

Τ.Ε.Ι ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΙ ΜΕΤΑΔΟΣΗ ΒΙΝΤΕΟ

Λαμπίρης Νικόλαος

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων
ΚΑΡΚΑΝΗΣ ΣΤΑΥΡΟΣ
ΚΑΘΗΓΗΤΗΣ

Λαμία 2014

«Εγώ, Λαμπίρης Νικόλαος, δηλώνω υπεύθυνα ότι το περιεχόμενο της εργασίας που περιέχεται σε αυτό το βιβλίο είναι αποκλειστικά δικό μου. Σε όποιο σημείο περιέχονται πληροφορίες από άλλες πηγές, δηλώνω ότι αυτό επισημαίνεται στην εργασία.

Ονοματεπώνυμο: Λαμπίρης Νικόλαος

Υπογραφή:

Η έγκριση της εργασίας από το **Τμήμα Μηχανικών Πληροφορικής Τ.Ε.** του **Τ.Ε.Ι. Στερεάς Ελλάδας** δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος (*N.5343/32 αρ.202 παρ.2*).

Αυτή η εργασία χορηγείται με άδεια **Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0 Διεθνής Άδεια**. Για να δείτε αντίγραφο της άδειας, επισκεφθείτε <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.el>



ΕΥΧΑΡΙΣΤΙΕΣ

Η ολοκλήρωση αυτής της πτυχιακής εργασίας υλοποιήθηκε με την υποστήριξη ενός αριθμού ανθρώπων στους οποίους θα ήθελα να εκφράσω τις θερμότερες ευχαριστίες μου. Πρώτα από όλους θα ήθελα να **ΕΥΧΑΡΙΣΤΗΣΩ** τους γονείς μου που χωρίς την υποστήριξη τους και τη συμβολή τους αυτά τα χρόνια δεν θα είχα φτάσει εδώ που είμαι τώρα. Θα ήθελα επίσης να ευχαριστήσω τον κ. *Σταύρο Καρκάνη* για την πολύτιμη βοήθεια του και τη συνεργασία που είχαμε όχι μόνο κατά τη διάρκεια της εργασίας αλλά και στα μαθήματα του στα οποία συμμετείχα αλλά και τις εμπειρίες που απέκτησα για να μπορέσω να φέρω σε πέρας την παρούσα εργασία. Τέλος, να ευχαριστήσω τον φίλο και συνάδελφο, συμφοιτητή, *Γαβριήλ Μανιά* για την υπομονή του αλλά και την άμεση ανταπόκριση όποτε τον χρειαζόμουν, ο οποίος αποδείχτηκε πολύτιμος συνεργάτης κατά την διάρκεια των δικτυακών δοκιμών που αφορούσε συγκεκριμένο κομμάτι της εργασίας.

Λαμπίρης Νικόλαος

Σεπτέμβριος 2014

ΠΕΡΙΛΗΨΗ

Στις μέρες μας η ψηφιακή επεξεργασία βίντεο είναι ένας μεγάλος κλάδος που συνεχώς εξελίσσεται και διαθέτει αρκετές εφαρμογές και ποικίλα ερευνητικά πεδία για όποιον ενδιαφέρεται να ασχοληθεί. Βρίσκεται στα σύνορα με την ψηφιακή επεξεργασία εικόνας μόνο που ένα βίντεο αποτελείται από πολλές εικόνες όχι μόνο μια οι οποίες διαδέχονται η μία την άλλη με ένα συγκεκριμένο ρυθμό. Συνεπώς έχουμε να κάνουμε με πολλές εικόνες όπου κάθε μία έχει τα δικά της χαρακτηριστικά και πρέπει να αναλυθεί ξεχωριστά. Το συμπέρασμα είναι ότι η ψηφιακή επεξεργασία βίντεο είναι σε κάθε περίπτωση πιο απαιτητική από την άποψη των υπολογιστικών πόρων. Υπάρχουν διάφορες εφαρμογές που γίνεται χρήση της ψηφιακής επεξεργασίας βίντεο όπως είναι συστήματα παρακολούθησης, συστήματα υγείας, κλπ.

Στην παρούσα εργασία θα ασχοληθούμε με την εξαγωγή των **καρέ** (*frames*) από ένα βίντεο χρησιμοποιώντας διάφορους αλγορίθμους και τη μετάδοση του επεξεργασμένου βίντεο μέσω δικτύου με τη χρήση του πρωτοκόλλου TCP και της **διεπαφής υποδοχών** (*socket interface*). Στόχος την εργασίας είναι να βρεθούν αλγόριθμοι που θα επιλέγουν τα καρέ τα οποία μας δίνουν τις πιο σημαντικές σκηνές του βίντεο. Χρησιμοποιούμε διαφορετικά είδη βίντεο έτσι ώστε να έχουμε ένα καλό εύρος αποτελεσμάτων για να κρίνουμε την αποδοτικότητα των αλγορίθμων. Αρχικά κάνουμε κάποιους υπολογισμούς χρησιμοποιώντας διάφορους αλγορίθμους εξάγοντας χαρακτηριστικά από το κάθε καρέ όπως είναι οι τιμές των pixels και στη συνέχεια με κάποιο κατώφλι γίνεται η επιλογή των καρέ που θα αποτελούν το τελικό βίντεο. Στη συνέχεια, δημιουργούμε από αυτά τα μαρκαρισμένα καρέ ένα βίντεο το οποίο έχει τη χρονική διάρκεια του αρχικού. Τέλος πραγματοποιείται η μετάδοση του επεξεργασμένου βίντεο με κάποιες αλλαγές εξοικονομώντας δικτυακούς πόρους και κερδίζοντας σε απόδοση.

