I ή/και P κομμάτια. Δεν έχουμε τα διπλής κατεύθυνσης B καθόλου. Ένα I κομμάτι περιέχει μόνο intra-coded MB, το οποίο έχει κωδικοποιηθεί χρησιμοποιώντας μόνο πληροφορία από το προηγούμενο I αν έχει υπάρξει. Ένα P κομμάτι περιέχει inter-coded πληροφορία, χρησιμοποιώντας δείγματα που έχουν ήδη κωδικοποιηθεί.

Στο baseline profile μπορούμε επίσης να έχουμε και την χρήση των λεγόμενων «ομάδων κομματιών»(slices groups). Σε αυτές τις ομάδες έχουμε ένα υποσύνολο MB, τα οποία μπορεί να περιέχουν και παραπάνω από ένα κομμάτι. Αυτό βοηθάει αρκετά τόσο των κωδικοποίηση όσο και την αποκωδικοποίηση. Την κωδικοποίηση γιατί υπάρχει μεγαλύτερη ελευθερία κατά την κωδικοποίηση των MB, κάτι το οποίο ισχύει και για την αποκωδικοποίηση. Απλά για την αποκωδικοποίηση θα πρέπει να έχουμε προσέξει κατά την κωδικοποίηση να τοποθετήσουμε κατάλληλα σημάδια στα MB ώστε να γίνει σωστά, χωρίς επικάλυψη.

1.6.1.2. Inter prediction/Intra-prediction

Έχουμε δει μέχρι αυτό το σημείο αρκετές φορές τις έννοιες inter-coded και intra-coded. Καλό θα είναι να αναφέρουμε τις έννοιες αυτές καλύτερα ώστε να είναι δυνατόν να συνδυαστούν τόσο με προηγούμενα κομμάτια, όσο και με επόμενα για τα extended και main profile, αφού ισχύουν παρόμοια πράγματα.

Κατά το inter-prediction έχουμε ουσιαστικά ένα μοντέλο πρόβλεψης για ένα frame από προηγούμενα frame ή frames που έχουν ήδη κωδικοποιηθεί για το ίδιο βίντεο. Διαφορές από άλλα profiles βρίσκονται στο γεγονός ότι εδώ

έχουμε μία ευρύτητα από μεγέθη block που χρησιμοποιούνται (από 16X16 ως και 4X4) καθώς και στα διανύσματα κίνησης που χρησιμοποιούμε.

Αντίστοιχα intra-prediction είναι το μοντέλο πρόβλεψης όπου βάση ήδη κωδικοποιημένα frames προβλέπουμε κωδικοποίηση που θα χρειαστεί για επόμενα.

1.6.1.3. CAVLC

Το CAVLC είναι τα αρχικά για το context-based adaptive variable length encoding. Είναι μία τεχνική ιδιαίτερα σημαντική για την κωδικοποίηση baseline profile, αλλά και για τα υπόλοιπα profiles, αφού αποτελούν υπερσύνολο του baseline. Το CAVLC είναι μία μέθοδος για κωδικοποίηση υπολοίπων, zig-zag και 4X4 ή 2X2 blocks των συντελεστών μετασχηματισμού. Το CAVLC είναι σχεδιασμένο ώστε να εκμεταλλεύεται τα πλεονεκτήματα των 4X4 κβαντισμένων block:

- 1) Μετά από την πρόβλεψη, μετασχηματισμό και κβάντιση το block αποτελείται κυρίως από μηδενικά. Έτσι το CAVLC χρησιμοποιεί run-length κωδικοποίηση ώστε να μπορέσει να ομαδοποιήσει αυτά τα μηδενικά.
- 2) Οι υψηλότεροι συντελεστές μετά το zig-zag και πέρα από τα μηδενικά είναι +-1 και έτσι το CAVLC προσπαθεί να τα παραστήσει με ένα πιο συμπαγή τρόπο.
- 3) Ο αριθμός των μη μηδενικών σε γειτονικά block πολλές φορές συσχετίζεται. Έτσι για τον αριθμό των συντελεστών χρησιμοποιείται ένας look-up πίνακας για να μην υπάρχει ο συσχετισμός αυτός.
- 4) Το πλήθος των μη μηδενικών συντελεστών τείνει να είναι μεγαλύτερος στην αρχή του πίνακα. Ο CAVLC προσπαθεί να το κάνει προς συμφέρον του προσαρμόζοντας τον look-up πίνακα.

Η κωδικοποίηση στον CAVLC ενός block μετασχηματισμένων συντελεστών πηγαίνει ως ακολούθως:

- 1) Κωδικοποίηση του αριθμού των συντελεστών.
- 2) Κωδικοποιείται ότι υπολείπεται.
- 3) Κωδικοποίηση του επιπέδου όλων των μη μηδενικών συντελεστών.
- 4) Κωδικοποίηση του συνολικού αριθμού των μηδενικών πριν από τον τελευταίο συντελεστή.
- 5) Κωδικοποίηση όλων των μηδενικών.

coeff_token	encodes the number of non-zero coefficients (TotalCoeff) and TrailingOnes (one per block)
trailing_ones_sign_flag	sign of TrailingOne value (one per trailing one)
level_prefix	first part of code for non-zero coefficient (one per coefficient, excluding trailing ones)
level_suffix	second part of code for non-zero coefficient (not always present)
total_zeros	encodes the total number of zeros occurring after the first non-zero coefficient (in zig-zag order) (one per block)
run_before	encodes number of zeros preceding each non-zero coefficient <i>in reverse zig-zag order</i>

Σχήμα 1.8 : Κωδικοποίηση block μετασχηματισμένων συντελεστών.

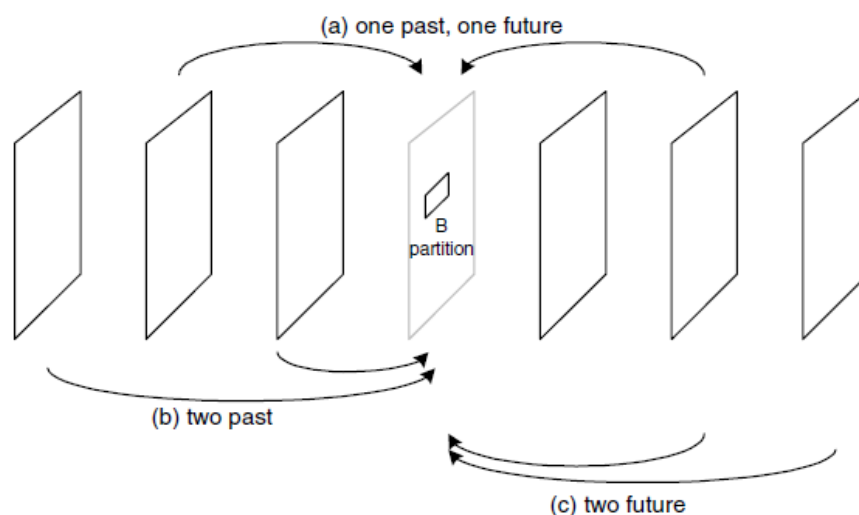
1.6.2. Main Profile

Κατάλληλες εφαρμογές για το Main Profile περιλαμβάνουν (αλλά δεν περιορίζονται σε αυτές) την μετάδοση σήματος για ψηφιακή τηλεόραση και για αναλογικό σήμα. Το Main Profile είναι κατά ένα σημείο υπερσύνολο του Baseline Profile. Μόνο πολλαπλά slices groups και υπολειπόμενα slices δεν υποστηρίζονται στο Main Profile, αλλά υποστηρίζονται στο Baseline. Αυτό φαίνεται και στο σχήμα 1.12.

Τα πρόσθετα εργαλεία που υποστηρίζονται από το Main Profile είναι τα B slices (δηλαδή διπλής κατεύθυνσης), βάρη στην πρόβλεψη, υποστήριξη interlaced βίντεο και CABAC (μία εναλλακτική μέθοδο εντροπίας που βασίζεται στην αριθμητική κωδικοποίηση).

1.6.2.1. B-slices

Σε ένα inter-coded macroblock ένα B-slice μπορεί να προβλεφθεί από μία ή δύο εικόνες αναφοράς, πριν ή μετά από την εικόνα που είμαστε σε μία συγκεκριμένη χρονική στιγμή. Δίνονται διάφορες επιλογές στον κωδικοποιητή, ανάλογα από τις αναφορές προβλέψεων που έχουν ήδη κρατηθεί. Στην σχήμα 1.9 βλέπουμε 3 από αυτές τις περιπτώσεις: (α) μία παλαιά και μία μελλοντική αναφορά (β) δύο παλαιές αναφορές (γ) δύο μελλοντικές αναφορές.



Σχήμα 1.9: Παραδείγματα πρόβλεψης σε ένα B macroblock.

1.6.2.2. Πρόβλεψη με βάρη

Η πρόβλεψη με βάρη είναι μία μέθοδος τροποποίησης δειγμάτων σε motion-compensated δεδομένα που αφορούν είτε P είτε B macroblock (όχι I δηλαδή). Υπάρχουν 3 είδη πρόβλεψης με βάρη στο H.264:

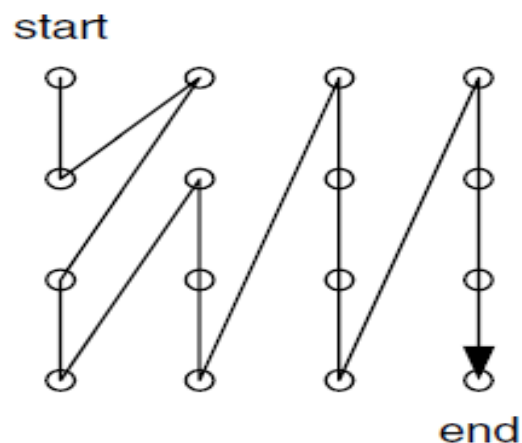
- 1) Πρόβλεψη με σταθερά βάρη σε ένα P macroblock.
- 2) Πρόβλεψη με σταθερά βάρη σε ένα B macroblock.
- 3) Πρόβλεψη με μεταβαλλόμενα βάρη σε ένα B macroblock.

Στις δύο περιπτώσεις που έχουμε σταθερά βάρη, τα βάρη μεταδίδονται με την κεφαλίδα και παραμένουν έτσι. Στην περίπτωση με μεταβαλλόμενα βάρη, θέτουμε μεγαλύτερο βάρος αν η εικόνα αναφοράς είναι κοντά στην τρέχουσα εικόνα και μικρότερο βάρος αν είναι μακριά.

Η περίπτωση της πρόβλεψης με βάρη έχει ιδιαίτερη αξία όταν έχουμε ουσιαστικά εικόνα να «χάνεται» μέσα σε μία άλλη εικόνα (εικόνα δηλαδή στο βάθος μίας άλλης).

1.6.2.3. Interlaced βίντεο

Η αποτελεσματική κωδικοποίηση για interlaced βίντεο απαιτεί εργαλεία τα οποία έχουν βελτιστοποιηθεί για συμπίεση πεδίων macroblock. Αν υποστηρίζεται κωδικοποίηση πεδίου αυτό φαίνεται από την κεφαλίδα κάθε slice. Η επιλογή για κωδικοποίηση πεδίου είτε κωδικοποίηση πλάνου (field-coding και frame-coding αντίστοιχα) μπορεί να γίνεται και στο επίπεδο του macroblock και όχι στην αρχή κεφαλίδας κάθε slice. Σε αυτή την περίπτωση για την κωδικοποίηση του interlaced βίντεο ο κωδικοποιητής θα μπορούσε να κάνει δύο πράγματα (α) να κωδικοποιεί ανά δυάδες frame macroblocks (β) να κωδικοποιεί ανά δυάδες field macroblocks. Η πρόβλεψη σε ένα interlaced βίντεο μπορεί να γίνει με reordering σάρωση είτε με zig-zag σάρωση



Σχήμα 1.10: Reordering σάρωση.

1.6.2.4. CABAC

CABAC είναι τα αρχικά για το context-based adaptive binary arithmetic coding. Με το CABAC μπορούμε να πετύχουμε καλή ποιότητα συμπίεσης με (α) επιλογή πιθανοτικών μοντέλων για κάθε στοιχείο (syntax elements) βάση του περιεχομένου του στοιχείου (β) προσαρμοστική πιθανότητα βάση των στατιστικών που υπάρχουν (γ) χρήση αριθμητικής κωδικοποίησης και όχι μεταβλητού μήκους.

Η κωδικοποίηση δεδομένων στο CABAC περιλαμβάνει τα ακόλουθα στάδια:

- 1) Το CABAC χρησιμοποιεί δυαδική αριθμητική κωδικοποίηση πράγμα που σημαίνει ότι μόνο δυαδικά δεδομένα κωδικοποιούνται (0 ή 1).

Τα στάδια 2,3 και 4 επαναλαμβάνονται για κάθε ψηφίο.

- 2) Κατασκευή πιθανοτικού μοντέλου, βάση του οποίου κάθε ψηφίο θα καθορίζεται σαν 0 ή 1.
- 3) Αριθμητική κωδικοποίηση, όπου κάθε ψηφίο βάση του πιθανοτικού μοντέλου που έχει κατασκευαστεί κωδικοποιείται.
- 4) Κάθε φορά που έχουμε κωδικοποίηση ενός ψηφίου θα έχουμε και ανανέωση του πιθανοτικού μοντέλου, βάση των νέων στοιχείων που υπάρχουν.

1.6.3. Extended Profile

Το Extended Profile (γνωστό και X profile) είναι ιδιαίτερα χρήσιμο για εφαρμογές όπως η αναπαραγωγή βίντεο στο διαδίκτυο (video streaming). Είναι υπερσύνολο του Baseline Profile, όπως βλέπουμε και στο σχήμα 1.12 (σε αντίθεση με το Main Profile που δεν έχει όλα τα στοιχεία του Baseline Profile). Παρουσιάζει επίσης χαρακτηριστικά που είδαμε στο Main Profile, όπως τις στάθμες πρόβλεψης, το Interlaced βίντεο, τα B-slices. Παρουσιάζει επίσης τα SP, SI slices και slices διαμοιρασμού δεδομένων που βοηθούν στην ελαχιστοποίηση του λάθους.

1.6.3.1. SP και SI slices

Τα SP και SI slices είναι ειδικά τροποποιημένα κομμάτια, τα οποία επιτρέπουν αποτελεσματική εναλλαγή μεταξύ μεταδόσεων βίντεο και αποτελεσματική πρόσβαση στον αποκωδικοποιητή. Μία συνηθισμένη απαίτηση σε μία μετάδοση βίντεο, είναι ο αποκωδικοποιητής του βίντεο να μπορεί να εναλλάσσεται μεταξύ μία ή παραπάνω από μία ακολουθίες για κωδικοποίηση. Για παράδειγμα, το ίδιο βίντεο μπορεί να κωδικοποιείται σε διάφορους ρυθμούς μετάδοσης, αλλά ο αποκωδικοποιητής θα πρέπει να μπορεί να αποκωδικοποιήσει την ακολουθία βίντεο με το καλύτερο δυνατό τρόπο.

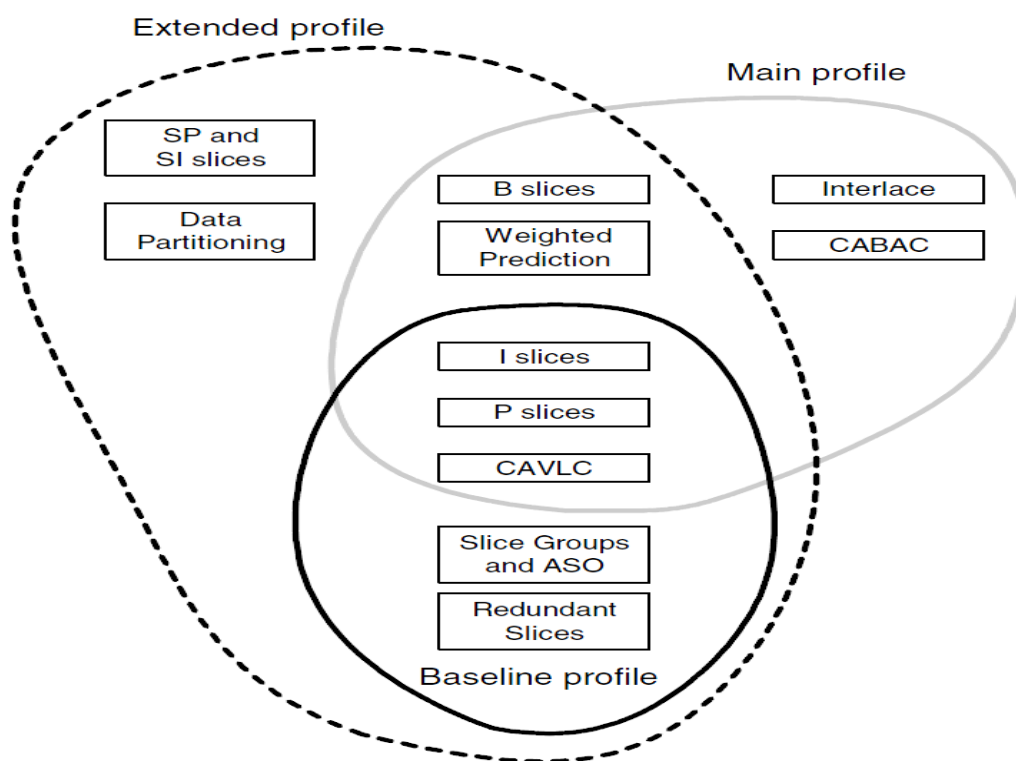
Sequence parameter set	SEI	Picture parameter set	I slice	Picture delimiter	P slice	P slice
---------------------------	-----	--------------------------	---------	----------------------	---------	---------	-------

Σχήμα 1.11: Παράδειγμα μίας ακολουθίας.

1.6.3.2. Data partitioning slices

Τα κωδικοποιημένα δεδομένα, τα οποία συνθέτουν ένα slice τοποθετούνται σε 3 ξεχωριστά τμήματα (A,B,C) , που το καθένα περιέχει ένα υποσύνολο του κωδικοποιημένου slice. Το A τμήμα θα περιέχει το slice της κεφαλίδας καθώς και τα δεδομένα της κεφαλίδας για κάθε macroblock στο slice. Το B τμήμα περιέχει τα υπολειπόμενα κωδικοποιημένα δεδομένα που έχουν μείνει από τα Intra και SI slices. Τα C τμήμα περιέχει τα υπολοιπόμενα κωδικοποιημένα δεδομένα που έχουν μείνει από τα inter κωδικοποιημένα macroblocks. Κάθε τμήμα μπορεί να τοποθετηθεί ξεχωριστά σε μία NAL ακολουθία και να μεταδοθεί ξεχωριστά.

Το A τμήμα είναι το πιο σημαντικό, αφού αν χαθεί θα είναι αδύνατο να αποκωδικοποιηθεί το slice χωρίς λάθος. Τα B και C τμήματα μπορούν κάθε φορά να αποκωδικοποιούνται παράλληλα με το A και έτσι να γίνεται σωστά η αποκωδικοποίηση με ελάχιστο λάθος.



Σχήμα 1.12: Σχέσεις μεταξύ Baseline, Main και Extended Profiles.

1.6.4. Υπόλοιπα Profiles

➤ **Constrained Baseline Profile (CBP)**

Χρησιμοποιείται κυρίως για εφαρμογές χαμηλού κόστους που αφορούν είτε τηλεδιασκέψεις είτε εφαρμογές κινητής τηλεφωνίας. Έχει κάποια κοινά χαρακτηριστικά με τα Baseline, Main και High Profile.

➤ **High Profile**

Το High Profile χρησιμοποιείται κυρίως για μεταδόσεις και αποθηκεύσεις σε δίσκους, ιδιαίτερα για υψηλής ευκρίνειας μεταδόσεις που απαιτούν υψηλό ποσοστό αποθήκευσης (για παράδειγμα αυτό είναι το πρότυπο το οποίο χρησιμοποιούν οι Blu –Ray δίσκοι και DVB HDTV).

➤ **High 10 Profile (Hi10P)**

Αυτό το Profile έχει άμεση σχέση με το High Profile. Προσθέτει στο High Profile 10 bit ανά δείγμα για μεγαλύτερη ακρίβεια.

➤ **High 4:2:2 Profile (Hi422P)**

Αυτό το πρότυπο χρησιμοποιείται κυρίως για εφαρμογές που χρησιμοποιούν interlaced βίντεο. Βρίσκεται σε άμεση σχέση με το High 10 Profile και υποστηρίζει 4:2:2 υποδειγματοληψία χρώματος και 10 bit ανά δείγμα για μεγαλύτερη ακρίβεια.