Λέξεις κλειδιά: βίντεο, επεξεργασία, μετάδοση, μαρκάρισμα καρέ, pixel, σύγκριση, διαφορές, κωδικοποίηση, MPEG-1, H.264, OpenCV, C

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ

ΠΕΡΙΛΗΨΗ

1. ΕΙΣΑΓΩΓΗ.....	6
1.1 ΕΙΣΑΓΩΓΗ ΣΤΟ ΧΩΡΟ ΤΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΒΙΝΤΕΟ.....	6
1.2 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	6
1.3 ΔΟΜΗ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	6
2. ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΒΙΝΤΕΟ.....	8
2.1 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ.....	8
2.1.1 ΒΙΝΤΕΟ.....	8
2.1.2 ΚΑΡΕ.....	9
2.1.3 ΑΝΑΛΥΣΗ ΒΙΝΤΕΟ.....	10
2.1.4 COMPRESSION - CONTAINER.....	10
2.1.5 FPS (Frames Per Second).....	11
2.1.6 ΟΙΚΟΓΕΝΕΙΑ ΠΡΩΤΟΚΟΛΛΩΝ TCP/IP.....	12
2.1.7 Berkeley Sockets.....	12
3 ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ.....	14
3.1 ΛΟΓΙΣΜΙΚΟ.....	14
3.1.1 ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΡΕ.....	17
3.1.1.1 ΑΛΓΟΡΙΘΜΟΙ.....	17
3.1.2 ΜΑΡΚΑΡΙΣΜΑ ΚΑΡΕ.....	20
3.1.2.1 ΑΛΓΟΡΙΘΜΟΙ.....	21
3.1.2.2 ΠΑΡΑΜΕΤΡΟΙ - ΣΕΝΑΡΙΑ.....	24
3.1.3 ΜΕΤΑΔΟΣΗ ΚΑΡΕ.....	27
3.1.3.1 ΠΡΟΓΡΑΜΜΑ ΕΞΥΠΗΡΕΤΗΤΗ.....	28
3.1.3.2 ΠΡΟΓΡΑΜΜΑ ΠΕΛΑΤΗ.....	29
3.2 ΥΛΙΚΟ (ΒΙΝΤΕΟ).....	30
4. ΠΕΙΡΑΜΑΤΑ.....	31
5. ΑΠΟΤΕΛΕΣΜΑΤΑ - ΣΥΜΠΕΡΑΣΜΑΤΑ.....	35
6. ΒΙΒΛΙΟΓΡΑΦΙΑ.....	37

ΠΑΡΑΡΤΗΜΑ

1. ΕΙΣΑΓΩΓΗ

1.1 Εισαγωγή στο χώρο της ψηφιακής επεξεργασίας βίντεο

Η ψηφιακή επεξεργασία βίντεο είναι παρακλάδι της επεξεργασίας σήματος, μιας και μπορούμε να ορίσουμε το βίντεο σαν ένα σήμα.

Η εξέλιξη της τεχνολογίας και των μέσων παραγωγής πολυμεσικού υλικού έχει οδηγήσει σε μεγάλο όγκο διαθέσιμης πληροφορίας. Αυτό από τη μια λειτουργεί ως πλεονέκτημα, γιατί ο καθένας μπορεί να βρει άφθονες πηγές πληροφοριών για ένα θέμα που τον ενδιαφέρει, αλλά από την άλλη υπάρχει πάντα ο κίνδυνος να χαθεί μέσα στον όγκο της πληροφορίας. Συνεπώς δημιουργείται η επιτακτική ανάγκη για αποδοτική αξιοποίηση της πληροφορίας, με τη χρήση βεβαίως των κατάλληλων μέσων.

1.2 Αντικείμενο της εργασίας

Η εργασία αυτή έχει ως αντικείμενο την ψηφιακή επεξεργασία βίντεο. Σκοπός της εργασίας είναι η εξαγωγή χαρακτηριστικών από το κάθε καρέ του βίντεο χρησιμοποιώντας διάφορες μεθόδους και στη συνέχεια με τη βοήθεια διαφόρων αλγορίθμων γίνεται η επιλογή συγκεκριμένων καρέ τα οποία στη συνέχεια ενώνονται για να αποτελέσουν ένα νέο βίντεο ή αν είναι μικρός ο αριθμός τους εξάγονται ως εικόνες. Στη συνέχεια το επεξεργασμένο βίντεο μεταδίδεται μέσω δικτύου χρησιμοποιώντας τη μέθοδο επικοινωνίας πελάτη – εξυπηρετητή και με τη βοήθεια της οικογένειας πρωτοκόλλων TCP/IP. Ουσιαστικά περιέχουν την ουσία του βίντεο μειώνοντας έτσι την ποσότητα των δεδομένων αρκετά και κάνοντας πιο παραγωγική την εύρεση και κατανόηση του επιθυμητού περιεχομένου.

1.3 Δομή της εργασίας

Στο **δεύτερο κεφάλαιο** γίνεται παρουσίαση των βασικών εννοιών όσον αφορά την ψηφιακή επεξεργασία βίντεο. Είναι μια εισαγωγή σε κάποιους ορισμούς που θα συναντήσουμε στη συνέχεια. Έννοιες όπως καρέ ανά δευτερόλεπτο ή ανάλυση βίντεο είναι θεμελιώδης για να κατανοήσουμε τη συνέχεια.

Στο **τρίτο κεφάλαιο** γίνεται αναφορά στο λογισμικό και τις μεθόδους που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας και την παραγωγή των αποτελεσμάτων. Γίνετε λεπτομερή αναφορά στους αλγορίθμους αλλά και τις παραμέτρους που εφαρμόστηκαν αλλά και στον τρόπο

μετάδοσης του τελικού βίντεο. Τέλος παρουσιάζεται το υλικό πάνω στο οποίο εφαρμόστηκαν οι αλγόριθμοι όπου κατά πλειοψηφία είναι ιατρικά βίντεο.

Στο **τέταρτο** και **πέμπτο κεφάλαιο** καταγράφονται τα πειράματα, τα σενάρια που εφαρμόστηκαν για κάθε βίντεο ακολουθούμενα από τις παραμέτρους που εφαρμόστηκαν ενώ στη συνέχεια παρουσιάζονται τα αποτελέσματα αυτών των μετρήσεων με τα κάποια συμπεράσματα που εξάγονται από αυτά.

Το **έκτο κεφάλαιο** περιέχει τις βιβλιογραφικές αναφορές που χρησιμοποιήθηκαν για την περάτωση της εργασίας.

2. ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΒΙΝΤΕΟ

2.1 Βασικές έννοιες

2.1.1 Βίντεο

Ο γενικός ορισμός είναι ότι το βίντεο είναι ένα σήμα και στην περίπτωση μας ένα ψηφιακό σήμα. Το βίντεο είναι μια ροή εικόνων οι οποίες διαδέχονται η μία την άλλη με συγκεκριμένο ρυθμό. Αυτή η τεχνική δημιουργεί την ψευδαίσθηση ότι υπάρχει συνεχή κίνηση. Τα πρώτα χρόνια ένα βίντεο ήταν ασπρόμαυρο και έτσι περιείχε ουσιαστικά μια πληροφορία. Μετέπειτα με τη δημιουργία του έγχρωμου βίντεο δημιουργήθηκε το πρόβλημα της μετάδοσης αυτού καθώς το μέγεθος του μεγάλωσε και συνεπώς έπρεπε να βρεθούν τεχνικές για την αποδοτική και αποτελεσματική μετάδοση του.

Περνώντας στο επίπεδο του τρόπου μετάδοσης του βίντεο, εδώ ουσιαστικά υπάρχουν κάποια πρότυπα που είναι κυρίως πιο γνωστά στην τηλεόραση αλλά και ψηφιακά μέσα όπως είναι τα DVD. Τα **NTSC**, **PAL** και **SECAM** είναι πρότυπα που χρησιμοποιούνται σε διάφορες χώρες του κόσμου και μεταξύ άλλων καθορίζουν και τη συμβατότητα μεταξύ των συσκευών στις εκάστοτε χώρες με τα συγκεκριμένα πρότυπα. Αυτό σημαίνει ότι μια συσκευή κατασκευασμένη στην Ευρώπη για παράδειγμα χρησιμοποιεί το πρότυπο PAL και έτσι είναι αδύνατο να αναπαράγει κάποιο βίντεο από την Αμερική που έχει παραχθεί σύμφωνα με το πρότυπο NTSC.

Για την προβολή ενός βίντεο είναι απαραίτητος και κάποιος εξοπλισμός ο οποίος πρέπει να διαχειριστεί αυτό το ψηφιακό σήμα. Μιας και το βίντεο αποτελείται από εικόνες πρέπει και οι οθόνες που προβάλλονται να έχουν και ένα συγκεκριμένο αριθμό ανανέωσης. Οι περισσότερες οθόνες υπολογιστών είτε επίπεδες (*TFT/LCD*) είτε όχι (*CRT*) έχουν ένα συγκεκριμένο ρυθμό ανανέωσης. Για τις TFT/LCD είναι γύρω στα 60 Hz ενώ για τις CRT είναι γύρω στα 70 Hz. Οι οθόνες τηλεοράσεων με τη βοήθεια και της εξέλιξης της τεχνολογίας μπορούν να φτάσουν και τα 120 Hz. Πρέπει να σημειωθεί εδώ ότι ο ρυθμός ανανέωσης είναι ανεξάρτητος του ρυθμού εναλλαγής των καρέ σε ένα βίντεο (*FPS, Frames Per Second*).

Ένα σημαντικό ακόμα χαρακτηριστικό που αφορά το **υλικό** (*hardware*) είναι ο τρόπος με τον οποίο γίνεται η παρουσίαση των καρέ στην οθόνη του χρήστη. Σίγουρα η λογική απάντηση είναι ότι κάθε καρέ εμφανίζεται ολόκληρο και μετά προχωράμε στο επόμενο. Αυτή η μέθοδος ονομάζεται **progressive scan** όπου κάθε καρέ “ζωγραφίζεται” από την πάνω αριστερή γωνία έως στην κάτω δεξιά γωνία. Μια εναλλακτική μέθοδος η οποία είναι γενικά πιο αποδοτική και χρησιμοποιείται ευρέως είναι το **interlaced scan**. Με αυτή τη τεχνική μειώνεται ο όγκος της πληροφορίας που αποστέλλεται γιατί αρχικά εμφανίζονται οι άρτιες γραμμές και στη συνέχεια οι

περιττές γραμμές του καρέ. Οι περισσότερες οθόνες LCD χρησιμοποιούν progressive scan ενώ οθόνες CRT ή τηλεοπτικές μεταδόσεις χρησιμοποιούν interlaced scan.

Κλείνοντας αυτό το εισαγωγικό κομμάτι όσον αφορά το βίντεο αλλά και το υλικό αναπαραγωγής, πρέπει να έχουμε στο μυαλό μας ότι η τεχνολογία εξελίσσεται και συνεπώς τα δεδομένα αλλάζουν στο χώρο της μετάδοσης, προβολής και επεξεργασίας βίντεο. Πολλές εφαρμογές όπου βρίσκουν εφαρμογή τα παραπάνω εντοπίζονται στη παραγωγή ταινιών, ηλεκτρονικών παιχνιδιών, στην τηλεόραση αλλά και σε προσομοιωτές. Μερικά χαρακτηριστικά του βίντεο είναι τα εξής: Καρέ ανά δευτερόλεπτο, λόγος διαστάσεων, χρωματικός χώρος, ποιότητα, συμπίεση. Στη συνέχεια θα γίνει αναφορά σε κάποια από αυτά πιο αναλυτικά.