➤ **High 4:4:4 Predictive Profile (Hi444PP)**

Αυτό το Profile βρίσκεται σε άμεση σχέση με το High 4:2:2 Profile.

Υποστηρίζει 4:4:4 υποδειγματοληψία χρώματος, χρησιμοποιώντας μέχρι και 14 bit ανα δείγμα για μεγαλύτερη ακρίβεια. Υποστηρίζει επίσης αποτελεσματικά lossless region κωδικοποίηση και κωδικοποίηση κάθε εικόνας σαν 3 διαφορετικές περιοχές.

Για βιντεοκάμερες, επεξεργασία και επαγγελματικές εφαρμογές υπάρχουν 4 ακόμα Intra profiles, τα οποία ουσιαστικά λειτουργούν σαν υποσύνολα των προηγούμενων profiles.

- **High 10 Intra Profile**
- **High 4:2:2 Intra Profile**
- **High 4:4:4 Intra Profile**
- **CAVLC 4:4:4 Intra Profile**

Ως αποτέλεσμα της Scalable βίντεο κωδικοποίησης, ορίζονται 3 πρόσθετα profiles, τα οποία ορίζονται ως συνδυασμός ενός profile για το H.264 και εργαλείων με τα οποία επιτυγχάνεται η επέκταση που επιθυμούμε.

➤ **Scalable Baseline Profile**

Βρίσκεται σε άμεση σχέση με το Baseline Profile. Χρησιμοποιείται κυρίως για εφαρμογές τηλεδιάσκεψης, κινητής τηλεφωνίας και παρακολούθησης.

➤ **Scalable High Profile**

Βρίσκεται σε άμεση σχέση με το High Profile και στοχεύει κυρίως εφαρμογές που αφορούν μετάδοση βίντεο.

➤ **Scalable High Intra Profile**

Χρησιμοποιείται κυρίως για παραγωγή βίντεο, βρίσκεται σε άμεση σχέση με το High Intra Profile.

Για multiview βίντεο κωδικοποίηση χρησιμοποιούνται 2 επιπλέον profiles τα οποία είναι τα ακόλουθα:

➤ **Stereo High Profile**

Το profile αυτό στοχεύει στα διπλής θέασης (με 2 κάμερες από διαφορετική οπτική γωνία) 3D βίντεο και συνδυάζει τα εργαλεία του High profile με τις inter-view πρόβλεψη ικανότητες της multiview επέκτασης.

➤ **Multiview High Profile**

Αυτό το profile υποστηρίζει 2 ή περισσότερης θέασης βίντεο, που χρησιμοποιεί inter-picture και inter-view prediction για τα multiview video (θα εξετάσουμε αυτές τις δυνατότητες παρακάτω).

1.7. Εργαλεία του H.264/MPEG-4

Το H.264/MPEG-4 περιλαμβάνει ορισμένες βελτιώσεις σε σχέση με τα προηγούμενα πρότυπα κωδικοποίησης κάνοντας χρήση παρόμοιας βασικής δομής. Αυτές οι βελτιώσεις παρότι αλλάζουν σε ορισμένα σημεία από το ένα profile στο άλλο είναι κατά κύριο βαθμό οι ακόλουθες:

- Ικανοποιητικό χειρισμό interlaced βίντεο, είτε κωδικοποιώντας κάθε interlaced πεδίο σαν διακριτή εικόνα, είτε κωδικοποιώντας περιοχές 16X16 ή 32X32 χρησιμοποιώντας κωδικοποίηση πεδίου (field-based) ή πλάνου (frame-based).
- Χωρική πρόβλεψη intrapicture, με χρήση μεταβλητού μεγέθους block.
- Καλύτερες τεχνικές motion compensation, κυρίως με χρήση μεταβλητής εικόνας αναφοράς και μεταβλητών βαρών πρόβλεψης.
- Μερικοί μετασχηματισμοί, βασισμένοι κυρίως στον DCT βοηθούν στην κωδικοποίηση.
- Λογαριθμική περίοδος κβάντισης.
- Απεικόνιση χωρίς απώλειες, αν χρειαστεί, για ορισμένες περιοχές μίας εικόνας είναι εφικτή.
- Προηγμένη κωδικοποίηση εντροπίας με την χρήση είτε κωδικοποίηση προσαρμοσμένου πεδίου είτε προσαρμοσμένου μήκους.
- Ένας προσαρμοσμένος κωδικοποιητής που μπορεί να προσφέρει καλύτερη interpicture πρόβλεψη.
- Τα αποτελέσματα είναι αρκετά πιο «ανθεκτικά» αν υπάρξει απώλεια για τον οποιοδήποτε λόγο. Μικρότερη πιθανότητα για οποιοδήποτε λάθος.

H.264/AVC Hardware and Software Products

Adopting Bodies and Consortia

- 3GPP and 3GPP2 mobile environments
- ARIB (Japan), DMB (Korea), DAB (Europe), and DVB (Europe) broadcast standards
- AVC Alliance, IMTC, and MPEGIF promotional organizations
- Blu-Ray Disc Association and DVD Forum high-definition storage formats
- DLNA and ISMA multimedia streaming systems
- IETF audio-video transport real-time protocol payload packetization
- ITU-R broadcast and professional usage standards and ITU-T real-time multimedia conferencing systems
- MPEG storage file format and multiplex system standards
- NATO military specifications

Hardware Products

- Mobile phones by Nokia, Samsung, LG, etc.
- Broadcast HD encoders by Harmonic, Modulus, Scientific Atlanta/Cisco, Tandberg TV, Tut Systems, Thomson, etc.
- Direct-broadcast satellite deployments by BSkyB, DirecTV, Dish Network, Euro1080, Premiere, and ProSiebenSat.1.
- All HD DVD and Blu-Ray Disc players
- The Sony Playstation Portable and Apple iPod portable media player devices
- Videoconferencing systems by Polycom, Tandberg Telecom, etc.
- HD-capable decoding chips by Broadcom, Conexant, Micronas, Sigma Designs, and ST Microsystems

Software Products

- Media coder/player software packages, including Apple Quicktime, Cyberlink, Nero Digital, etc.
- DSP implementations for various processors
- SDK packages by Ateame, ATI, Elecard, FastVDO, Intel, KDDI, MainConcept, Sorenson, etc.
- Hardware accelerators for PC software decoders using graphics processing units (GPUs) by ATI, nVidia, etc.

Σχήμα 1.13: Hardware και Software που στηρίζεται στο H.264/MPEG-4.

ΚΕΦΑΛΑΙΟ 2

ΥΛΟΠΟΙΗΣΗ ΠΑΡΑΔΕΙΓΜΑΤΩΝ ΓΙΑ ΤΟ H.264/MPEG-4 ΜΕ ΧΡΗΣΗ ΜΙΑΣ ΚΑΜΕΡΑΣ

2.1. Εισαγωγή

Σε αυτό το κεφάλαιο θα εξετάσουμε κάποια βίντεο-παραδείγματα, τα οποία κωδικοποιήθηκαν με χρήση του H.264/MPEG-4 προτύπου. Συγκεκριμένα, χρησιμοποιήθηκε η διαδικασία *lencod.exe*, όπου με κατάλληλες μετατροπές στο *encoder.cfg* μπορούμε να κωδικοποιήσουμε το βίντεο που θέλουμε.

Αντίστοιχα για την αποκωδικοποίηση χρησιμοποιείται η διαδικασία *ldecod.exe*, με κατάλληλες μετατροπές των μεταβλητών στο *decoder.cfg*.

Πρώτα θα παρουσιάσουμε κάποια χρήσιμα στοιχεία, που πρέπει να λαμβάνουμε υπόψη πριν χρησιμοποιήσουμε αυτές τις ρουτίνες και έπειτα τα αποτελέσματα που πήραμε από την κωδικοποίηση και αποκωδικοποίηση κάποιων συγκεκριμένων βίντεο με χρήση μίας μόνο κάμερας. Στο επόμενο κεφάλαιο θα εξετάσουμε και τα αποτελέσματα που πήραμε από την κωδικοποίηση και αποκωδικοποίηση βίντεο, που λήφθηκαν από παραπάνω από μία κάμερα (multiview).

2.2. Χρήσιμα στοιχεία για την κωδικοποίηση και αποκωδικοποίηση

Προτού δούμε τα αποτελέσματα που πήραμε από την κωδικοποίηση και αποκωδικοποίηση βίντεο με χρήση μίας κάμερας, θα πρέπει να δούμε κάποια στοιχεία όσον αφορά το *encoder.cfg* και *decoder.cfg* που θα προσαρμόζουμε αναλόγως με το βίντεο που θέλουμε να κωδικοποιηθεί και αποκωδικοποιηθεί αντίστοιχα.

Βέβαια, εκτός από αυτά τα κύρια στοιχεία που θα παρουσιαστούν είναι αρκετές ακόμα μεταβλητές που θα πρέπει να εξετάσουμε ανάλογα με την περίπτωση του βίντεο που έχουμε.

2.2.1. Όνομα βίντεο κατά την κωδικοποίηση και παραγόμενα αποτελέσματα

Το όνομα του βίντεο που θέλουμε να κωδικοποιήσουμε το τοποθετούμε στην μεταβλητή: *InputFile*. Αυτό το κάνουμε μόνο στην περίπτωση που έχουμε βίντεο με μία κάμερα. Αν έχουμε multiview βίντεο θα πρέπει να τοποθετηθεί σε άλλη μεταβλητή. Το αρχείο, που θα πρέπει να είναι ίδιο με το αρχικό που παράγεται κατά την κωδικοποίηση το βάζουμε στην μεταβλητή: *ReconFile*. Η H.264 ακολουθία που θα πάρουμε βρίσκεται στην μεταβλητή: *OutputFile*.

Επίσης σημαντικό είναι να τοποθετήσουμε τις διαστάσεις του βίντεο. Αυτά τοποθετούνται στις μεταβλητές *SourceWidth* και *SourceHeight* αντιστοίχως.

2.2.2. Frames κατά την κωδικοποίηση

Τα πλάνα (frames) που έχει κάθε βίντεο για την κωδικοποίηση τα τοποθετούμε στην μεταβλητή: *FramesToBeEncoded*. Ο ρυθμός με τον οποίο έρχεται κάθε πλάνο στο βίντεο μας τοποθετείται στην μεταβλητή: *FrameRate*. Επίσης μπορούμε να διαλέξουμε από που θα ξεκινήσει η κωδικοποίηση μας μέσω της μεταβλητής: *StartFrame*

Αν έχουμε multiview βίντεο θα πρέπει σε αυτή την μεταβλητή να τοποθετήσουμε πόσα frames έχει κάθε βίντεο, και όχι τα συνολικά frames. Φυσικά θα πρέπει όλα τα frames να είναι ίσα σε όλα τα βίντεο.

2.2.3. Profile του βίντεο

Είδαμε στο προηγούμενο κεφάλαιο κάποια σημαντικά profiles τα οποία χρησιμοποιούνται στο H.264/MPEG-4 πρότυπο. Συγκεκριμένα τοποθετούμε στην μεταβλητή *ProfileIDC* το profile που θέλουμε. Ας δούμε μερικές περιπτώσεις:

- Με 66 αν θέλουμε το baseline profile.
- Με 77 αν θέλουμε το main profile.
- Με 88 αν θέλουμε το extended profile.
- Με 100 αν θέλουμε το high profile.
- Με 110 αν θέλουμε το high 10 profile.
- Με 122 αν θέλουμε το high 4:2:2 profile.
- Με 244 αν θέλουμε το high 4:4:4 profile.
- Με 44 αν θέλουμε το 4:4:4 intra profile.

2.2.4. Άλλες μεταβλητές κατά την κωδικοποίηση

Όπως αναφέρθηκε πριν, οι μεταβλητές που είδαμε δεν είναι οι μόνες που θα πρέπει να λαμβάνει υπόψη ο χρήστης αλλά ίσως οι πιο σημαντικές. Άλλες χρήσιμες μεταβλητές είναι οι:

- *Bitrate* που μας δίνει τον ρυθμό bps.
- *NumberBframes* όπου βάζουμε τον αριθμό των B frames αν υπάρχουν.
- *YuvFormat* όπου τοποθετούμε ποιο είδος υποδειγματοληψίας θέλουμε να χρησιμοποιήσουμε. Με 0 χρησιμοποιούμε το 4:0:0, με 1 το 4:2:0, με 2 το 4:2:2 και με 3 το 4:4:4.
- *RGBInput* όπου βάζουμε 0 αν έχουμε βίντεο σε μορφή YUV και 1 αν είναι σε μορφή RGB.

Οι άλλες μεταβλητές είναι στην διάθεση του αναγνώστη να τις βρει στο encoder.cfg.

2.2.5. Στοιχεία για την αποκωδικοποίηση

Σημαντικό είναι να εξετάσουμε και κάποιες μεταβλητές που πρέπει να λάβουμε υπόψην μας κατά την αποκωδικοποίηση. Αυτές βρίσκονται στο `decoder.cfg`. Συγκεκριμένα έχουμε:

- *H.26L coded bitstream* όπου θα πρέπει να βάλουμε την H.264 ακολουθία που έχουμε πάρει κατά την κωδικοποίηση.
- *Output file structure* όπου θα πρέπει να βάλουμε το αρχείο YUV ή RGB που έχουμε πάρει κατά την κωδικοποίηση.
- *Write 4:2:0 chroma components for monochrome streams* όπου βάζουμε ποιο είδος υποδειγματοληψίας έχουμε χρησιμοποιήσει. Οι αντίστοιχοι αριθμοί είναι ίδιοι και με την κωδικοποίηση.

Οι άλλες μεταβλητές είναι στην διάθεση του αναγνώστη να τις βρει στο `decoder.cfg`.

2.3. Αποτελέσματα κωδικοποιήσεων με χρήση μίας κάμερας

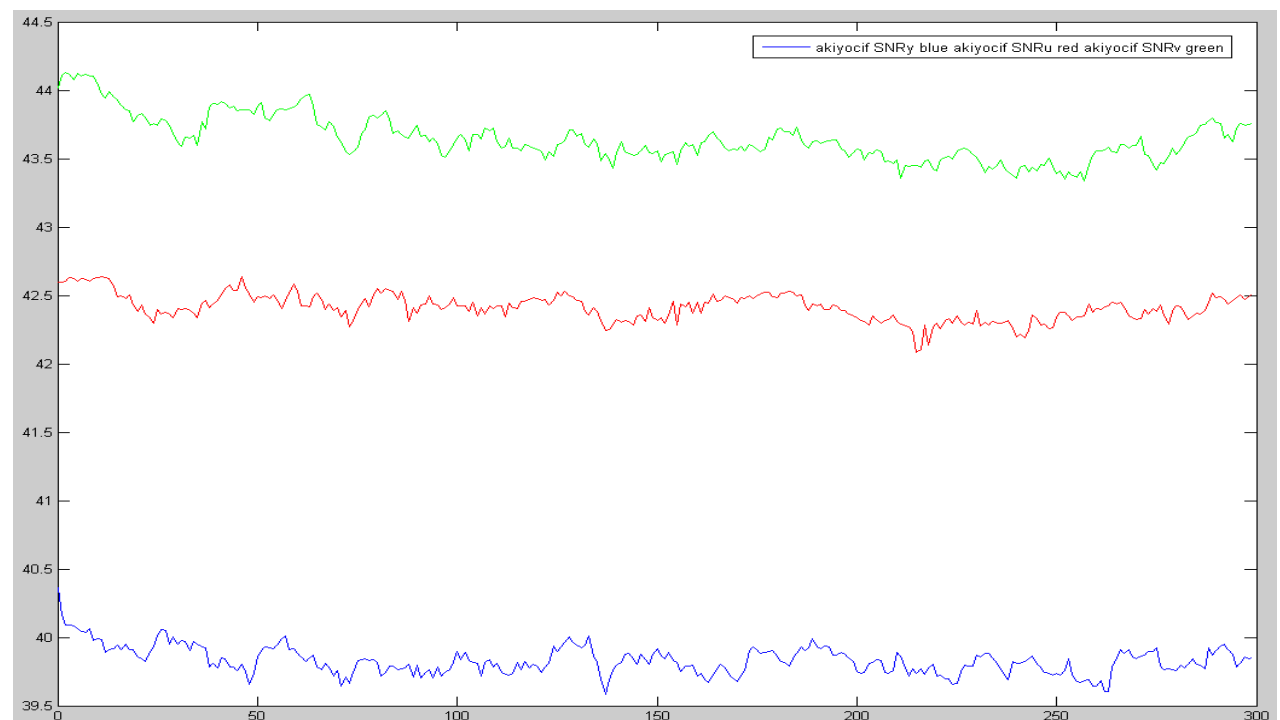
Σε αυτή την ενότητα θα μελετήσουμε τα αποτελέσματα που έχουμε αν πάρουμε τις προσομοιώσεις που έγιναν με την χρήση ενός βίντεο και μίας κάμερας. Συγκεκριμένα θα παρουσιαστούν τα αποτελέσματα από το τρέξιμο 4 βίντεο των: `akiyo_cif`, `coastguard_cif`, `container_cif` και `carphone_qcif`. Έπειτα θα παρουσιαστούν αποτελέσματα για το `Door_Flowers` βίντεο που θα χρησιμοποιηθεί στο επόμενο κεφάλαιο για το Multiview, ώστε αν θέλει ο αναγνώστης να μπορεί να κάνει μία αντίστοιχη σύγκριση. Σε κάθε βίντεο θα παρουσιάζονται οι γραφικές παραστάσεις για τις συνιστώσες Y, U και V καθώς και κάποια σημαντικά στοιχεία που πρέπει να λαμβάνονται υπόψη κατά την κωδικοποίηση. Αντίστοιχα παρουσιάζεται και αν γίνεται σωστά η αποκωδικοποίηση.

2.3.1. Akiyo_cif

Το αρχείο akiyo_cif έχει τα ακόλουθα χαρακτηριστικά:

- 300 frames άρα τοποθετείται 300 στο *FrameToBeEncoded*
- 352X288 διαστάσεις άρα τοποθετήθηκαν 352 και 288 στο *SourceWidth* και *SourceHeight* αντίστοιχα.
- Χρησιμοποιείται main profile για αυτό τοποθετούμε 77 στην μεταβλητή *ProfileIDC*
- Το βίντεο που παίρνουμε σαν αποτέλεσμα είναι το outakiyo_cif και τοποθετούμε αυτή την ονομασία στο *ReconFile* και έχουμε test.264 όπου και κρατάμε το αρχείο H.264 που θα χρησιμοποιηθεί και κατά την αποκωδικοποίηση.
- Χρησιμοποιείται 4:2:0 μέθοδος υποδειγματοληψίας χρώματος. Οπότε τοποθετείται 1 στο *YUVFormat*.
- Έχουμε YUV βίντεο οπότε τοποθετούμε 0 στο *RGBInput*.

Από τις Y, U και V συνιστώσες πήραμε τα ακόλουθα αποτελέσματα:



Σχήμα 2.1: Παρουσίαση αποτελεσμάτων για το SNR στο βίντεο akiyo_cif των συνιστωσών Y (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα).

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

----- Average data all frames -----

SNR Y(dB)	: 39.83
SNR U(dB)	: 42.41
SNR V(dB)	: 43.64
cSNR Y(dB)	: 39.82 (6.77)
cSNR U(dB)	: 42.41 (3.73)
cSNR V(dB)	: 43.64 (2.81)
Total bits	: 770264 (I 43448, P 726632, NVB 184)
Bit rate (kbit/s) @ 30.00 Hz	: 77.03
Bits to avoid Startcode Emulation	: 0
Bits for parameter sets	: 184

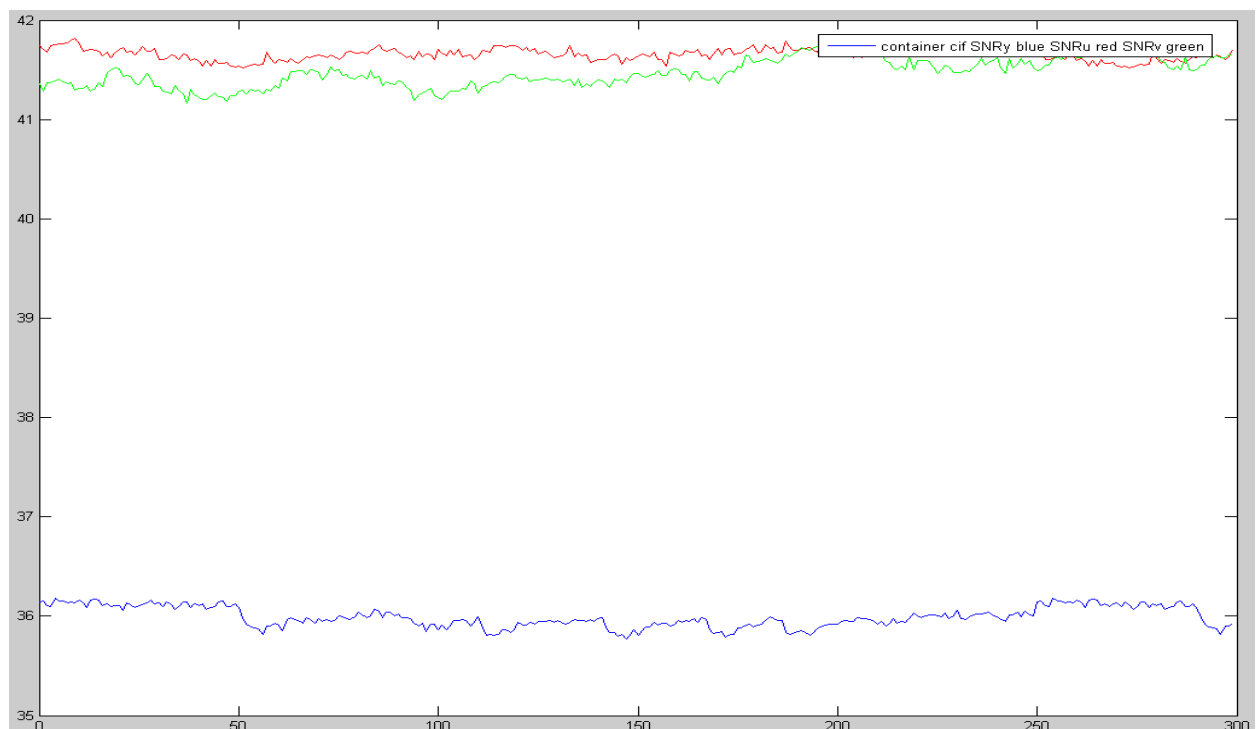
Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλαμε το βίντεο outakiyo_cif που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

2.3.2. Container_cif

Το αρχείο coantainer_cif έχει τα ακόλουθα χαρακτηριστικά:

- 300 frames άρα τοποθετείται 300 στο *FrameToBeEncoded*
- 352X288 διαστάσεις άρα τοποθετήθηκαν 352 και 288 στο *SourceWidth* και *SourceHeight* αντίστοιχα.
- Χρησιμοποιείται extended profile για αυτό τοποθετούμε 88 στην μεταβλητή *ProfileIDC*
- Το βίντεο που παίρνουμε σαν αποτέλεσμα είναι το oucontainer_cif και τοποθετούμε αυτή την ονομασία στο *ReconFile* και έχουμε test.264 όπου και κρατάμε το αρχείο H.264 που θα χρησιμοποιηθεί και κατά την αποκωδικοποίηση.
- Χρησιμοποιείται 4:2:0 μέθοδος υποδειγματοληψίας χρώματος. Οπότε τοποθετείται 1 στο *YUVFormat*.
- Έχουμε YUV βίντεο οπότε τοποθετούμε 0 στο *RGBInput*.

Από τις Y, U και V συνιστώσες πήραμε τα ακόλουθα αποτελέσματα:



Σχήμα 2.2: Παρουσίαση αποτελεσμάτων για το SNR στο βίντεο container_cif των συνιστωσών Y (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα).

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

----- *Average data all frames* -----

<i>SNR Y(dB)</i>	<i>: 36.01</i>
<i>SNR U(dB)</i>	<i>: 41.70</i>
<i>SNR V(dB)</i>	<i>: 41.46</i>
<i>cSNR Y(dB)</i>	<i>: 36.01 (16.29)</i>
<i>cSNR U(dB)</i>	<i>: 41.70 (4.39)</i>
<i>cSNR V(dB)</i>	<i>: 41.46 (4.65)</i>
<i>Total bits</i>	<i>: 1882120 (I 81944, P 1799992, NVB 184)</i>
<i>Bit rate (kbit/s) @ 30.00 Hz</i>	<i>: 188.21</i>
<i>Bits to avoid Startcode Emulation</i>	<i>: 0</i>
<i>Bits for parameter sets</i>	<i>: 184</i>

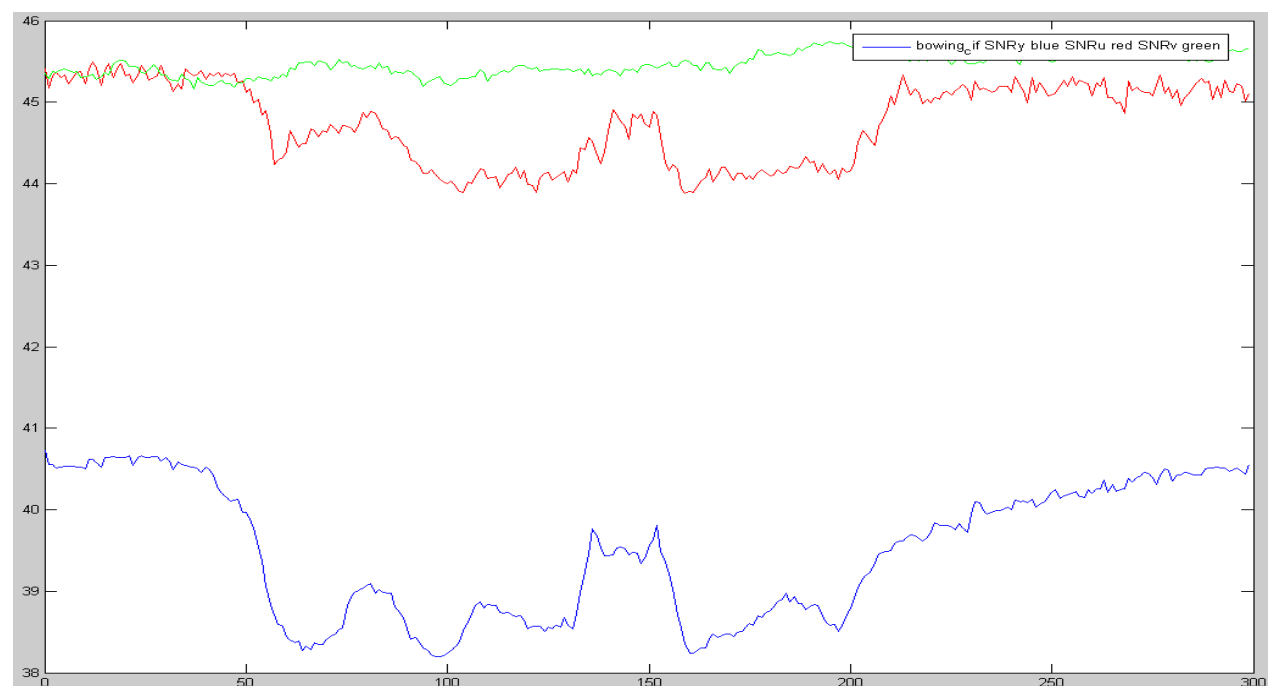
Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλουμε το βίντεο outcontainer_cif που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

2.3.3. Bowling_cif

Το αρχείο bowling_cif έχει τα ακόλουθα χαρακτηριστικά:

- 300 frames άρα τοποθετείται 300 στο *FrameToBeEncoded*
- 352X288 διαστάσεις άρα τοποθετήθηκαν 352 και 288 στο *SourceWidth* και *SourceHeight* αντίστοιχα.
- Χρησιμοποιείται baseling profile για αυτό τοποθετούμε 66 στην μεταβλητή *ProfileIDC*
- Το βίντεο που παίρνουμε σαν αποτέλεσμα είναι το outbowling_cif και τοποθετούμε αυτή την ονομασία στο *ReconFile* και έχουμε test.264 όπου και κρατάμε το αρχείο H.264 που θα χρησιμοποιηθεί και κατά την αποκωδικοποίηση.
- Χρησιμοποιείται 4:2:0 μέθοδος υποδειγματοληψίας χρώματος. Οπότε τοποθετείται 1 στο *YUVFormat*.
- Έχουμε YUV βίντεο οπότε τοποθετούμε 0 στο *RGBInput*.

Από τις Y, U και V συνιστώσες πήραμε τα ακόλουθα αποτελέσματα:



Σχήμα 2.3: Παρουσίαση αποτελεσμάτων για το SNR στο βίντεο bowling_cif των συνιστωσών Y (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα).

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

----- Average data all frames -----

SNR Y(dB)	: 39.49
SNR U(dB)	: 44.75
SNR V(dB)	: 45.16
cSNR Y(dB)	: 39.42 (7.44)
cSNR U(dB)	: 44.72 (2.19)
cSNR V(dB)	: 45.14 (1.99)
Total bits	: 1208624 (I 39312, P 1169136, NVB 176)
Bit rate (kbit/s) @ 30.00 Hz	: 120.86
Bits to avoid Startcode Emulation	: 0
Bits for parameter sets	: 176

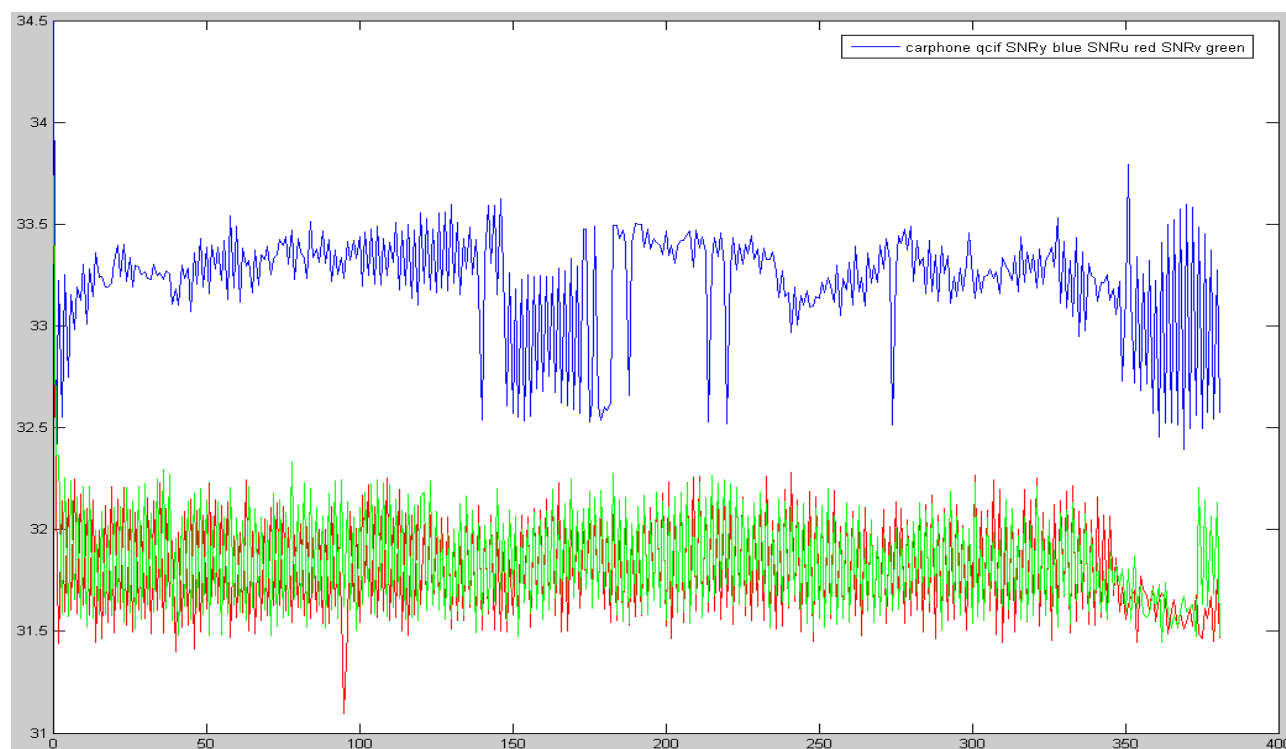
Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλαμε το βίντεο outbowing_cif που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

2.3.4. Carphone_qcif

Το αρχείο carphone_qcif έχει τα ακόλουθα χαρακτηριστικά:

- 382 frames άρα τοποθετείται 382 στο *FrameToBeEncoded*
- 176X144 διαστάσεις άρα τοποθετήθηκαν 176 και 144 στο *SourceWidth* και *SourceHeight* αντίστοιχα.
- Χρησιμοποιείται extended profile για αυτό τοποθετούμε 88 στην μεταβλητή *ProfileIDC*
- Το βίντεο που παίρνουμε σαν αποτέλεσμα είναι το outcarphone_qcif και τοποθετούμε αυτή την ονομασία στο *ReconFile* και έχουμε test.264 όπου και κρατάμε το αρχείο H.264 που θα χρησιμοποιηθεί και κατά την αποκωδικοποίηση.
- Χρησιμοποιείται 4:2:0 μέθοδος υποδειγματοληψίας χρώματος. Οπότε τοποθετείται 1 στο *YUVFormat*.
- Έχουμε YUV βίντεο οπότε τοποθετούμε 0 στο *RGBInput*.

Από τις Y, U και V συνιστώσες πήραμε τα ακόλουθα αποτελέσματα:



Σχήμα 2.4: Παρουσίαση αποτελεσμάτων για το SNR στο βίντεο carphone_qcif των συνιστωσών Y (με μπλε χρώμα), U (με κόκκινο χρώμα) και V (με πράσινο χρώμα).

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

```
----- Average data all frames -----  
  
SNR Y(dB)           : 33.29  
SNR U(dB)           : 31.82  
SNR V(dB)           : 31.85  
cSNR Y(dB)          : 33.28 (30.55)  
cSNR U(dB)          : 31.81 (42.88)  
cSNR V(dB)          : 31.84 (42.59)  
Total bits          : 16963208 (I 82936, P 16880096, NVB 176)  
Bit rate (kbit/s) @ 30.00 Hz : 1332.19  
Bits to avoid Startcode Emulation : 64  
Bits for parameter sets : 176
```

Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλαμε το βίντεο outcarphone_qcif που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

2.3.5. Door_Flowers

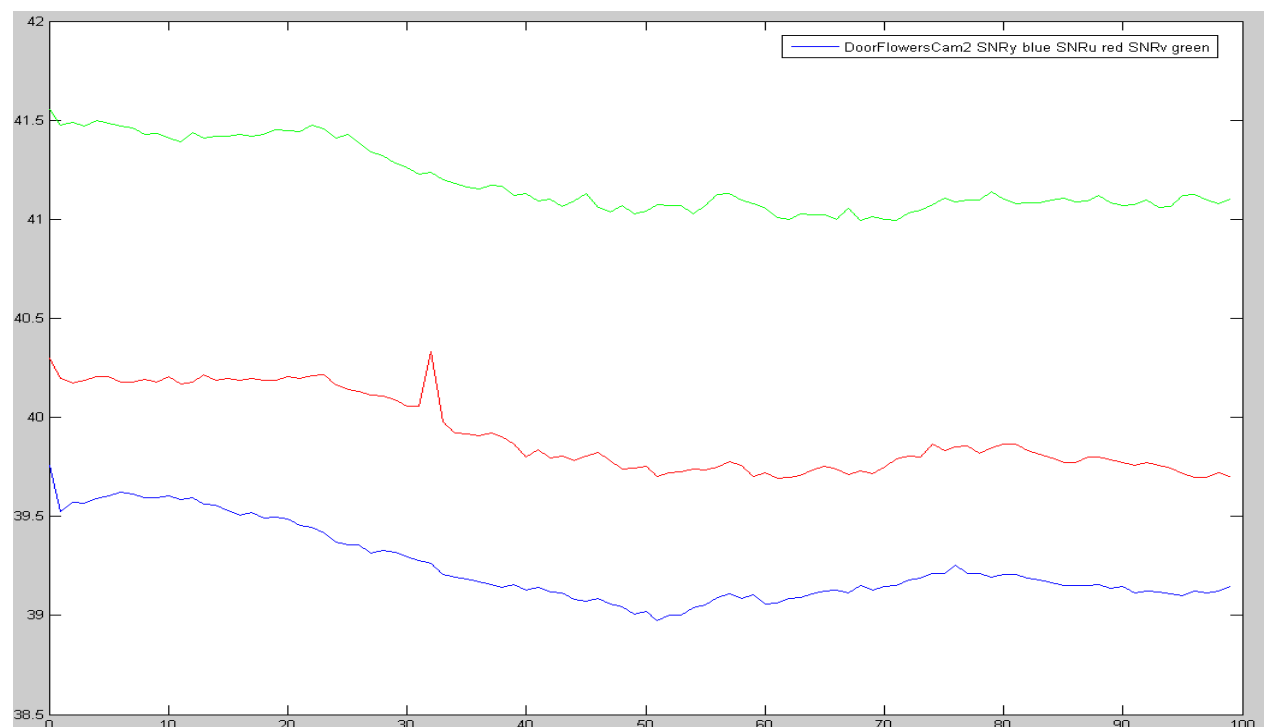
Σε αυτή την ενότητα θα εξετάσουμε τα 16 βίντεο από το Door_Flowers. Οι μεταβλητές που τοποθετούνται για το κάθε ένα είναι ίδιες για αυτό παρουσιάζονται μία φορά:

- 100 frames άρα τοποθετείται 100 στο *FrameToBeEncoded*
- 1024X768 διαστάσεις άρα τοποθετήθηκαν 1024 και 768 στο *SourceWidth* και *SourceHeight* αντίστοιχα.
- Χρησιμοποιείται baseline profile για αυτό τοποθετούμε 66 στην μεταβλητή *ProfileIDC*
- Τα βίντεο που παίρνουμε σαν αποτέλεσμα είναι τα outdoorFlowers_cif με το αντίστοιχο μούμερο από την κάμερα, και τοποθετούμε αυτή την ονομασία στο *ReconFile* και έχουμε test.264 όπου και κρατάμε το αρχείο H.264 που θα χρησιμοποιηθεί και κατά την αποκωδικοποίηση.
- Χρησιμοποιείται 4:2:0 μέθοδος υποδειγματοληψίας χρώματος. Οπότε τοποθετείται 1 στο *YUVFormat*.
- Έχουμε YUV βίντεο οπότε τοποθετούμε 0 στο *RGBInput*.

Τώρα θα παρουσιάσουμε τα διαγράμματα για το κάθε βίντεο για τις Y U και V συνιστώσες:



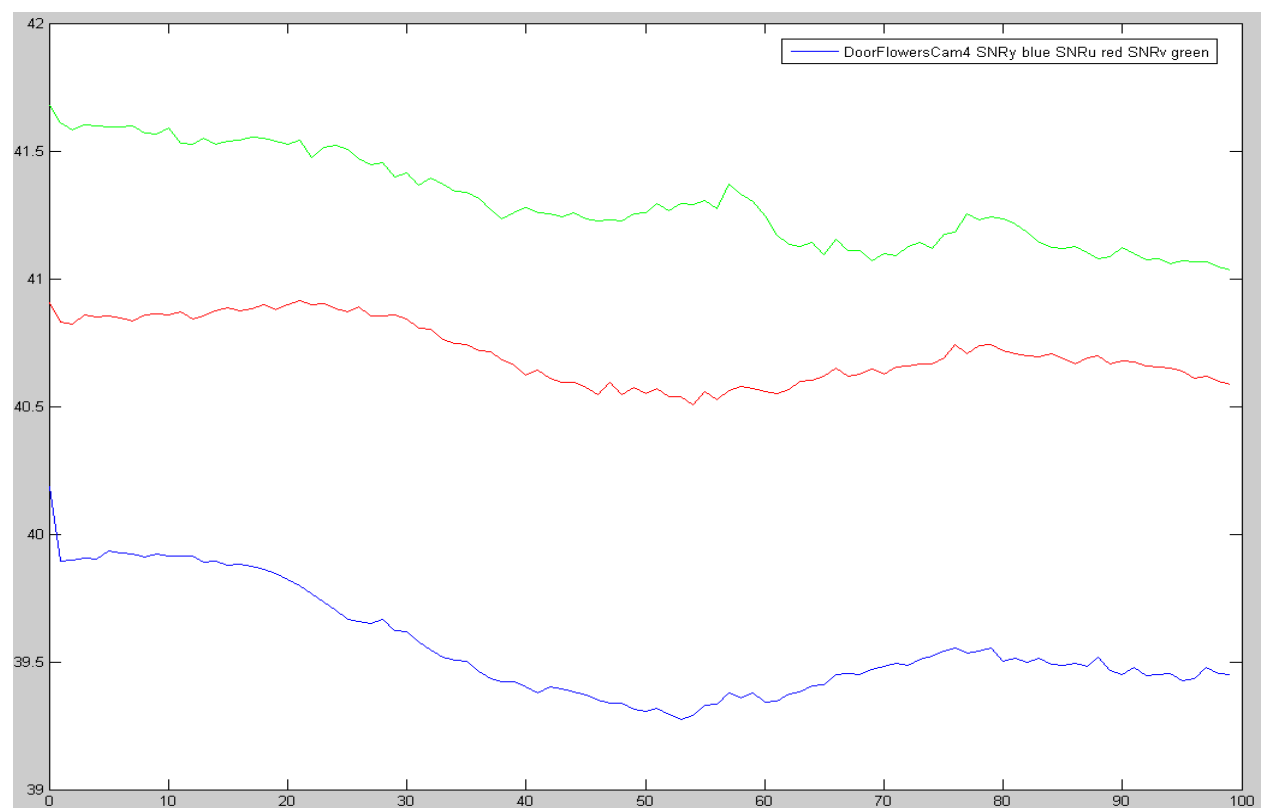
Σχήμα 2.5: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam01.