Σε μετέπειτα παραγράφους θα αναφερθούμε σε συγκεκριμένα χαρακτηριστικά του βίντεο.

2.1.2 Καρέ

Όπως προαναφέρθηκε και στην εισαγωγή, ένα βίντεο αποτελείται από στατικές εικόνες οι οποίες διαδέχονται η μία την άλλη με σταθερό ρυθμό. Η κάθε εικόνα (καρέ) που αποτελεί το βίντεο έχει τα ίδια χαρακτηριστικά (διαστάσεις, βάθος χρώματος κλπ) καθ' όλη τη διάρκεια αυτού και αποτελεί τον δομικό λίθο για ένα βίντεο. Ένα καρέ αποτελείται από σημεία ή αλλιώς pixels. Ο συνολικός αριθμός των pixels ισοδυναμεί με το γινόμενο του πλάτους και του ύψους του καρέ. Αν και θα αναφερθούμε εκτενέστερα παρακάτω σε αυτό καλό είναι να γνωρίζουμε ότι το ύψος και το πλάτος του καρέ μας δίνει και την ανάλυση του βίντεο.

Ανεξάρτητα από τον τύπο του βίντεο, το κάθε καρέ μπορεί να είναι **ασπρόμαυρο** (αποχρώσεις του γκρι) ή **έγχρωμο**. Αν το καρέ είναι ασπρόμαυρο τότε οι τιμές των pixels κυμαίνονται από $[0,255]$, σύνολο 256 τιμές, όπου μεγαλύτερες τιμές σημαίνουν λευκές αποχρώσεις ενώ τιμές κοντά στο 0 σημαίνουν σκούρες αποχρώσεις. Ουσιαστικά οι τιμές αυτές εκφράζουν την ένταση της φωτεινότητας στο εκάστοτε pixel. Στην περίπτωση που έχουμε έγχρωμο καρέ, τότε το κάθε ένα αποτελείται από **3 κανάλια**: **Κόκκινο** (Red), **Πράσινο** (Green) και **Μπλε** (Blue), το γνωστό χρωματικό σύστημα *RGB*. Κάθε κανάλι λαμβάνει τιμές από $[0,255]$ και ο συνδυασμός αυτών των 3 τιμών μπορεί να μας δώσει έως 16.777.216 αποχρώσεις. [Παπαμάρκος 2010] [Ωραιοζήλη 2011]

2.1.3 Ανάλυση βίντεο

Η **ανάλυση ενός βίντεο** (*video resolution*) είναι βασικά οι οριζόντιες και κάθετες γραμμές ενός βίντεο που καθορίζουν και τις διαστάσεις του βίντεο. Ανάλογα τον τύπο του υλικού που διαθέτουμε, στην προκειμένη περίπτωση μια οθόνη, είτε CRT είτε LCD, θα έχουμε και συγκεκριμένες αναλύσεις. Αυτές οι διαστάσεις μπορούν να μεταφραστούν σε έναν άλλο συνδυασμό ο οποίος είναι γνωστός και σαν **λόγος διαστάσεων** (*aspect ratio*). Στη συνέχεια η ανάλυση αυτή κανονικοποιείται σαν 1:1 λόγο διαστάσεων. Για τις οθόνες CRT ή τηλεοράσεων, ο λόγος διαστάσεων είναι 4:3 και οι αναλύσεις είναι σχετικά μικρές (π.χ. 1024x768, 720x576). Στις οθόνες LCD αλλά και στην ψηφιακή τηλεόραση (ψηφιακό σήμα) ο λόγος διαστάσεων είναι 16:9 (*widescreen*) συνεπώς έχουμε και μεγαλύτερες αναλύσεις (π.χ. 1920x1080). Για να γίνει πιο κατανοητό, η ανάλυση π.χ. 1920x1080 σημαίνει 1920 pixels οριζόντια και 1080 pixels κάθετα. Τα κάθετα pixels είναι αυτά που αναφέρονται και στις κατηγορίες αναλύσεων.

Έχουμε 3 κατηγορίες όπου μπορούν να καταχωρηθούν τα βίντεο: **Standard Definition (SD)**, **Enhanced Definition (ED)**, **High Definition (HD)**. Στην κατηγορία SD ανήκουν πολλά βίντεο μικρής ανάλυσης π.χ. 480, 576 αλλά και γενικά το τηλεοπτικό σήμα κάνουν χρήση παρόμοιων αναλύσεων. Στην ED κατηγορία βρίσκουμε τις ίδιες αναλύσεις με τη μόνη διαφορά ότι στην SD κατηγορία χρησιμοποιείται η τεχνική *interlaced scan* ενώ στην ED χρησιμοποιείται *progressive scan*. Στην τελευταία κατηγορία HD συναντάμε μεγαλύτερες αναλύσεις π.χ. 720, 1080, 2160 όπου προβάλλονται κυρίως από LCD οθόνες. Επίσης το ψηφιακό τηλεοπτικό σήμα εξελίσσεται και εμφανίζεται και σε μεγαλύτερες αναλύσεις.

2.1.4 Compression – Container

Στις μέρες μας έχουν αναπτυχθεί αρκετά πρότυπα σχετικά με τη συμπίεση του βίντεο για καλύτερη απόδοση στη μετάδοση και στην προβολή. Προς αυτή την κατεύθυνση βοήθησε και η εξέλιξη των συσκευών αναπαραγωγής πολυμέσων όπως είναι η ψηφιακή τηλεόραση, τα DVD players αλλά και τα κινητά τηλέφωνα. Τα προηγούμενα χρόνια χρησιμοποιούνταν ευρέως κωδικοποιήσεις MPEG-1 και MPEG-2 όπου για την εποχή ήταν αποδοτικές αλλά και για τις αναλύσεις που προβάλλονταν από τις εκάστοτε συσκευές. Τώρα πια έχουν αναπτυχθεί κωδικοποιήσεις όπου προσφέρουν καλύτερη απόδοση, μικρότερο μέγεθος κρατώντας την ποιότητα στο ίδιο επίπεδο με βάση την πηγή του βίντεο ή του σήματος.