Σχήμα 2.6: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam02.



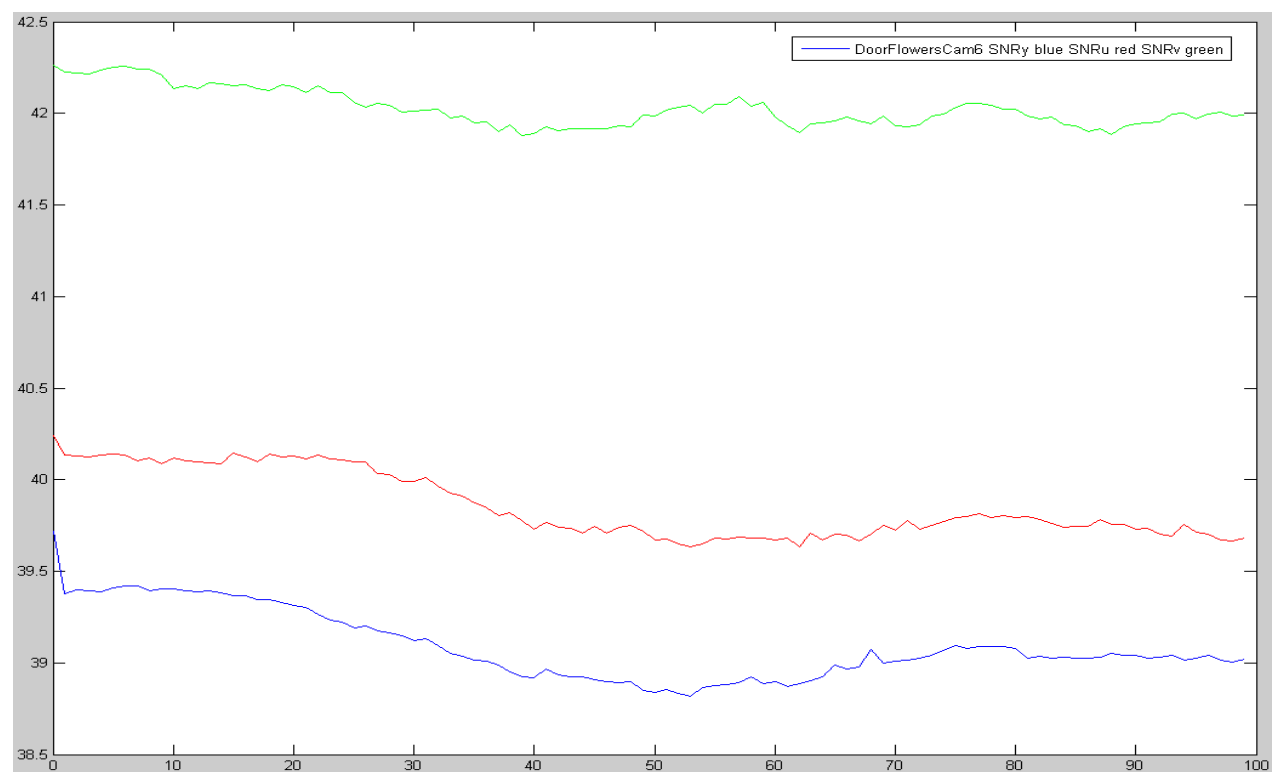
Σχήμα 2.7: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam03.



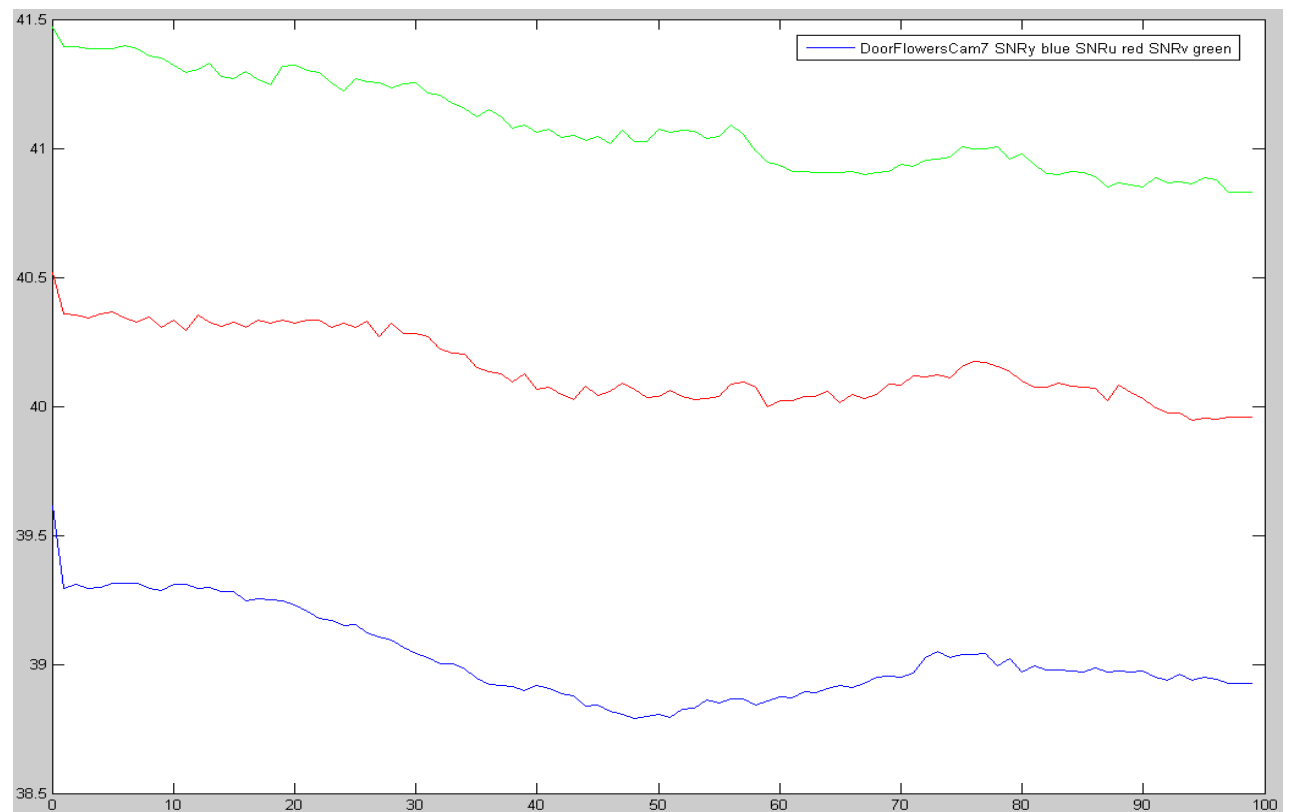
Σχήμα 2.8: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam04.



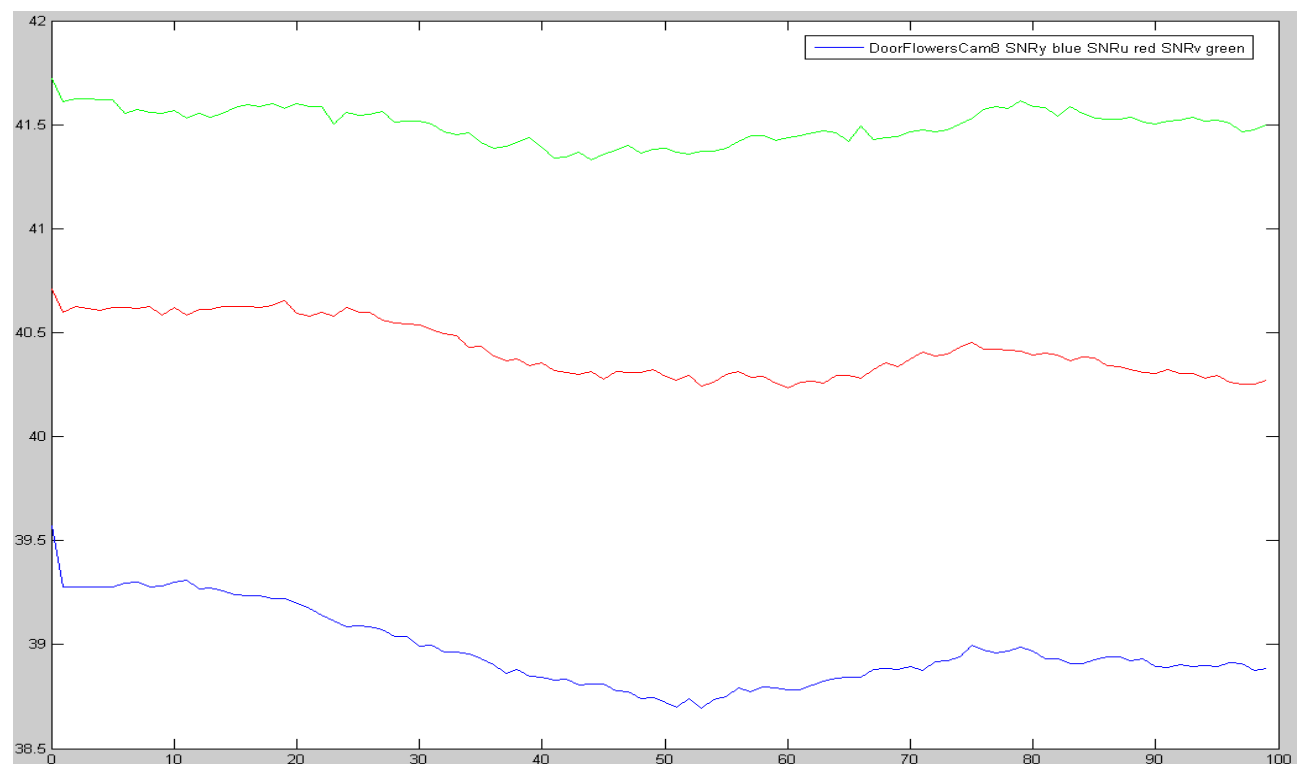
Σχήμα 2.9: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam05.



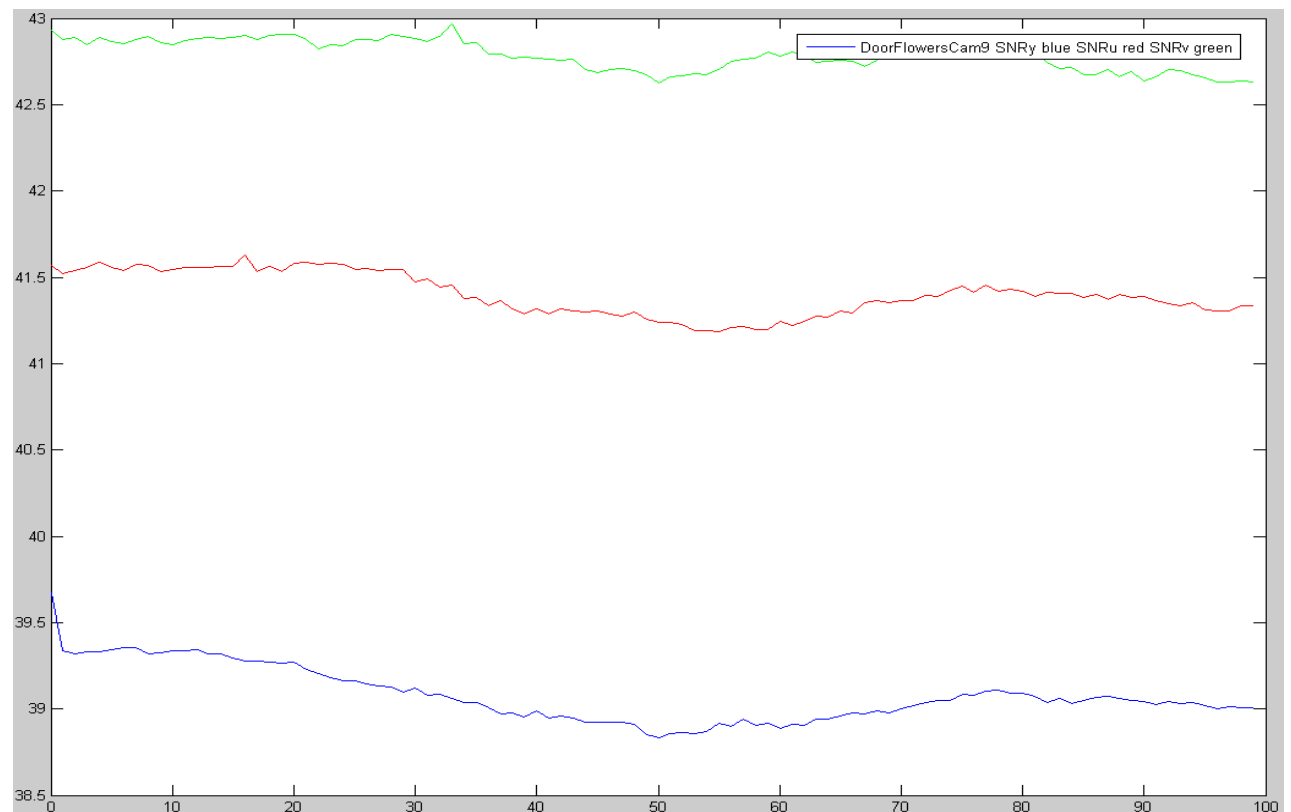
Σχήμα 2.10: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam06.



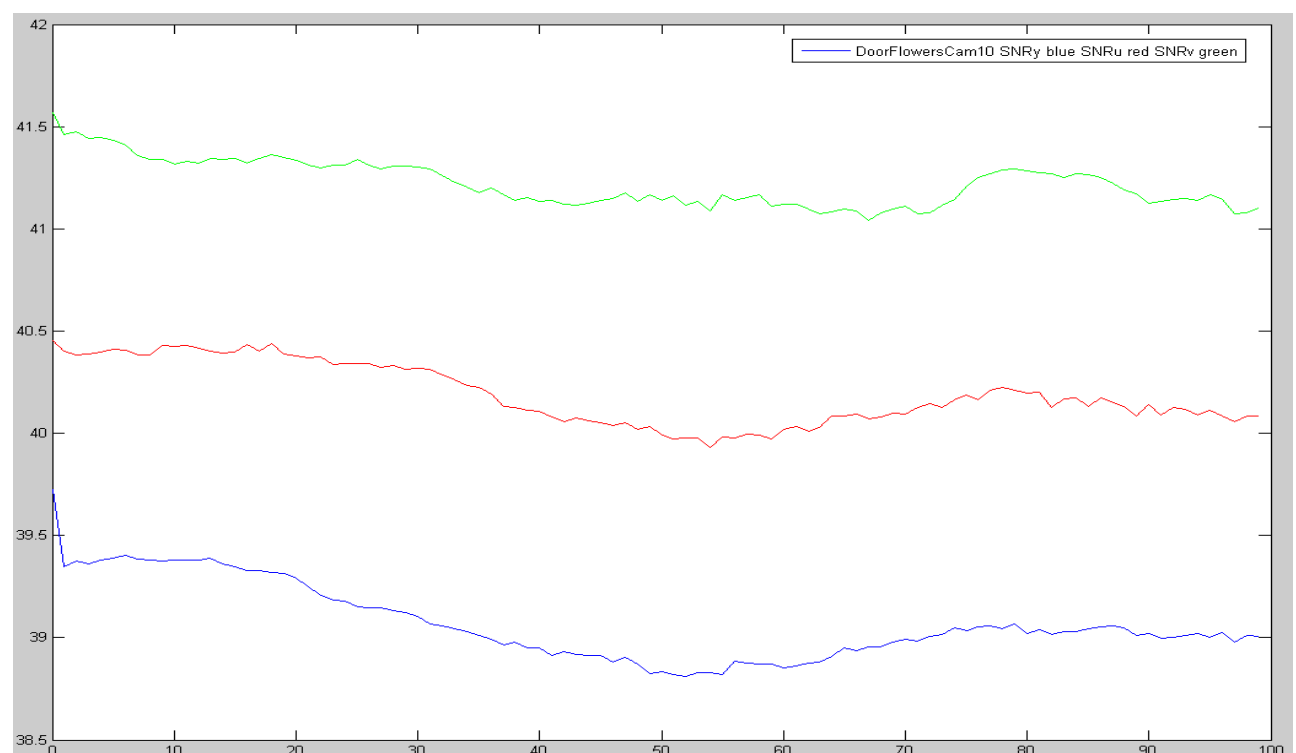
Σχήμα 2.11: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam07.



Σχήμα 2.12: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam08.



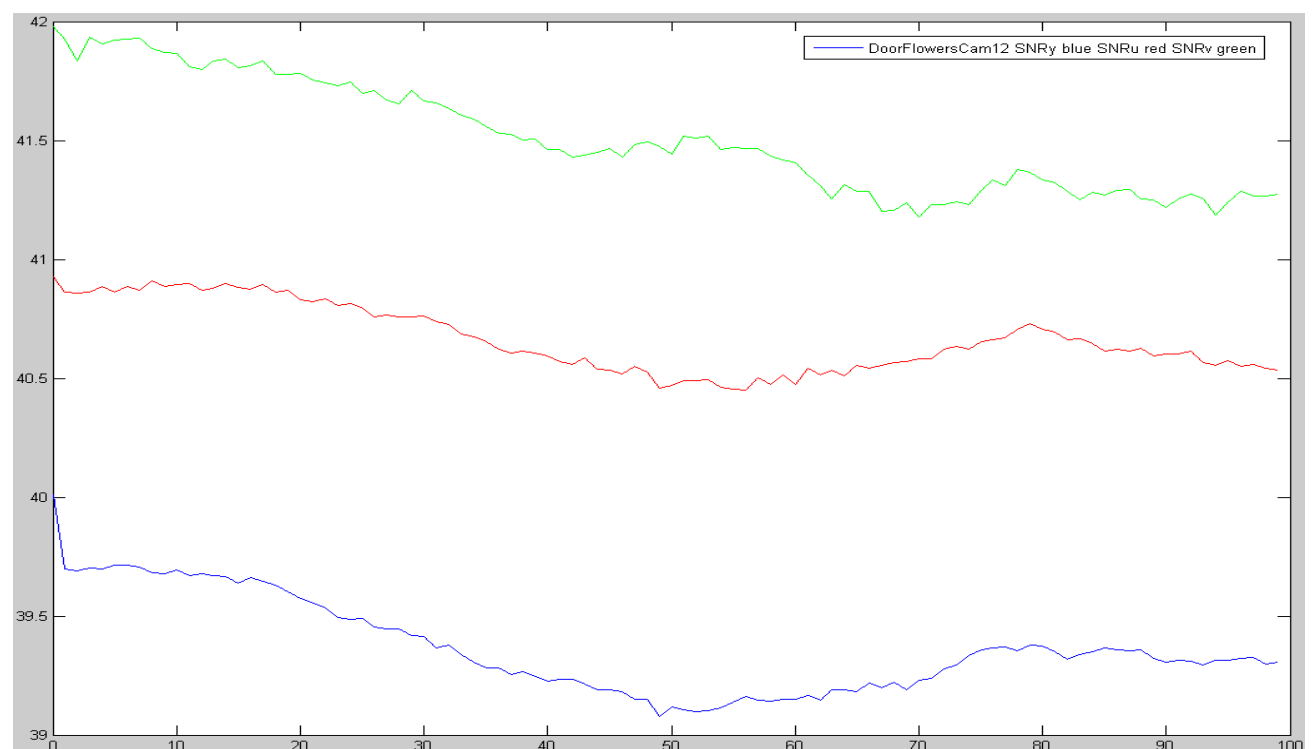
Σχήμα 2.13: Συνιστώσες SNR Υ (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam09.



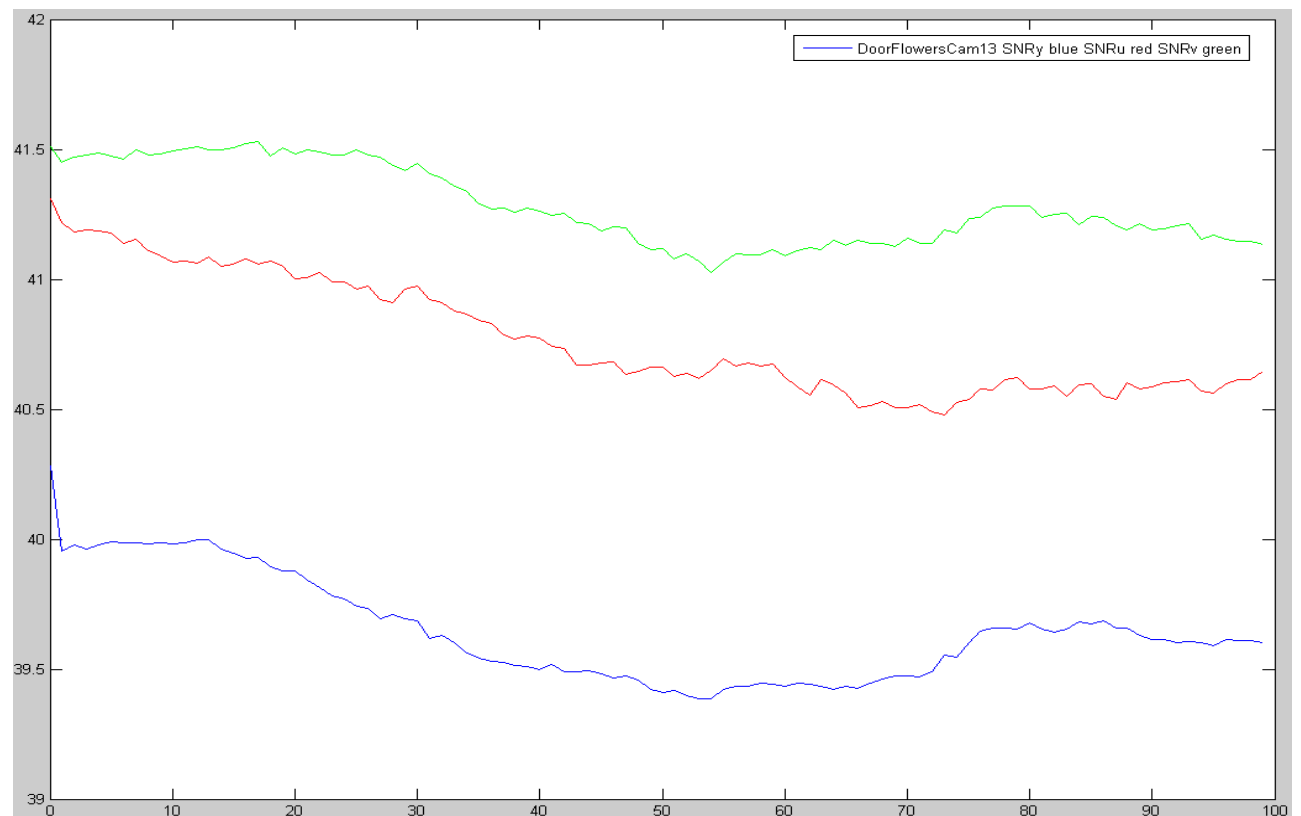
Σχήμα 2.14: Συνιστώσες SNR Υ (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam10.



Σχήμα 2.15: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam11.



Σχήμα 2.16: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam12.



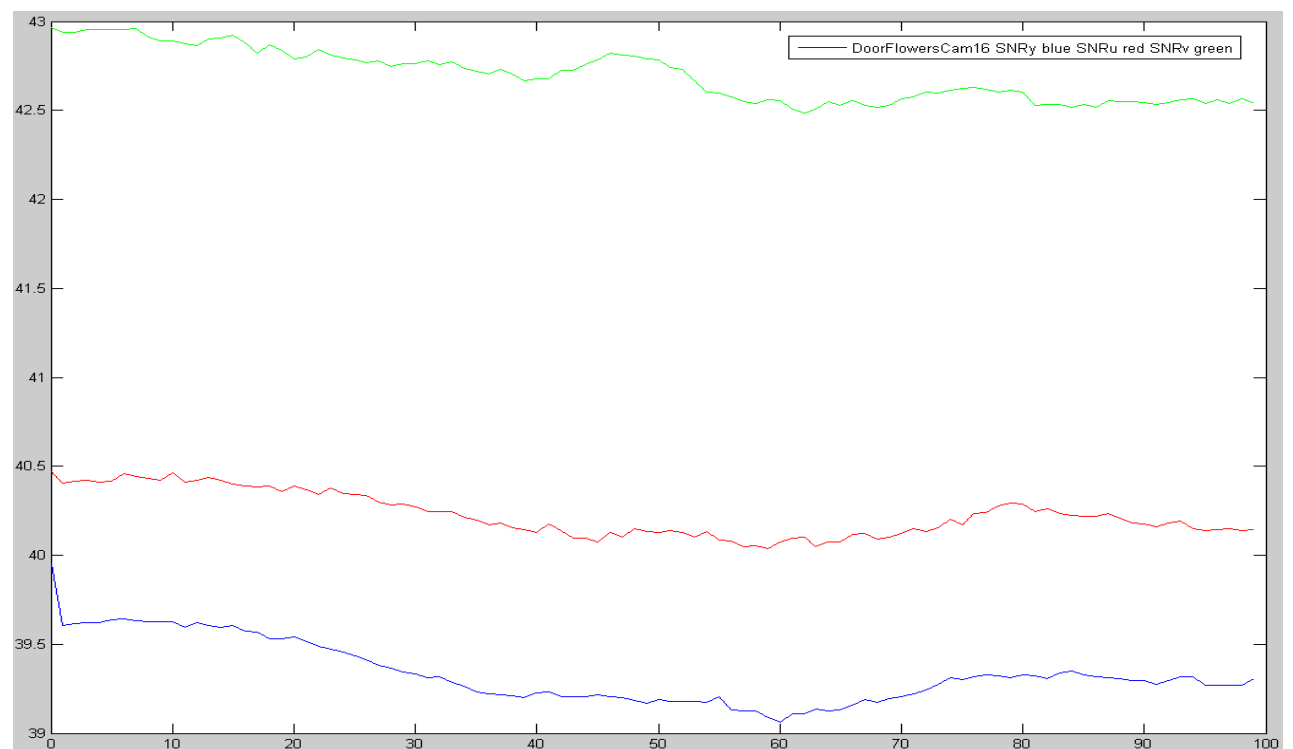
Σχήμα 2.17: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam13.



Σχήμα 2.18: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam14.



Σχήμα 2.19: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam15.



Σχήμα 2.20: Συνιστώσες SNR Y (με μπλε χρώμα) U (με κόκκινο χρώμα) και V (με πράσινο χρώμα) για το Door_FlowersCam16.

Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλαμε το βίντεο outDoorFlowers με το νούμερο της κάμερας που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

ΚΕΦΑΛΑΙΟ 3

ΤΡΕΞΙΜΑΤΑ ΓΙΑ ΠΟΛΛΑΠΛΗΣ ΘΕΑΣΗΣ ΒΙΝΤΕΟ ΜΕ ΔΙΑΦΟΡΑ ΜΟΝΤΕΛΑ ΑΝΑΦΟΡΑΣ

3.1. Εισαγωγή για multiview

Στο προηγούμενο κεφάλαιο είδαμε κάποια στοιχεία για την κωδικοποίηση ενός βίντεο με χρήση μίας κάμερας με το πρότυπο H.264/MPEG-4, μέσω των διαδικασιών *lencod.exe* και *encoder.cfg*. Αντίστοιχα είδαμε και την αποκωδικοποίηση μέσω των ίδιων διαδικασιών. Σε αυτό το κεφάλαιο θα επικεντρωθούμε στην εξέταση αποτελεσμάτων μέσω των ίδιων διαδικασιών για βίντεο που συλλαμβάνονται ταυτόχρονα από πολλαπλές κάμερες.

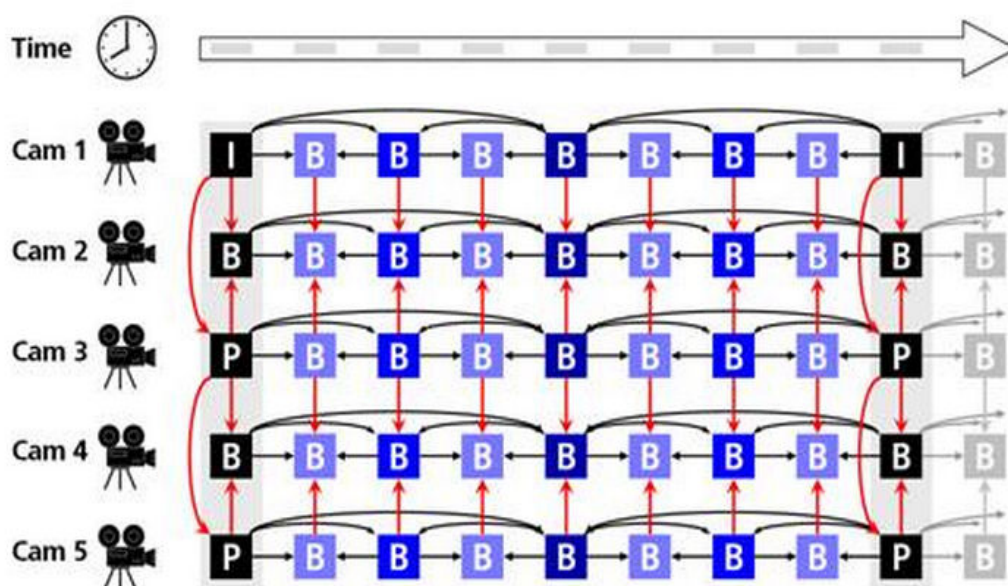
Θα δούμε κάποιους χρήσιμους ορισμούς για τα πολλαπλές θέασης βίντεο (multiview) και γενικότερα κάποια χρήσιμα στοιχεία για την κωδικοποίηση πολλαπλής θέασης βίντεο. Έπειτα μέσω διάφορων μοντέλων αναφοράς που θα εξετάσουμε αρχικά θα δούμε τα αποτελέσματα που πήραμε από την κωδικοποίηση και αποκωδικοποίηση multiview βίντεο.

3.2. Χρήσιμοι ορισμοί

Η κωδικοποίηση πολλαπλής θέασης βίντεο (multiview video coding – MVC) είναι μία τροποποίηση του H.264/MPEG-4 προτύπου. Επιτρέπει την αποτελεσματική κωδικοποίηση ακολουθιών που συλλαμβάνονται ταυτόχρονα από πολλαπλές κάμερες, χρησιμοποιώντας μία ενιαία ροή βίντεο.

Το MVC προορίζεται για στερεοσκοπική κωδικοποίηση βίντεο, όπως επίσης και δωρεάν τηλεοπτικό σήμα, όπως και τρισδιάστατη κωδικοποίηση πολλαπλής θέασης βίντεο. Το Stereo High Profile είναι αυτό που χρησιμοποιείται περισσότερο σαν πρότυπο για τα πολλαπλής θέασης βίντεο.

Τα multiview βίντεο περιέχουν ένα μεγάλο ποσοστό στατιστικών εξαρτήσεων μεταξύ τους, δεδομένου ότι λαμβάνεται η ίδια σκηνή από πολλαπλές μεριές. Οπότε η σωστή χωρική και χρονική πρόβλεψη είναι απαραίτητη για την σωστή κωδικοποίηση multiview βίντεο. Ένα πλάνο (frame) από μία κάμερα σχετίζεται και θα πρέπει να προβλέπεται σε σύγκριση με πλάνα τόσο από την ίδια όσο και από τις άλλες κάμερες. Αυτό θα πρέπει να το λαμβάνουμε υπόψη και κατά την πρόβλεψη μας στα διάφορα μοντέλα αναφοράς.



Σχήμα 3.1: Παράδειγμα μοντέλου αναφοράς ανά 8 πλάνα (frames).

3.3. Σημαντικά ζητήματα για τα multiview βίντεο

Πριν δούμε τις εφαρμογές και την χρησιμότητα των multiview βίντεο, θα πρέπει να δούμε κάποια ζητήματα που θα πρέπει να λαμβάνονται υπόψη κατά στον σχεδιασμό κωδικοποιητή και αποκωδικοποιητή για multiview βίντεο με χρήση του προτύπου H.264/MPEG-4.

Το βασικό ζήτημα που τίθεται είναι το πως θα μπορέσουν να εξαλειφθούν οι εξαρτήσεις μεταξύ των διαφορετικών καμερών που λαμβάνουν την ίδια εικόνα. Θα πρέπει ο σχεδιαστής του κωδικοποιητή να λάβει υπόψη το πως μπορεί να αποβάλλει την ίδια εικόνα που είναι ίδια από διαφορετική κάμερα. Το θέμα είναι να έχει από κάθε κάμερα κάτι το ξεχωριστό σε σύγκριση με τις προηγούμενες. Εδώ έρχονται τα μοντέλα αναφοράς, ένα από τα οποία βλέπουμε στο σχήμα 3.1. Ο σχεδιαστής θα πρέπει να έχει στο μυαλό του κάποια μοντέλα αναφοράς, ώστε να μπορέσει να εκμεταλλευτεί αυτές τις εξαρτήσεις προς όφελος του σχεδιασμού του.

Εξίσου, σημαντικό είναι ότι το διάγραμμα ρυθμού-παραμόρφωσης γίνεται και χειρότερο κάθε φορά που έχουμε και μεγαλύτερο αριθμό καμερών. Το ζητούμενο είναι όσο ανεβαίνει ο αριθμός των καμερών να μπορούμε να διατηρούμε το διάγραμμα σε σταθερό ρυθμό.

Επίσης, θα πρέπει ο πίνακας που θα κρατήσουμε την H.264 ακολουθία να είναι αρκετά μεγάλος και να αλλάζει κάθε φορά που έχουμε μία νέα ακολουθία για αποθήκευση.

Σημαντικό θέμα είναι επίσης η συμπίεση. Ολοένα και περισσότερα δεδομένα είναι αναγκαίο να αποθηκεύονται οπότε θα πρέπει να έχουμε καλή συμπίεση μέσω της κωδικοποίησης μας. Στα αποτελέσματα που θα δούμε η συμπίεση που έχουμε είναι αρκετά ικανοποιητική.

3.4. Εφαρμογές των multiview βίντεο

Εξετάσαμε κάποια κύρια στοιχεία για τα multiview βίντεο. Όμως, είναι ιδιαίτερα σημαντικό να δούμε και τι εφαρμογές μπορούν να έχουν στην καθημερινότητα μας. Σε αυτή λοιπόν την ενότητα θα δούμε κάποιες κύριες εφαρμογές των multiview βίντεο.

3.4.1. 3-D βίντεο

Ίσως, η πιο σημαντική εφαρμογή των multiview βίντεο είναι το 3-D βίντεο. Για αυτή την εφαρμογή απαραίτητος είναι ο κατάλληλος εξοπλισμός τόσο για την παράσταση του βίντεο, όσο και να το δει κάποιος ανάλογος με το σχήμα 3.2. Στο 3-D βίντεο έχουμε μία 3-D βάθους αναπαράσταση μίας εικόνας και αντίστοιχα ενός βίντεο.



Σχήμα 3.2: 3-D γυαλιά απαραίτητα στο να δει κάποιος κανονικά ένα 3-D βίντεο.

Σημαντικές εφαρμογές που απορρέουν από την λειτουργία του 3-D βίντεο είναι οι τηλεδιασκέψεις και 3-D TV, που τώρα έχει μπει στην αγορά με μεγάλους ρυθμούς ανάπτυξης.

3.4.2. Βίντεο ελεύθερης θέασης

Άλλη μία σημαντική εφαρμογή των multiview βίντεο είναι το βίντεο ελεύθερης θέασης (freewviewpoint video). Σε ένα τέτοιο βίντεο δίνεται η δυνατότητα για εύκολη αλλαγή οπτικής γωνίας, αφού έχουμε λήψη με πολλές κάμερες. Σημαντικές εφαρμογές που απορρέουν από την λειτουργία το βίντεο πολλαπλής θέασης είναι η παρακολούθηση ενός χώρου, η κάλυψη αθλητικών γεγονότων και συναυλιών με μεγαλύτερη ευκολία για προσαρμογή κάμερας.

3.4.3. Απεικόνιση με μεγάλη ευκρίνεια

Άλλη μία σημαντική εφαρμογή των multiview βίντεο είναι η απεικόνιση με μεγάλη ευκρίνεια (high performance imaging). Πολλές κάμερες μαζί μπορούν να απεικονίσουν ένα βίντεο με μεγαλύτερη ευκρίνεια. Αυτό συμβαίνει τόσο για βίντεο όσο και αν χρειαστεί απλά παράσταση μίας εικόνας. Οπότε σημαντική εφαρμογή που συνδέεται είναι η παρακολούθηση ενός χώρου.

3.5. Βασικά στοιχεία για την κωδικοποίηση και αποκωδικοποίηση multiview βίντεο

Στο προηγούμενο κεφάλαιο είδαμε κάποια χρήσιμα στοιχεία για την κωδικοποίηση και αποκωδικοποίηση βίντεο με χρήση μίας κάμερας. Επίσης αναφέρθηκε ότι υπάρχουν στοιχεία είναι στην διάθεση του αναγνώστη να ανατρέξει στο *encoder.cfg* ώστε να βρει τι κάνουν.

Αυτά τα στοιχεία παραμένουν ανάλογα και για την κωδικοποίηση και αποκωδικοποίηση βίντεο με χρήση μίας κάμερας. Η σημαντική διαφορά βρίσκεται στην είσοδο και έξοδο που παίρνουμε μετά την κωδικοποίηση. Χρήσιμα στοιχεία που έχουμε στο *encoder.cfg* για τα multiview βίντεο είναι τα παρακάτω:

```
#####
#####
#Multi View Coding
#####
#####

MultiViewCount      = 8# 1 monoscopic, else number of cameras
InputFileStructure  = LeavingLaptop_Cam%d.yuv          #overwrites inputFile - Starts
from 0
ReconFileStructure  = outLeaving_Laptop1reference.yuv    #overwrites
ReconFile - Starts from 0
ReferenceMode        = 1  # 1:..... , 2:..... , 3:..... , 4:..... , #AnalyzeFile      =
"analyze_flamenco1_mono"
StandardCompatible   = 0
```

Στο *MultiViewCount* τοποθετούμε των αριθμό των καμερών που έχουμε για το βίντεο που θέλουμε (ή να τεθεί καλύτερα τα διαφορετικά βίντεο από διαφορετική οπτική γωνία που έχουμε).

Στο *InputFileStructure* θα πρέπει να βάλουμε το όνομα του βίντεο που θέλουμε να γίνει κωδικοποίηση. Αν για παράδειγμα θέλουμε το *LeavingLaptop* θα πρέπει να βάλουμε το όνομα του, βασιζόμενοι στην ιδέα ότι θα πρέπει να βάλουμε την αρχική μας κάμερα στον αριθμό 0 μέχρι τον αριθμό που θέλουμε (αν π.χ. έχουμε 8 βίντεο τότε μέχρι το 7).

Στο *ReconFileStructure* τοποθετούμε το όνομα του βίντεο που θέλουμε να έχει ότι παραχθεί μετά από το τρέξιμο. Ουσιαστικά παράγεται ένα βίντεο από μία μέση οπτική γωνία από όλα όσα χρησιμοποιήθηκαν.

Στο *ReferenceMode* τοποθετούμε ποιο μοντέλο αναφοράς θέλουμε να χρησιμοποιήσουμε. Επειδή βάση αυτού έχουμε τα διάφορα τρεξίματα που έγιναν ακολουθή ανάλυση στην επόμενη ενότητα.

Στα άλλα κρατάμε κάποια βασικά στοιχεία τα οποία θα μας δίνουν στο τέλος σαν στατιστικά στοιχεία.