Τα τελευταία χρόνια έχει αναπτυχθεί ένα πρότυπο κωδικοποίησης το οποίο χρησιμοποιείται ευρέως σε εμπορικά προϊόντα όπως είναι οι δίσκοι DVD ή BluRay που περιέχουν ταινίες, βίντεο που διανέμονται στο Internet καθώς και ψηφιακό σήμα από κάμερες ή το ψηφιακό τηλεοπτικό

σήμα το οποίο μπορεί να κωδικοποιηθεί αποδοτικά. Το **H.264/MPEG-4 Part 10 (High profile)** ή αλλιώς **AVC (Advanced Video Coding)** δημιουργήθηκε έτσι ώστε να υπάρχει ένα ενιαίο πρότυπο κωδικοποίησης πολυμεσικού περιεχομένου. Είναι το κυρίαρχο πρότυπο και ιδανικό και για μετάδοση βίντεο καθώς το μέγεθος διατηρείται χαμηλά ενώ παράλληλα η ποιότητα δεν αλλοιώνεται σε σχέση με άλλα πρότυπα. Επιπλέον, εκτός από την κωδικοποίηση του βίντεο υπάρχει και αυτή του ήχου η οποία χρησιμοποιείται ζευγάρι με το AVC και είναι το **AAC (Advanced Audio Coding)**. Το επόμενο πρότυπο που θα συναντάμε περισσότερο τα επόμενα χρόνια και φιλοδοξεί να αντικαταστήσει τον προκάτοχο του είναι το **HEVC/H.265**. Υπόσχεται διπλασιασμό στη συμπίεση του βίντεο διατηρώντας όμως την ποιότητα στα ίδια επίπεδα. Στις αρχές του 2013 δημοσιεύτηκε η πρώτη έκδοση του προτύπου το οποίο συνεχώς και εξελίσσεται με τους πρώτους **encoders** (προγράμματα κωδικοποίησης) να έχουν ήδη δημιουργηθεί. Με την παραγωγή βίντεο όλο και μεγαλύτερης ανάλυσης είναι απαραίτητο να βρεθεί μια κωδικοποίηση που θα μπορεί να πετύχει την μεγαλύτερη δυνατή συμπίεση για τον περιορισμό του χώρου που απαιτείται για αποθήκευση χωρίς να θυσιάζεται όμως η ποιότητα.[Sullivan et al 2012]

Στο συγκεκριμένο κομμάτι θα αναφερθούμε κυρίως για τα αρχεία βίντεο και τον τρόπο αποθήκευσης και αναγνώρισης τους από τα διάφορα λειτουργικά συστήματα (*Windows, Linux, Mac OS*). Εκτός από τα πρότυπα κωδικοποίησης που είναι υπεύθυνα για ποιότητα των δεδομένων (στην περίπτωση των αρχείων βίντεο ή ήχου) υπάρχουν και οι **τύποι αρχείων (file types)** που χρησιμοποιούνται για να μπορούν τα λειτουργικά συστήματα να αναγνωρίζουν ποια αρχεία περιέχουν πολυμέσα όπως είναι το βίντεο και ο ήχος. Με αυτόν τον τρόπο επιτυγχάνεται και η μετάδοση τέτοιων αρχείων μέσω δικτύου. Το πρότυπο **MIME (Multi-Purpose Internet Mail Extensions)** καθορίζει τον τρόπο αναγνώρισης τέτοιων αρχείων από τα διάφορα υπολογιστικά συστήματα τα οποία διακινούνται μέσω του Διαδικτύου. Κάθε αρχείο βίντεο μπορεί να έχει και έναν διαφορετικό τύπο ο οποίος ερμηνεύεται με βάση την κατάληξη που έχει το αρχείο π.χ. *video.mp4*. Μόνο από την κατάληξη του αρχείου δεν μπορούμε να εξάγουμε κάποιο συμπέρασμα για την κωδικοποίηση που έχει χρησιμοποιηθεί. Κάποιες συνήθεις καταλήξεις αρχείων βίντεο είναι: **mp4, mkv, avi, mpg, wmv, mov** κ.ά. Αυτές οι καταλήξεις ονομάζονται και **containers**. Είναι ουσιαστικά το "κουτί" που περιέχει τα δεδομένα του βίντεο. Τα περισσότερα από αυτά έχουν δημιουργηθεί από κοινοπραξίες οργανισμών ή εταιριών με σκοπό πάντα τη δημιουργία κάποιου προτύπου που θα είναι ευρέως διαδεδομένο. Μοναδική εξαίρεση αποτελεί το *mkv* όπου είναι ένα πρότυπο το οποίο δημιουργήθηκε από την "κοινότητα", είναι ανοιχτού κώδικα και τα τελευταία χρόνια έχει αρχίσει να χρησιμοποιείται ευρέως σε βίντεο σε αντίθεση με το *mp4*.