Θα πρέπει να σημειώσουμε ότι θα πρέπει να λάβουμε και άλλα κομμάτια υπόψη. Στο *Output File* θα έχουμε το buffer στο οποίο θα βρίσκεται το H.264 αρχείο μας. Επίσης θα πρέπει να έχουμε βάλει *NumberReferenceFrames* σε μεγαλύτερο αριθμό από το *MultiViewCount* που έχουμε τοποθετήσει. Αλλιώς δεν θα μπορούμε να πραγματοποιήσουμε κάποιο τρέξιμο.

Κατά την αποκωδικοποίηση δεν συναντάμε καμία διαφορά σε σύγκριση με την αποκωδικοποίηση βίντεο με χρήση μίας κάμερας. Αν όλα πάνε καλά θα πρέπει να μας επιστρέφεται SNR 0 για κάθε Y,U και V συνιστώσα.

3.6. Μοντέλα αναφοράς

Καλό είναι πριν προχωρήσουμε στην παρουσίαση των αποτελεσμάτων για τα βίντεο Door_Flowers και Leaving_Laptop που είναι στην διάθεση του αναγνώστη καλό είναι να παρουσιαστούν τα μοντέλα πρόβλεψης που χρησιμοποιήθηκαν για τα διάφορα τρεξίματα που έγιναν. Συνολικά χρησιμοποιήθηκαν 4 μοντέλα αναφοράς (reference modes).

3.6.1. 1^ο Μοντέλο αναφοράς

Για να μπορέσουμε να τρέξουμε το multiview video μας με το πρώτο μοντέλο αναφοράς αρκεί να κάνουμε μία αλλαγή όπως φαίνεται εδώ:

```
#Multi View Coding
```

```
#####  
#####
```

```
MultiViewCount      = 8# 1 monoscopic, else number of cameras
```

```
InputFileStructure   = LeavingLaptop_Cam%d.yuv          #overwrites inputFile - Starts  
from 0
```

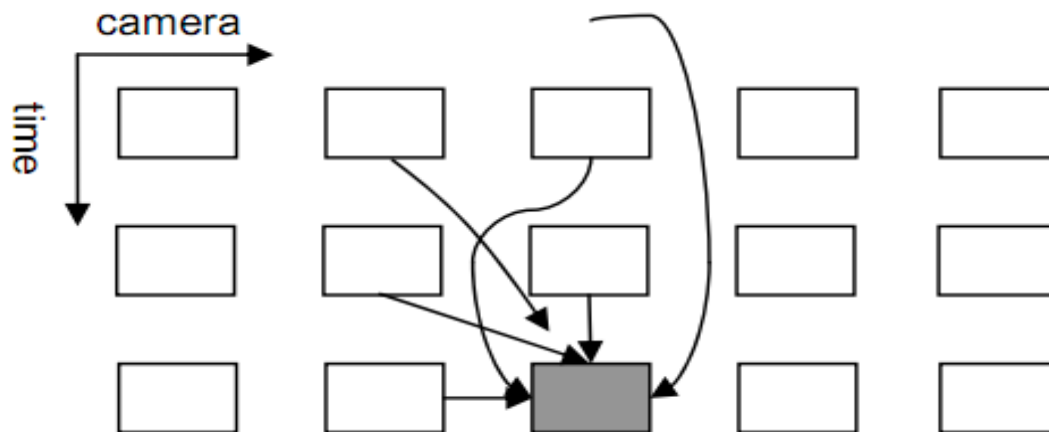
```
ReconFileStructure   = outLeaving_Laptop1reference.yuv    #overwrites  
ReconFile - Starts from 0
```

```
ReferenceMode        = 1  # 1:..... , 2:..... , 3:..... , 4:..... ,
```

```
#AnalyzeFile         = "analyze_flamenco1_mono"
```

```
StandardCompatible    = 0
```

Σε αυτό το μοντέλο αναφοράς έχουμε το ακόλουθο σχήμα:



Σχήμα 3.3 :1^ο reference mode.

Λαμβάνονται υπόψη τα frames που έχουν κωδικοποιηθεί από προηγούμενη κοντινή κάμερα καθώς και προηγούμενα frames από την κάμερα που κωδικοποιείται εκείνη την χρονική στιγμή.

3.6.2. 2^ο Μοντέλο αναφοράς

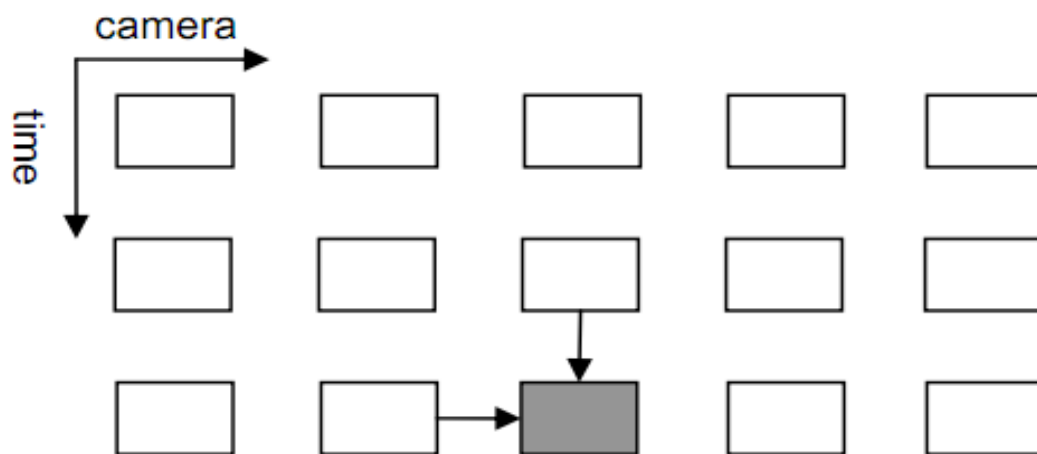
Για να μπορέσουμε να τρέξουμε το multiview video μας με το δεύτερο μοντέλο αναφοράς αρκεί να κάνουμε μία αλλαγή όπως φαίνεται εδώ:

```
#Multi View Coding
```

```
#####  
#####
```

```
MultiViewCount      = 8# 1 monoscopic, else number of cameras  
InputFileStructure  = LeavingLaptop_Cam%d.yuv          #overwrites inputFile - Starts  
from 0  
ReconFileStructure  = outLeaving_Laptop1reference.yuv    #overwrites  
ReconFile - Starts from 0  
ReferenceMode        = 2  # 1:..... , 2:..... , 3:..... , 4:..... ,  
#AnalyzeFile         = "analyze_flamenco1_mono"  
StandardCompatible   = 0
```

Σε αυτό το μοντέλο αναφοράς έχουμε το ακόλουθο σχήμα:



Σχήμα 3.4 :2^ο reference mode.

Λαμβάνονται υπόψη μόνο το κοντύτερο frame από την κάμερα που έχει προηγουμένως κωδικοποιηθεί καθώς και το προηγούμενο frame από την κάμερα που κωδικοποιείται εκείνη την χρονική στιγμή.

3.6.3. 3^ο Μοντέλο αναφοράς

Για να μπορέσουμε να τρέξουμε το multiview video μας με το τρίτο μοντέλο αναφοράς αρκεί να κάνουμε μία αλλαγή όπως φαίνεται εδώ:

```
#Multi View Coding
```

```
#####  
#####
```

```
MultiViewCount      = 8# 1 monoscopic, else number of cameras
```

```
InputFileStructure   = LeavingLaptop_Cam%d.yuv          #overwrites inputFile - Starts  
from 0
```

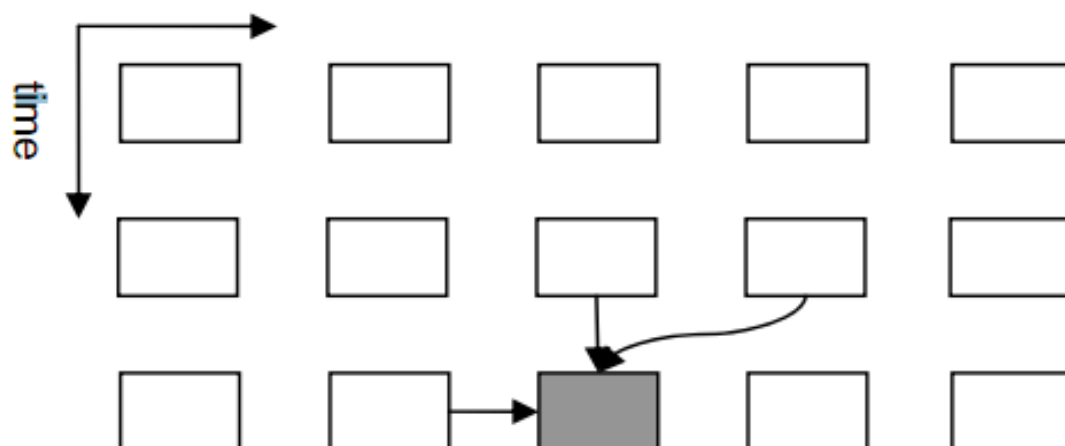
```
ReconFileStructure   = outLeaving_Laptop1reference.yuv    #overwrites  
ReconFile - Starts from 0
```

```
ReferenceMode        = 3  # 1:..... , 2:..... , 3:..... , 4:..... ,
```

```
#AnalyzeFile         = "analyze_flamenco1_mono"
```

```
StandardCompatible   = 0
```

Σε αυτό το μοντέλο αναφοράς έχουμε το ακόλουθο σχήμα:



Σχήμα 3.5: 3^ο reference mode.

Λαμβάνονται υπόψη τα κοντινά frame από τις δύο διπλανές κάμερες μία πριν και μία μετά καθώς και το προηγούμενο frame από την κάμερα που κωδικοποιείται εκείνη την χρονική στιγμή.

3.6.4. 4^ο Μοντέλο αναφοράς

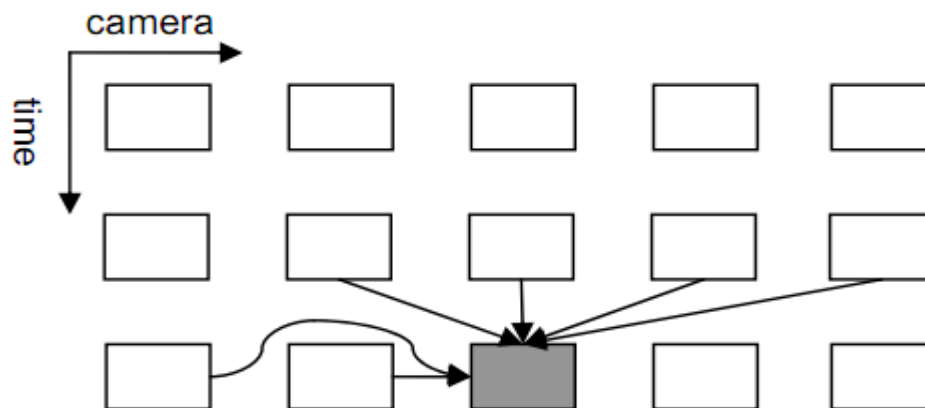
Για να μπορέσουμε να τρέξουμε το multiview video μας με τέταρτο μοντέλο αναφοράς αρκεί να κάνουμε μία αλλαγή όπως φαίνεται εδώ:

```
#Multi View Coding
```

```
#####  
#####
```

```
MultiViewCount      = 8# 1 monoscopic, else number of cameras  
InputFileStructure  = LeavingLaptop_Cam%d.yuv          #overwrites inputFile - Starts  
from 0  
ReconFileStructure  = outLeaving_Laptop1reference.yuv    #overwrites  
ReconFile - Starts from 0  
ReferenceMode       = 4  # 1:..... , 2:..... , 3:..... , 4:..... ,  
#AnalyzeFile        = "analyze_flamenco1_mono"  
StandardCompatible  = 0
```

Σε αυτό το μοντέλο αναφοράς έχουμε το ακόλουθο σχήμα:



Σχήμα 3.6 :4° reference mode.

Λαμβάνονται υπόψη κοντινά frame από όλες τις κάμερες καθώς και ένα frame από μία κοντινή κάμερα σε σχέση με αυτό το frame που κωδικοποιείται εκείνη την χρονική στιγμή.

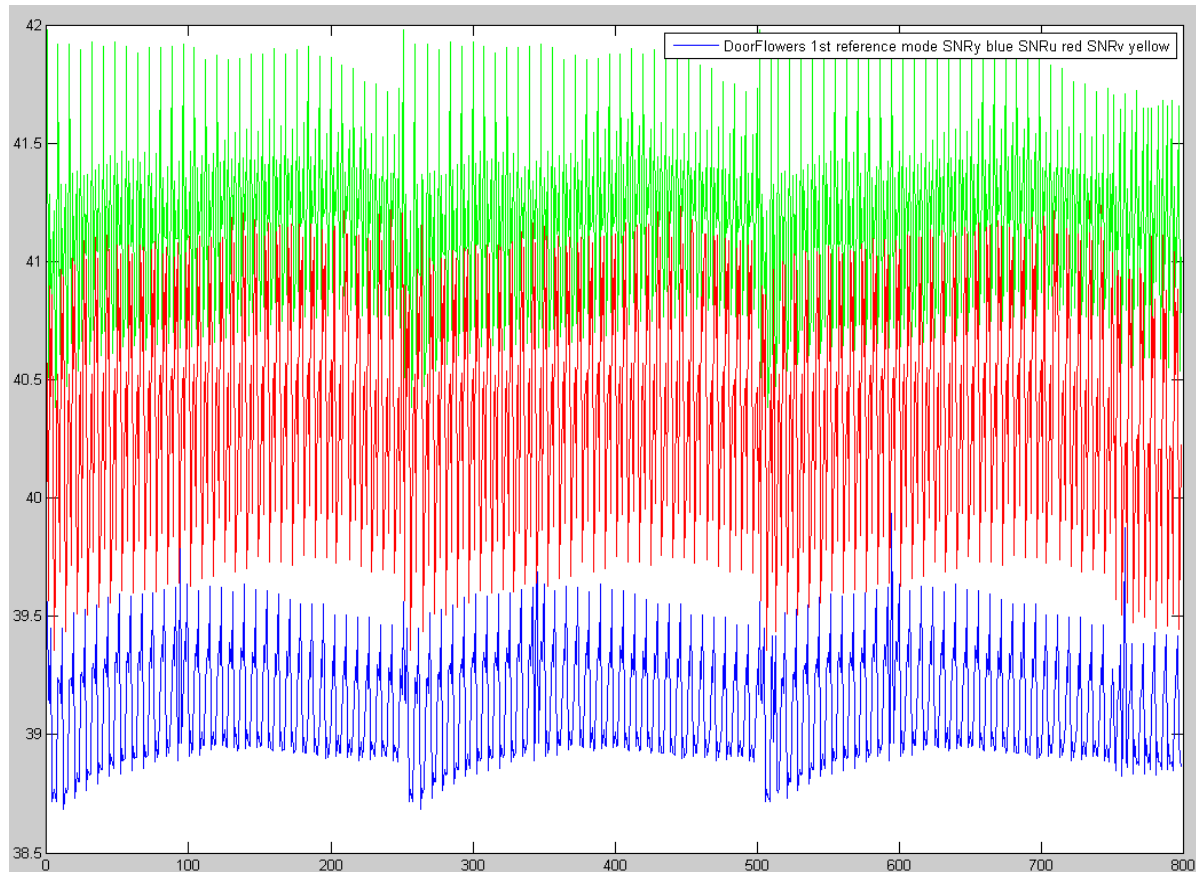
3.7. Door_Flowers multiview

Σε αυτή την ενότητα θα εξετάσουμε τα αποτελέσματα που πήραμε, τρέχοντας τα Door_Flowers βίντεο σαν 8άδες για δούμε τα αποτελέσματα σαν multiview βίντεο. Έγιναν συνολικά 4 τρεξίματα το καθένα για διαφορετικό reference mode. Τα περισσότερα στοιχεία παρέμειναν αμετάβλητα, πέρα από το reference mode που άλλαζε κάθε φορά. Έχουμε λοιπόν τα ακόλουθα στοιχεία σταθερά:

- 100 frames για κάθε βίντεο του multiview, άρα συνολικά 800 frames. Τοποθετείται 100 στο *FrameToBeEncoded*
- 1024X768 διαστάσεις άρα τοποθετήθηκαν 1024 και 768 στο *SourceWidth* και *SourceHeight* αντίστοιχα.
- Χρησιμοποιείται stereo high profile, αφού έχουμε multiview video για αυτό τοποθετούμε 100 στην μεταβλητή *ProfileIDC*.
- Τοποθετούμε 8 στο *MultiViewCount*, αφού βάζουμε 8 κάμερες.
- Τοποθετούμε *DoorFlower_Cam%d.yuv* στο *InputFileStructure*.
- Το βίντεο που παίρνουμε σαν αποτέλεσμα είναι το *outdoorflowers_cif* και τοποθετούμε αυτή την ονομασία στο *ReconFileStructure* και έχουμε test.264 όπου και κρατάμε το αρχείο H.264 που θα χρησιμοποιηθεί και κατά την αποκωδικοποίηση. Σε καθένα από τα 4 διαφορετικά βίντεος που θα βγάλουμε θα βάλουμε αυτό το όνομα
- Χρησιμοποιείται 4:2:0 μέθοδος υποδειγματοληψίας χρώματος. Οπότε τοποθετείται 1 στο *YUVFormat*.
- Έχουμε YUV βίντεο οπότε τοποθετούμε 0 στο *RGBInput*.
- Βάζουμε 16 στο *NumberOfReferenceFrames* απλά επειδή πρέπει να ξεπερνάει το *MultiViewCount*.

3.7.1. Door_Flowers 1^ο μοντέλο αναφοράς

Ας δούμε εδώ τα αποτελέσματα που παίρνουμε αν χρησιμοποιήσουμε το 1st reference mode:



Σχήμα 3.7: SNR για το Door Flowers για 8 βίντεο χρησιμοποιώντας το 1 reference mode. Με μπλε το SNR της Y συνιστώσας, με κόκκινο της U και με πράσινο της V.

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

Total Frames: 800 (800)

LeakyBucketRate File does not exist. Using rate calculated from avg. rate

Number Leaky Buckets: 8

Rmin Bmin Fmin

707940 2187704 792822

884910 751529 751529

1061880 710236 710236

1238850 668943 668943

1415820 627650 627650

1592790 586357 586357

1769760 545064 545064

1946730 503771 503771

Freq. for encoded bitstream : 30

Hadamard transform : Used

Image format : 1024x768

Error robustness : Off

Search range : 16

Total number of references : 16

References for P slices : 16

Total encoding time for the seq. : 28181.109 sec (0.03 fps)

Total ME time for sequence : 26669.190 sec

Sequence type : IPPP (QP: I 28, P 28)

Entropy coding method : CAVLC

Profile/Level IDC : (77,51)

Motion Estimation Scheme : Full Search

Search range restrictions : none

RD-optimized mode decision : not used

Data Partitioning Mode : 1 partition

Output File Format : H.264 Bit Stream File Format

Residue Color Transform : not used

----- *Average data all frames* -----

SNR Y(dB) : 39.03

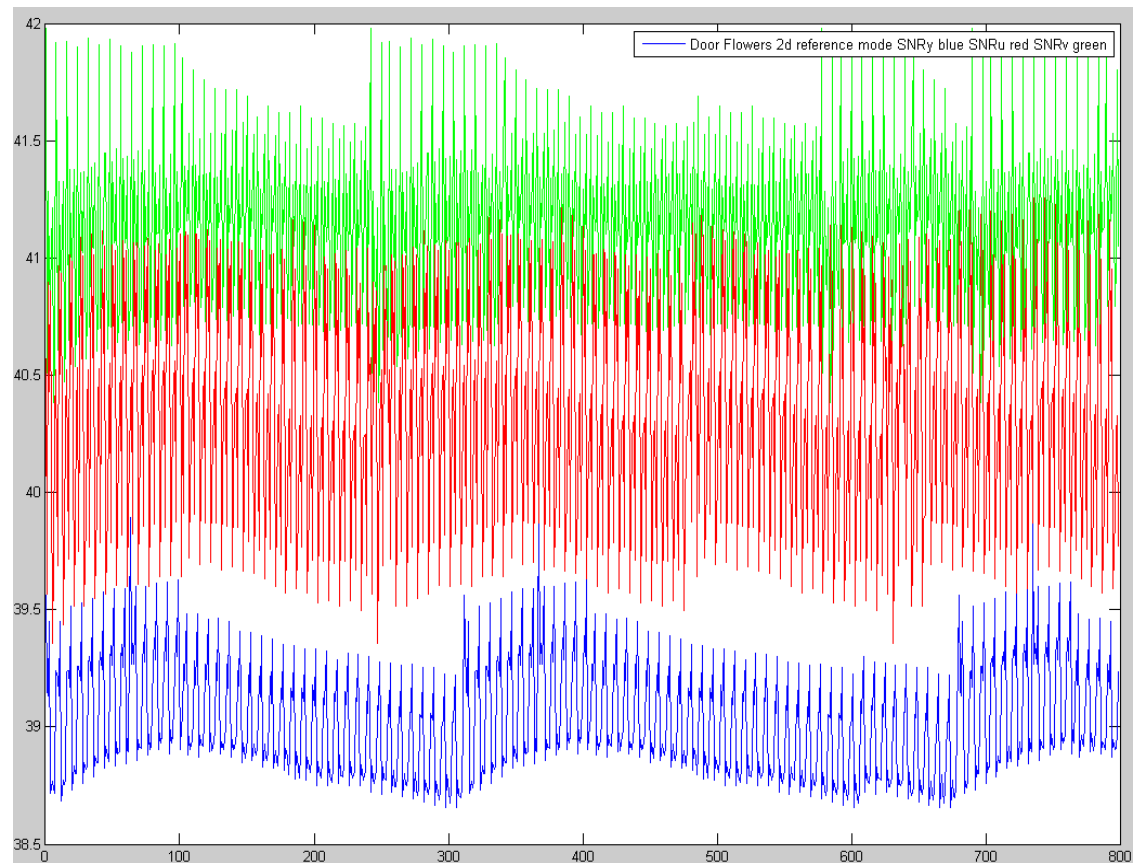
SNR U(dB) : 40.32

SNR V(dB) : 41.16
cSNR Y(dB) : 39.03 (8.13)
cSNR U(dB) : 40.28 (6.09)
cSNR V(dB) : 41.15 (4.99)
Total bits : 18878880 (I 228984, P 18649504, NVB 392)
Bit rate (kbit/s) @ 30.00 Hz : 707.96
Bits to avoid Startcode Emulation : 0

Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλαμε το βίντεο outDoor_flowers που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

3.7.2. Door_Flowers 2^ο μοντέλο αναφοράς

Ας δούμε εδώ τα αποτελέσματα που παίρνουμε αν χρησιμοποιήσουμε το 2nd reference mode:



Σχήμα 3.8: SNR για το Door Flowers για 8 βίντεο χρησιμοποιώντας το 2 reference mode. Με μπλε το SNR της Y συνιστώσας, με κόκκινο της U και με πράσινο της V.