2.1.5 FPS (Frames Per Second)

Βασικό χαρακτηριστικό ενός βίντεο που ουσιαστικά καθορίζει και το πόσο καλά αποδίδεται το βίντεο. Το **FPS (Frames Per Second)** είναι ένα αριθμός που καθορίζει το πόσα καρέ

εναλλάσσονται ανά δευτερόλεπτο και είναι υπεύθυνο για διάφορα φαινόμενα που παρατηρούνται κατά την προβολή ενός βίντεο είτε με χαμηλό είτε υψηλό FPS. Όπως προαναφέραμε ένα βίντεο αποτελείται από εικόνες που εναλλάσσονται με σταθερό ρυθμό. Αν το FPS είναι αρκετά χαμηλό τότε το χάνει αισθητά σε κίνηση και συνεπώς υποβαθμίζεται η ποιότητα ενώ παράλληλα παρατηρούνται φαινόμενα όπως είναι το "θάμπωμα" περιοχών κατά τη διάρκεια του βίντεο όπου παρατηρείται γρήγορη κίνηση ενώ αν είναι αρκετά υψηλό τότε παρατηρούνται απότομες αλλαγές (πιο πραγματικές) ειδικά στις γρήγορες σκηνές. Το υψηλό FPS είναι αρκετά χρήσιμο όπου ένα βίντεο περιέχει γρήγορες σκηνές με απότομες αλλαγές έτσι ώστε να αντιμετωπίζεται η θαμπάδα που αναφερθήκαμε παραπάνω. Το κατώφλι για να θεωρηθεί μια ακολουθία από εικόνες βίντεο είναι γύρω στα 16 FPS. Τα περισσότερα βίντεο που χρησιμοποιήθηκαν στην παρούσα εργασία βρίσκονται γύρω στα 24 με 30 FPS με εξαιρέσεις κάποια από αυτά να βρίσκονται στα 50 FPS. [Jack 2007]

2.1.6 Οικογένεια πρωτοκόλλων TCP/IP

Τα δίκτυα και επικοινωνίες γενικότερα έχουν αλλάξει και εξελιχθεί σε σχέση με ότι υπήρχε 30 χρόνια πριν. Τότε τα δίκτυα μεταγωγής πακέτων ήταν ελάχιστα και συνήθως ανήκαν σε μεγάλες εταιρίες όπου χρησιμοποιούνταν εσωτερικά για ερευνητικούς ή επαγγελματικούς σκοπούς. Κάθε ένα από αυτά τα δίκτυα χρησιμοποιούσε το δικό του πρωτόκολλο για τη μεταφορά των πακέτων εντός του δικτύου και συνεπώς ήταν αδύνατο να υπάρξει επικοινωνία με κάποιον κόμβο κάποιου άλλου δικτύου. Μπορούμε να φανταστούμε το πρωτόκολλο σαν μια γλώσσα. Αν όλοι οι υπολογιστές δεν “μιλούν” την ίδια γλώσσα τότε δεν υφίστανται επικοινωνία. Σαν αποτέλεσμα αυτού, οι Vinton Cerf και Robert Kahn δημιούργησαν το **TCP/IP** (*Transmission Control Protocol/Internet Protocol*) έτσι ώστε να εξαιρεθεί αυτό το πρόβλημα. Στη συνέχεια όμως αυτά τα 2 πρωτόκολλα ακολούθησαν διαφορετική πορεία. [Cerf et Kahn 1974] [Kurose et Ross 2012]

2.1.7 Berkeley sockets

Οι **υποδοχές του Berkeley** (*Berkeley sockets*) δημιουργήθηκαν από το ομώνυμο πανεπιστήμιο με σκοπό να μεταφέρει την οικογένεια πρωτοκόλλων TCP/IP στο λειτουργικό σύστημα UNIX. Μέσα από αυτή την εργασία προέκυψε μια διεπαφή η οποία θα προοριζόταν για τη δημιουργία εφαρμογών οι οποίες θα επικοινωνούν μεταξύ τους. Ακόμα και στις μέρες μας, το **TCP** (*Transmission Control Protocol*) είναι το κυρίαρχο πρωτόκολλο επικοινωνίας που χρησιμοποιείται

τόσο στο **Διαδίκτυο** (*Internet, WAN, Wide Area Network*) όσο και σε **τοπικά δίκτυα** (*LAN, Local Area Network*). Πιο συγκεκριμένα το TCP είναι ένα **συνδεσμικό πρωτόκολλο** (*connection-oriented*) που σημαίνει ότι προτού εγκαθιδρυθεί η σύνδεση απαιτείται μια προκαταρκτική επικοινωνία πελάτη και εξυπηρετητή η οποία ονομάζεται και “**χειραψία**” (*handshake*) και ουσιαστικά είναι ένας τρόπος για να γίνει η ταυτοποίηση των συμμετεχόντων στην επικείμενη επικοινωνία. Με τη δημιουργία της σύνδεσης σημαίνει ότι μόνο ένας πελάτης και ένας εξυπηρετητής επικοινωνούν πάνω από αυτήν την σύνδεση και χρησιμοποιούν όλο το εύρος της, αλλιώς είναι γνωστό σαν **από-σημείου-προς-σημείο** επικοινωνία. Πρέπει να μην συγχέεται με την επικοινωνία **από άκρο σε άκρο** καθώς στην από-σημείου-προς-σημείο επικοινωνία οι παράμετροι σύνδεσης αποθηκεύονται τοπικά και δεν επηρεάζουν την επικοινωνία σε δρομολογητές από όπου περνάνε τα δεδομένα.

Περνώντας σε πιο τεχνικό επίπεδο, στην εργασία χρησιμοποιήθηκαν βιβλιοθήκες της γλώσσας C που περιγράφουν διάφορες συναρτήσεις σχετικά με την επικοινωνία με χρήση sockets. Το πρωτόκολλο που χρησιμοποιήθηκε είναι το TCP κυρίως για την ευκολία υλοποίησης των προγραμμάτων αλλά και την αξιοπιστία που προσφέρει στην μετάδοση των δεδομένων. [Comer et Stevens 2003]

3. ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

3.1 Λογισμικό

Για την περάτωση της εργασίας αλλά και των αποτελεσμάτων που θα παρουσιαστούν στη συνέχεια χρησιμοποιήθηκαν διάφορα προγραμματιστικά εργαλεία. Η επιλογή αυτών έγινε με κριτήριο την υποστήριξη όταν παραστεί ανάγκη αλλά και τη λειτουργικότητα και ευκολία υλοποίησης.

Πρώτα απ' όλα είναι βασικό να αναφερθούμε στις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την υλοποίηση των εφαρμογών που χρησιμοποιήθηκαν. Κριτήριο επιλογής τους ήταν κυρίως η καλή γνώση και χρήση αυτών καθώς και εύκολη υποστήριξη στην περίπτωση προβλημάτων.

Η C είναι μια γλώσσα υψηλού επιπέδου με αρκετά δυνατά χαρακτηριστικά που συναντώνται σε γλώσσες χαμηλού επιπέδου και στη γλώσσα μηχανής. Η ευελιξία αλλά και η έλλειψη αυστηρού ελέγχου δίνει μεγάλη ελευθερία στον έμπειρο προγραμματιστή αλλά μπορεί να αποτελέσει και παγίδα σε κάποιες περιπτώσεις δημιουργώντας προβλήματα αν δεν χρησιμοποιηθεί με σύνεση. Διαθέτει μεγάλη ποικιλία βιβλιοθηκών οι οποίες περιέχουν συναρτήσεις για κάθε χρήση, από συναρτήσεις χειρισμού αλφαριθμητικών και μαθηματικών τύπων μέχρι συναρτήσεις για διαχείριση αρχείων και δικτύων. Η C είναι πάντα σε μεταγλωττιζόμενη μορφή. Κατά τη διάρκεια της μεταγλώττισης ο πηγαίος κώδικας του προγράμματος ελέγχεται για συντακτικά λάθη και αν δεν βρεθεί κάποιο τότε παράγεται το εκτελέσιμο αρχείο. Στην δική μας περίπτωση ο μεταγλωττιστής που χρησιμοποιείται είναι ο gcc (**gcc version 4.8.2**). Είναι σημαντικό να σημειωθεί εδώ ότι η βιβλιοθήκη που θα αναφερθεί παρακάτω χρησιμοποιείται κατά τη μεταγλώττιση και την παραγωγή του εκτελέσιμου προγράμματος. Η πλειοψηφία των προγραμμάτων που χρησιμοποιήθηκαν για την επιλογή και επεξεργασία των καρτέ είναι γραμμένα σε C.

Όσον αφορά την C++, ξεκίνησε σαν μια επέκταση της C προσθέτοντας της αντικειμενοστραφή χαρακτηριστικά. Οι αντικειμενοστραφείς γλώσσες δίνουν μεγαλύτερη έμφαση στα δεδομένα από ότι στον κώδικα. Βασική έννοια εδώ είναι τα αντικείμενα πάνω στα οποία ουσιαστικά χτίζεται το πρόγραμμα και στις διαδικασίες (μεθόδους) που εφαρμόζονται επάνω τους. Όμως η βάση της γλώσσας παραμένει ίδια. Χαρακτηριστικό παράδειγμα είναι ότι οποιοδήποτε πρόγραμμα σε C μπορεί να δημιουργηθεί από έναν μεταγλωττιστή της C++ (**gcc version 4.8.2**). Χρησιμοποιήθηκε μόνο στην περίπτωση των wavelets μιας και εκεί βρέθηκε σχετική υποστήριξη για την υλοποίηση του προγράμματος. [Χατζηγιαννάκης 2010]

Τέλος χρησιμοποιήθηκαν ευρέως διάφορα **σενάρια κελύφους** (*shell scripts*) που περιείχαν εντολές οι οποίες ουσιαστικά προσέφεραν ομαδικές εκτελέσεις των προγραμμάτων λαμβάνοντας ως ορίσματα πολλά βίντεο και επεξεργάζοντας τα ένα τη φορά. Αυτά τα scripts εκτελούνταν μέσω

της γραμμής εντολών (*bash*) του λειτουργικού συστήματος εμφανίζοντας και την πορεία της εκτέλεσης τους. Το **bash** είναι ένα κέλυφος (*shell*) το οποίο είναι μια διεπαφή χρήστη που δέχεται εντολές και τις μεταφέρει προς εκτέλεση στο λειτουργικό σύστημα. Είναι κάτι παρόμοιο με τη γραμμή εντολών των Windows ή το λειτουργικό σύστημα MS-DOS. Η εκτέλεση των εντολών είναι άμεση μόλις πατηθεί το πλήκτρο Enter (τις περισσότερες φορές εκτός και αν απαιτούνται δικαιώματα διαχειριστή (*root*)) ως εκ τούτου απαιτείται η δέουσα προσοχή έτσι ώστε να μην προκαλέσουμε άθελα μας ζημιά στο σύστημα. [Ζεϊναλιπούρ 2014][LinuxCommand.org]



Περνώντας τώρα στην ουσία ήρθε η ώρα να αναφερθούμε στο βασικό εργαλείο χάρη στο οποίο έγινε η διαχείριση όλων των βίντεο που χρησιμοποιήθηκαν. Το **OpenCV** (*Open source Computer Vision*) είναι μια βιβλιοθήκη με συναρτήσεις για εφαρμογές εικόνας και βίντεο πραγματικού χρόνου. Διαθέτει διεπαφές για C, C++, Python αλλά και Java και υποστηρίζει πλήθος λειτουργικών συστημάτων όπως Windows, Linux, Mac OS, iOS και Android. Η βιβλιοθήκη έχει σχεδιαστεί έτσι ώστε να εκμεταλλεύεται πλήρως την υπολογιστική ισχύ του υλικού (*hardware*) πάνω στο οποίο εκτελείται καθώς υποστηρίζει τον πολυνηματισμό (*multi-threading*) χρησιμοποιώντας όλους τους διαθέσιμους επεξεργαστές κατά τη διάρκεια επεξεργασίας ενός βίντεο (*multi-core processing*).