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

Total Frames: 800 (800)

LeakyBucketRate File does not exist. Using rate calculated from avg. rate

Number Leaky Buckets: 8

Rmin	Bmin	Fmin
702180	2152008	794166
877710	753209	753209
1053240	712252	712252
1228770	671295	671295
1404300	630338	630338
1579830	589381	589381
1755360	548424	548424
1930890	507467	507467

Freq. for encoded bitstream	: 30
Hadamard transform	: Used
Image format	: 1024x768
Error robustness	: Off
Search range	: 16
Total number of references	: 16
References for P slices	: 16
Total encoding time for the seq.	: 13628.640 sec (0.06 fps)
Total ME time for sequence	: 12406.776 sec
Sequence type	: IPPP (QP: I 28, P 28)
Entropy coding method	: CAVLC
Profile/Level IDC	: (66,51)
Motion Estimation Scheme	: Full Search
Search range restrictions	: none
RD-optimized mode decision	: not used
Data Partitioning Mode	: 1 partition
Output File Format	: H.264 Bit Stream File Format
Residue Color Transform	: not used

----- Average data all frames -----

SNR Y(dB) : 39.03

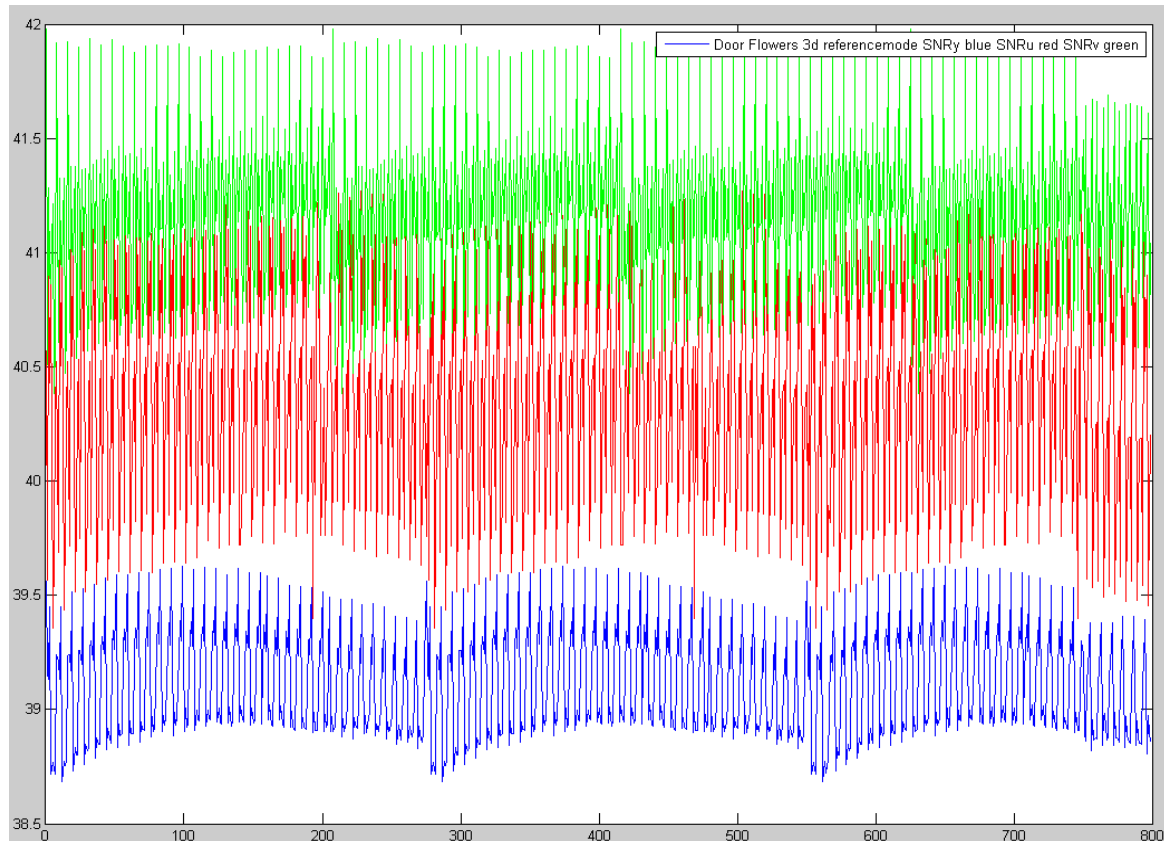
SNR U(dB) : 40.32

SNR V(dB) : 41.16
cSNR Y(dB) : 39.02 (8.14)
cSNR U(dB) : 40.28 (6.09)
cSNR V(dB) : 41.15 (4.99)
Total bits : 18725568 (I 228984, P 18496192, NVB 392)
Bit rate (kbit/s) @ 30.00 Hz : 702.21
Bits to avoid Startcode Emulation : 0
Bits for parameter sets : 392

Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλαμε το βίντεο outDoor_flowers που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

3.7.3. Door_Flowers 3^ο μοντέλο αναφοράς

Ας δούμε εδώ τα αποτελέσματα που παίρνουμε αν χρησιμοποιήσουμε το 3rd reference mode:



Σχήμα 3.9: SNR για το Door Flowers για 8 βίντεο χρησιμοποιώντας το 3 reference mode. Με μπλε το SNR της Y συνιστώσας, με κόκκινο της U και με πράσινο της V.

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

Total Frames: 800 (800)

LeakyBucketRate File does not exist. Using rate calculated from avg. rate

Number Leaky Buckets: 8

Rmin	Bmin	Fmin
702180	2152008	794166
877710	753209	753209
1053240	712252	712252
1228770	671295	671295
1404300	630338	630338
1579830	589381	589381
1755360	548424	548424
1930890	507467	507467

Freq. for encoded bitstream	: 30
Hadamard transform	: Used
Image format	: 1024x768
Error robustness	: Off
Search range	: 16
Total number of references	: 16
References for P slices	: 16
Total encoding time for the seq.	: 13585.692 sec (0.06 fps)
Total ME time for sequence	: 12401.252 sec
Sequence type	: IPPP (QP: I 28, P 28)
Entropy coding method	: CAVLC
Profile/Level IDC	: (66,51)
Motion Estimation Scheme	: Full Search
Search range restrictions	: none
RD-optimized mode decision	: not used
Data Partitioning Mode	: 1 partition
Output File Format	: H.264 Bit Stream File Format
Residue Color Transform	: not used

----- Average data all frames -----

SNR Y(dB) : 39.03

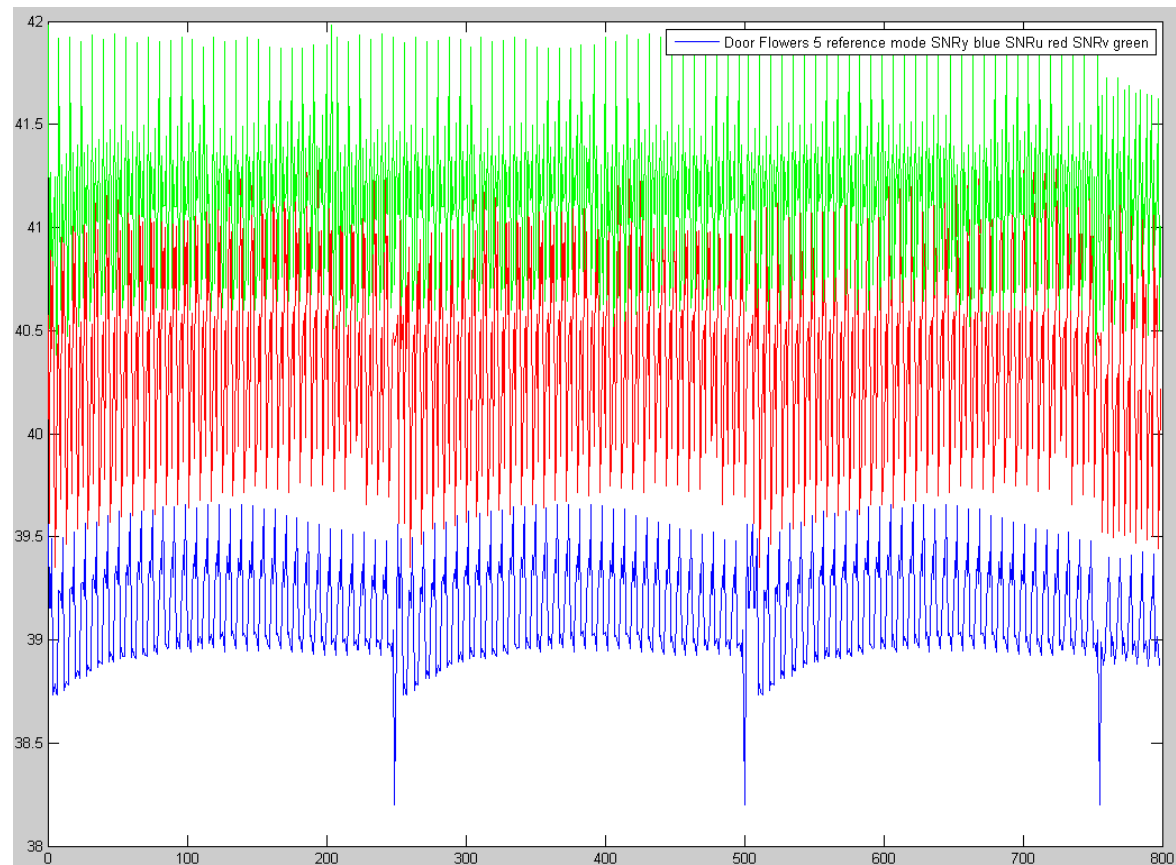
SNR U(dB) : 40.32

SNR V(dB) : 41.16
cSNR Y(dB) : 39.02 (8.14)
cSNR U(dB) : 40.28 (6.09)
cSNR V(dB) : 41.15 (4.99)
Total bits : 18725568 (I 228984, P 18496192, NVB 392)
Bit rate (kbit/s) @ 30.00 Hz : 702.21
Bits to avoid Startcode Emulation : 0
Bits for parameter sets : 392

Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλαμε το βίντεο outDoor_flowers που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

3.7.4. Door_Flowers 4^ο μοντέλο αναφοράς

Ας δούμε εδώ τα αποτελέσματα που παίρνουμε αν χρησιμοποιήσουμε το 4th reference mode:



Σχήμα 3.10: SNR για το Door Flowers για 8 βίντεο χρησιμοποιώντας το 4 reference mode. Με μπλε το SNR της Y συνιστώσας, με κόκκινο της U και με πράσινο της V.

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

Total Frames: 800 (800)

LeakyBucketRate File does not exist. Using rate calculated from avg. rate

Number Leaky Buckets: 8

Rmin	Bmin	Fmin
693630	2082310	782337
867030	741877	741877
1040430	701417	701417
1213830	660957	660957
1387230	620497	620497
1560630	580037	580037
1734030	539577	539577
1907430	499117	499117

Freq. for encoded bitstream	: 30
Hadamard transform	: Used
Image format	: 1024x768
Error robustness	: Off
Search range	: 16
Total number of references	: 16
References for P slices	: 16
Total encoding time for the seq.	: 63145.310 sec (0.01 fps)
Total ME time for sequence	: 61624.709 sec
Sequence type	: IPPP (QP: I 28, P 28)
Entropy coding method	: CAVLC
Profile/Level IDC	: (66,51)
Motion Estimation Scheme	: Full Search
Search range restrictions	: none
RD-optimized mode decision	: not used
Data Partitioning Mode	: 1 partition
Output File Format	: H.264 Bit Stream File Format
Residue Color Transform	: not used

----- *Average data all frames* -----

SNR Y(dB) : 39.08

SNR U(dB) : 40.31

SNR V(dB) : 41.16
cSNR Y(dB) : 39.07 (8.06)
cSNR U(dB) : 40.27 (6.11)
cSNR V(dB) : 41.14 (5.00)
Total bits : 18497680 (I 228984, P 18268304, NVB 392)
Bit rate (kbit/s) @ 30.00 Hz : 693.66
Bits to avoid Startcode Emulation : 0
Bits for parameter sets : 392

Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλουμε το βίντεο outDoor_flowers που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

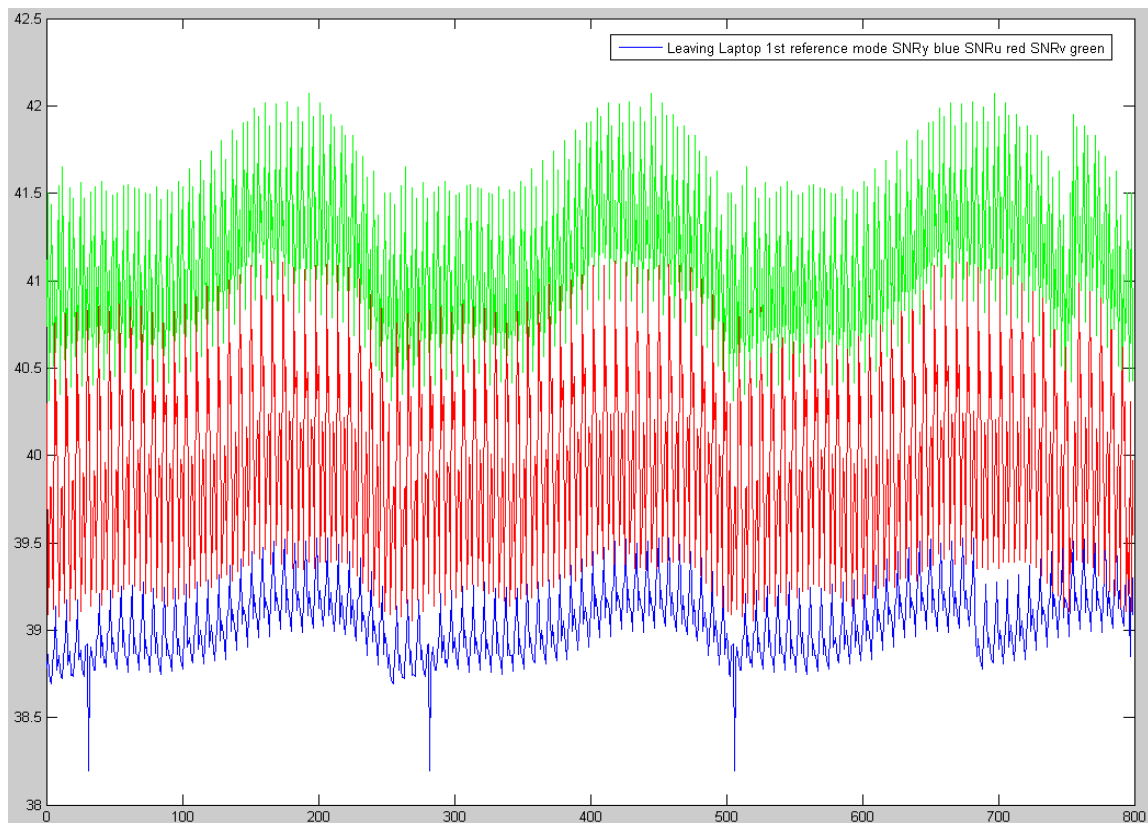
3.8. Leaving_Laptop multiview

Σε αυτή την ενότητα θα εξετάσουμε τα αποτελέσματα που πήραμε, τρέχοντας τα Leaving Laptop βίντεο σαν 8άδες για δούμε τα αποτελέσματα σαν multiview βίντεο. Έγιναν συνολικά 4 τρεξίματα το καθένα για διαφορετικό reference mode. Τα περισσότερα στοιχεία παρέμειναν αμετάβλητα, πέρα από το reference mode που άλλαζε κάθε φορά. Έχουμε λοιπόν τα ακόλουθα στοιχεία σταθερά:

- 100 frames για κάθε βίντεο του multiview, άρα συνολικά 800 frames. Τοποθετείται 100 στο *FrameToBeEncoded*
- 1024X768 διαστάσεις άρα τοποθετήθηκαν 1024 και 768 στο *SourceWidth* και *SourceHeight* αντίστοιχα.
- Χρησιμοποιείται stereo high profile, αφού έχουμε multiview video για αυτό τοποθετούμε 100 στην μεταβλητή *ProfileIDC*.
- Τοποθετούμε 8 στο *MultiViewCount*, αφού βάζουμε 8 κάμερες.
- Τοποθετούμε *LeavingLaptop_Cam%d.yuv* στο *InputFileStructure*.
- Το βίντεο που παίρνουμε σαν αποτέλεσμα είναι το *outLeavingLaptop_cif* και τοποθετούμε αυτή την ονομασία στο *ReconFileStructure* και έχουμε test.264 όπου και κρατάμε το αρχείο H.264 που θα χρησιμοποιηθεί και κατά την αποκωδικοποίηση. Σε κάθενα από τα 4 διαφορετικά βίντεος που θα βγάλουμε θα βάλουμε αυτό το όνομα
- Χρησιμοποιείται 4:2:0 μέθοδος υποδειγματοληψίας χρώματος. Οπότε τοποθετείται 1 στο *YUVFormat*.
- Έχουμε YUV βίντεο οπότε τοποθετούμε 0 στο *RGBInput*.
- Βάζουμε 16 στο *NumberOfReferenceFrames* απλά επειδή πρέπει να ξεπερνάει το *MultiViewCount*.

3.8.1. Leaving_Laptop 1^ο μοντέλο αναφοράς

Ας δούμε εδώ τα αποτελέσματα που παίρνουμε αν χρησιμοποιήσουμε το 1st reference mode:



Σχήμα 3.11: SNR για το Leaving_Laptop για 8 βίντεο χρησιμοποιώντας το 1 reference mode.
Με μπλε το SNR της Y συνιστώσας, με κόκκινο της U και με πράσινο της V.

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

Total Frames: 800 (800)

LeakyBucketRate File does not exist. Using rate calculated from avg. rate

Number Leaky Buckets: 8

Rmin Bmin Fmin

463909 3657008 640228

579882 864311 591529

695855 542830 542830

811828 494131 494131

927801 445432 445432

1043774 396733 396733

1159747 348034 348034

1275720 299335 299335

Freq. for encoded bitstream : 17

Hadamard transform : Used

Image format : 1024x768

Error robustness : Off

Search range : 16

Total number of references : 16

References for P slices : 16

Total encoding time for the seq. : 26327.493 sec (0.03 fps)

Total ME time for sequence : 24984.766 sec

Sequence type : IPPP (QP: I 28, P 28)

Entropy coding method : CAVLC

Profile/Level IDC : (77,51)

Motion Estimation Scheme : Full Search

Search range restrictions : none

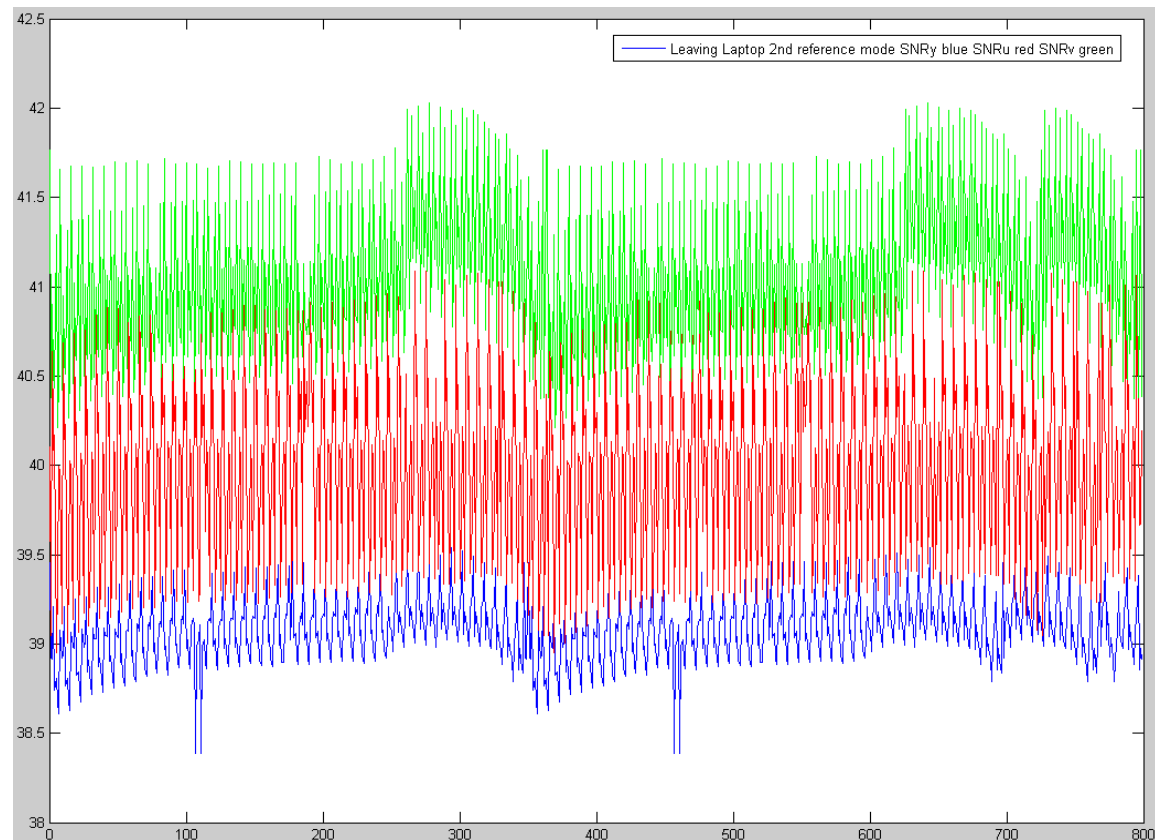
RD-optimized mode decision : not used

Data Partitioning Mode : 1 partition
Output File Format : H.264 Bit Stream File Format
Residue Color Transform : not used
----- Average data all frames -----
SNR Y(dB) : 39.02
SNR U(dB) : 39.98
SNR V(dB) : 41.05
cSNR Y(dB) : 39.02 (8.15)
cSNR U(dB) : 39.95 (6.58)
cSNR V(dB) : 41.03 (5.12)
Total bits : 22264336 (I 239008, P 22024936, NVB 392)
Bit rate (kbit/s) @ 16.67 Hz : 463.93
Bits to avoid Startcode Emulation : 0
Bits for parameter sets : 392

Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλαμε το βίντεο outLeaving_Laptop που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

3.8.2. Leaving_Laptop 2^ο μοντέλο αναφοράς

Ας δούμε εδώ τα αποτελέσματα που παίρνουμε αν χρησιμοποιήσουμε το 2nd reference mode:



Σχήμα 3.12: SNR για το Leaving_Laptop για 8 βίντεο χρησιμοποιώντας το 2 reference mode.
Με μπλε το SNR της Y συνιστώσας, με κόκκινο της U και με πράσινο της V.

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

Total Frames: 800 (800)

LeakyBucketRate File does not exist. Using rate calculated from avg. rate

Number Leaky Buckets: 8

Rmin Bmin Fmin

461275 3593070 641334

576581 835757 592915

691887 544496 544496

807193 496077 496077

922499 447658 447658

1037805 399239 399239

1153111 350820 350820

1268417 302401 302401

Freq. for encoded bitstream : 17

Hadamard transform : Used

Image format : 1024x768

Error robustness : Off

Search range : 16

Total number of references : 16

References for P slices : 16

Total encoding time for the seq. : 13785.408 sec (0.06 fps)

Total ME time for sequence : 12577.497 sec

Sequence type : IPPP (QP: I 28, P 28)

Entropy coding method : CAVLC

Profile/Level IDC : (77,51)

Motion Estimation Scheme : Full Search

Search range restrictions : none

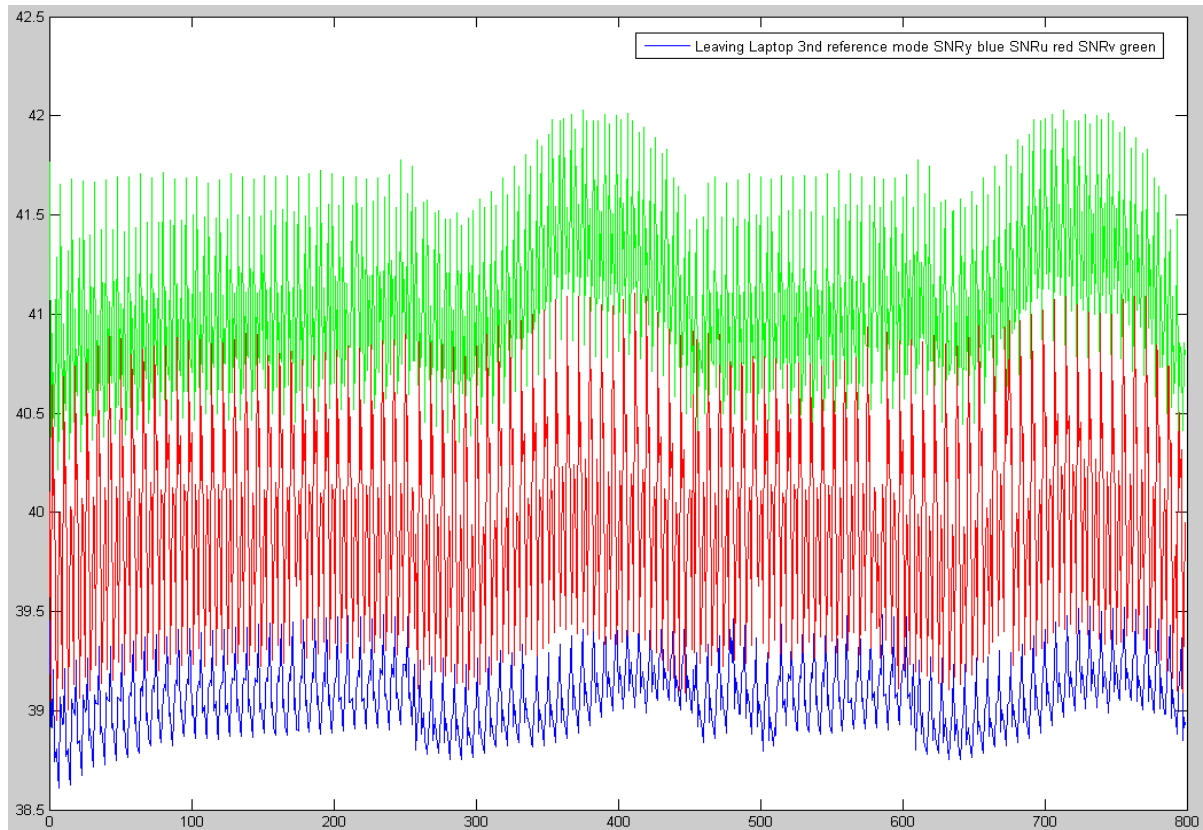
RD-optimized mode decision : not used

Data Partitioning Mode : 1 partition
Output File Format : H.264 Bit Stream File Format
Residue Color Transform : not used
----- Average data all frames -----
SNR Y(dB) : 39.02
SNR U(dB) : 39.98
SNR V(dB) : 41.05
cSNR Y(dB) : 39.01 (8.16)
cSNR U(dB) : 39.95 (6.58)
cSNR V(dB) : 41.03 (5.13)
Total bits : 22137872 (I 239008, P 21898472, NVB 392)
Bit rate (kbit/s) @ 16.67 Hz : 461.30
Bits to avoid Startcode Emulation : 0
Bits for parameter sets : 392

Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλαμε το βίντεο outLeaving_Laptop που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

3.8.3. Leaving_Laptop 3^ο μοντέλο αναφοράς

Ας δούμε εδώ τα αποτελέσματα που παίρνουμε αν χρησιμοποιήσουμε το 3rd reference mode:



Σχήμα 3.13: SNR για το Leaving_Laptop για 8 βίντεο χρησιμοποιώντας το 3 reference mode.
Με μπλε το SNR της Y συνιστώσας, με κόκκινο της U και με πράσινο της V.

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

Total Frames: 800 (800)

LeakyBucketRate File does not exist. Using rate calculated from avg. rate

Number Leaky Buckets: 8

Rmin Bmin Fmin

467710 3686736 638632

584633 923916 589534

701556 540436 540436

818479 491338 491338

935402 442240 442240

1052325 393142 393142

1169248 344044 344044

1286171 294946 294946

Freq. for encoded bitstream : 17

Hadamard transform : Used

Image format : 1024x768

Error robustness : Off

Search range : 16

Total number of references : 16

References for P slices : 16

Total encoding time for the seq. : 19498.731 sec (0.04 fps)

Total ME time for sequence : 18259.153 sec

Sequence type : IPPP (QP: I 28, P 28)

Entropy coding method : CAVLC

Profile/Level IDC : (77,51)

Motion Estimation Scheme : Full Search

Search range restrictions : none

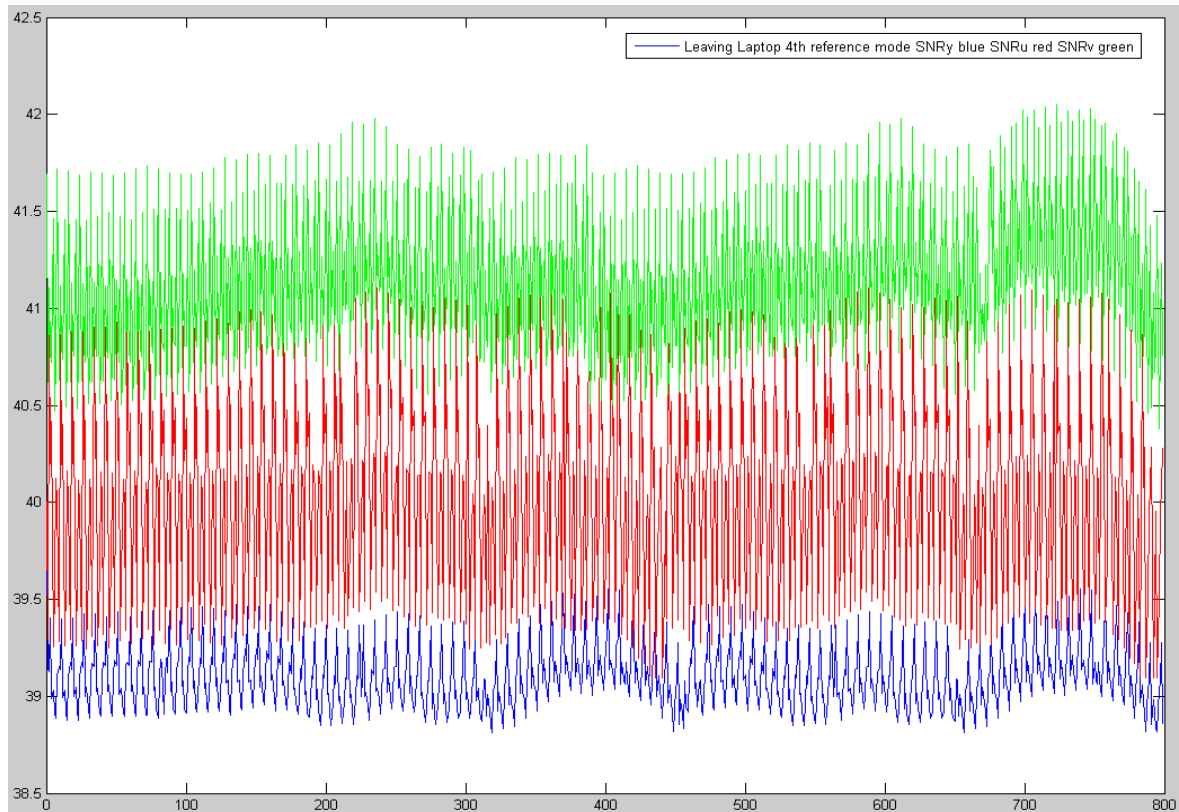
RD-optimized mode decision : not used

Data Partitioning Mode : 1 partition
Output File Format : H.264 Bit Stream File Format
Residue Color Transform : not used
----- Average data all frames -----
SNR Y(dB) : 39.02
SNR U(dB) : 39.98
SNR V(dB) : 41.06
cSNR Y(dB) : 39.02 (8.15)
cSNR U(dB) : 39.95 (6.58)
cSNR V(dB) : 41.04 (5.12)
Total bits : 22446296 (I 239008, P 22206896, NVB 392)
Bit rate (kbit/s) @ 16.67 Hz : 467.72
Bits to avoid Startcode Emulation : 0
Bits for parameter sets : 392

Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλαμε το βίντεο outLeaving_Laptop που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

3.8.4. Leaving_Laptop 4^ο μοντέλο αναφοράς

Ας δούμε εδώ τα αποτελέσματα που παίρνουμε αν χρησιμοποιήσουμε το 4th reference mode:



Σχήμα 3.14: SNR για το Leaving_Laptop για 8 βίντεο χρησιμοποιώντας το 4 reference mode.
Με μπλε το SNR της Y συνιστώσας, με κόκκινο της U και με πράσινο της V.

Επίσης στο τέλος της εκτέλεσης πήραμε τα ακόλουθα στοιχεία:

Total Frames: 800 (800)

LeakyBucketRate File does not exist. Using rate calculated from avg. rate

Number Leaky Buckets: 8

Rmin Bmin Fmin

467476 3713032 632226

584332 928176 583156

701188 534086 534086

818044 485016 485016

934900 435946 435946

1051756 386876 386876

1168612 337806 337806

1285468 288736 288736

Freq. for encoded bitstream : 17

Hadamard transform : Used

Image format : 1024x768

Error robustness : Off

Search range : 16

Total number of references : 16

References for P slices : 16

Total encoding time for the seq. : 62155.884 sec (0.01 fps)

Total ME time for sequence : 60693.769 sec

Sequence type : IPPP (QP: I 28, P 28)

Entropy coding method : CAVLC

Profile/Level IDC : (77,51)

Motion Estimation Scheme : Full Search

Search range restrictions : none

RD-optimized mode decision : not used

Data Partitioning Mode : 1 partition
Output File Format : H.264 Bit Stream File Format
Residue Color Transform : not used
----- Average data all frames -----
SNR Y(dB) : 39.03
SNR U(dB) : 39.98
SNR V(dB) : 41.06
cSNR Y(dB) : 39.02 (8.14)
cSNR U(dB) : 39.95 (6.58)
cSNR V(dB) : 41.04 (5.12)
Total bits : 22435096 (I 239008, P 22195696, NVB 392)
Bit rate (kbit/s) @ 16.67 Hz : 467.49
Bits to avoid Startcode Emulation : 0
Bits for parameter sets : 392

Κατά την αποκωδικοποίηση όλα πήγαν ομαλά και αυτό το καταλαβαίνουμε από το 0 που πήραμε για τα αποτελέσματα του SNR για κάθε συνιστώσα. Σαν είσοδο για την αποκωδικοποίηση βάλαμε το βίντεο outLeaving_Laptop που πήραμε κατά την κωδικοποίηση καθώς και το test.264 που βγάλαμε σαν αποτέλεσμα το H.264.

ΚΕΦΑΛΑΙΟ 4

ΕΠΙΛΟΓΟΣ

Είδαμε σε αυτά τα κεφάλαια κάποια στοιχεία για την ιστορία και λειτουργία του H.264/MPEG-4. Επίσης είδαμε αποτελέσματα από την κωδικοποίηση και αποκωδικοποίηση κάποιων βίντεο τόσο με χρήση μίας όσο και περισσότερων (συγκεκριμένα 8) καμερών.

Από όλα τα διαγράμματα μπορούμε να καταλήξουμε στο συμπέρασμα ότι υπάρχει μία αναλογία στα SNR ανεξάρτητα από το τι βίντεο έχουμε. Το SNR της Υ συνιστώσας είναι το χαμηλότερο μετά ακολουθεί της U και στο τέλος το μεγαλύτερο είναι της V.

Πέρα από αυτά τα αποτελέσματα είδαμε και κάποια οφέλη τα οποία έχουμε με την χρήση του H.264/MPEG-4. Είδαμε εργαλεία τα οποία χρησιμοποιούνται καθώς και το τι μας εξυπηρετούν. Στο μέλλον είναι σίγουρο ότι τόσο του πρότυπο H.264/MPEG-4 όσο και τυχόν βελτιώσεις τους θα έχουν ολοένα μεγαλύτερο ρόλο στην καθημερινή μας ζωή. Ήδη στην Ελλάδα με την έναρξη του ψηφιακού σήματος κάθε τηλεόραση χρειάζεται (αν δεν έχει ενσωματωμένο) αποκωδικοποιητή MPEG-4 έτσι ώστε να μπορεί να υπάρξει κατάλληλη μετατροπή σήματος. Οι Blu-Ray δίσκοι χρησιμοποιούν αποκλειστικά το H.264/MPEG-4 πρότυπο για συμπίεση και κωδικοποίηση. Σίγουρα λοιπόν στο μέλλον και άλλες συσκευές θα μπουں στην ζωή μας λόγω του προτύπου αυτού.

ΑΝΑΦΟΡΕΣ

1. ISO/IEC 14496 and ITU-T Rec. "H.264 Advanced Video Coding", 2003.
2. A. Hallapuro, M. Karczewicz and H. Malvar, "Low Complexity Transform and Quantization Part 1: Basic Implementation, JVT document JVT- B038", Geneva, February 2002.
3. Iain E. G. Richardson, "H.264 and MPEG-4 VIDEO COMPRESSION" pages 110-123.
4. G. Bjontegaard and K. Lillevold, "Context-adaptive VLC coding of coefficients, JVT document JVT-C028", Fairfax, May 2002.
5. S.W. Golomb, "Run-length encoding", pp. 399-401, 1966.
6. Yao Wang, Polytechnic University Brooklyn, "Video Coding Using Motion Compensation", 2002.
7. H. Schwartz, D. Marpe and T. Wiegand, "CABAC and slices, JVT document JVT-D020", Klagenfurt, Austria, July, 2002.
8. "Color subsampling"
<http://encyclopedia2.thefreedictionary.com/chroma+subsampling>.
9. Huifang Sun and Video Team Mitsubishi Electric Research Laboratories, "Multiview Video Coding".
10. "Video information" :
<http://sp.cs.tut.fi/mobile3dvt/stereo-video/> .
11. Philipp Merkle, Member IEEE, Aljoscha Smolic, Karsten Muller, Senior Member, IEEE, and Thomas Wiegand, Member, IEEE, "Efficient Prediction Structures for Multi-view Video Coding".
12. Markus Flierl and Bernd Girod "Multiview Video Compression", November 2007.
13. Yo-Sung Ho and Kwan-Jung Oh, "Overview of Multi-view Video Coding", 2007.