Το OpenCV έχει αρθρωτή μορφή, που σημαίνει ότι το πακέτο περιλαμβάνει διάφορες διαμοιραζόμενες αλλά και στατικές βιβλιοθήκες. Τα ακόλουθα βασικά τμήματα είναι διαθέσιμα:

core - Περιέχει τις βασικές δομές που χρησιμοποιούνται, όπως είναι η *IplImage*, *Mat*, από τα υπόλοιπα τμήματα της βιβλιοθήκης.

imgproc - Τμήμα που αφορά την ψηφιακή επεξεργασία εικόνας που περιλαμβάνει γραμμικούς και μη γραμμικούς αλγορίθμους επεξεργασίας, γεωμετρικούς μετασχηματισμούς (αλλαγή μεγέθους εικόνας), μετατροπή από έναν χρωματισμό χώρο σε έναν άλλο (*HSV*, *YUV*, *YCbCr*, *i1i2i3*), υπολογισμός ιστογραμμάτων κλπ.

video - Τμήμα αφιερωμένο στην επεξεργασία βίντεο που περιλαμβάνει αλγορίθμους για υπολογισμό κίνησης, ανίχνευση παρασκηνίου και αντικειμένων.

calib3d - Αλγόριθμοι που χρησιμοποιούνται σε 3D περιεχόμενο όπως σε calibration 3D και

απλών καμερών, διάφορες μετρήσεις με βάση το αντικείμενο που παρακολουθείται κλπ.

features2d - Σημαντικοί αλγόριθμοι εντοπισμού χαρακτηριστικών (*Haar classifiers*, *SURF*).

objdetect - Τμήμα που ειδικεύεται στον εντοπισμό αντικειμένων χρησιμοποιώντας έτοιμες βάσεις δεδομένων που περιλαμβάνονται στη βιβλιοθήκη παραδείγματος χάριν πρόσωπα, μάτια, ανθρώπους, αυτοκίνητα και ούτω καθεξής.

highgui - Ένα βασικό κομμάτι της βιβλιοθήκης που υλοποιεί ένα περιβάλλον εύκολο στη χρήση για καταγραφή βίντεο, εικόνας και διάφορων κωδικοποιητών βίντεο καθώς επίσης ένα απλό γραφικό περιβάλλον περιορισμένων δυνατοτήτων.

gpu - Τμήμα που ειδικεύεται σε προγράμματα που εκμεταλλεύονται τις δυνατότητες και την υπολογιστική ισχύ της κάρτας γραφικών (*GPU*). Υποστηρίζονται μόνο κάρτες γραφικών της NVIDIA που χρησιμοποιούν την τεχνολογία CUDA.

Πρόγονος του OpenCV είναι η **Intel Integrated Performance Primitives** (*Intel IPP*), μια βιβλιοθήκη για επεξεργασία πολυμέσων και δεδομένων εφαρμογών που δημιουργήθηκε από την Intel. Αν βρεθεί στο σύστημα μας τότε το OpenCV την εκμεταλλεύεται έτσι ώστε να πετύχει πιο γρήγορα αποτελέσματα. [OpenCV][Bradski et Kaehler 2008]



Το **GStreamer** είναι μια βιβλιοθήκη για δημιουργία γραφικών και περιέχει στοιχεία για το χειρισμό πολυμέσων. Υποστηρίζει εφαρμογές επεξεργασίας ήχου και βίντεο αλλά και μεταφορά αυτών μέσω δικτύου (*audio/video streaming*). Εκμεταλλεύονται τις αναβαθμίσεις στους κωδικοποιητές και στα φίλτρα αθόρυβα. Αυτό γίνεται χάρη στην εύκολη αναβάθμιση αυτών με τη προσθήκη μιας απλής επέκτασης. Στην εργασία χρησιμοποιείται μόνο για την εγγραφή του βίντεο αφού γίνει η επιλογή των καρτέ και είναι υπεύθυνο για τους κωδικοποιητές (*encoders*) που χρησιμοποιούνται για την εγγραφή. [Gstreamer]



Τέλος, θα αναφερθούμε σε ένα πρόγραμμα το οποίο χρησιμοποιήθηκε για τον έλεγχο της ποιότητας των αρχείων βίντεο που χρησιμοποιήθηκαν. Το **ffprobe** είναι μέρος της οικογένειας προγραμμάτων **ffmpeg** και αποτελεί έναν αναλυτή ροής ήχου και βίντεο (*πολυμέσων*). Βασίζεται στις βιβλιοθήκη **libav** που είναι υπεύθυνη για τη διαχείριση των δεδομένων που προέρχονται από το ffprobe. [FFmpeg]

3.1.1 Επεξεργασία Καρέ

Πρώτο βήμα πριν ξεκινήσουμε να μπαίνουμε στην ουσία της εργασίας μας θα αναφερθούμε σε κάποια βασικές επεξεργασίες που μπορούν να γίνουν σε ένα βίντεο. Βασικά, αυτό είναι το πρώτο βήμα που γίνεται κυρίως για να μπούμε στο κλίμα της ψηφιακής επεξεργασίας βίντεο παρατηρώντας τα αποτελέσματα που θα έχουν οι αλγόριθμοι που εφαρμόσαμε είτε αυτοί είναι χρήσιμοι για τη συνέχεια είτε όχι.

Επιπρόσθετα, μια προ επεξεργασία πριν την επιλογή (μαρκάρισμα) των καρέ είναι πιθανόν να ωφελήσει ως προς την απόδοση του αλγορίθμου που θα επιλέξουμε. Κάποιοι από τους αλγόριθμους που χρησιμοποιήθηκαν βοηθούν στην βελτίωση του βίντεο ενώ άλλοι για την ανακάλυψη κάποιων χαρακτηριστικών τα οποία δεν είναι ορατά με κάποια άλλη επεξεργασία.

3.1.1.1 Αλγόριθμοι

Στην συγκεκριμένη παράγραφο θα γίνει εκτενής αναφορά σχετικά με τους αλγορίθμους που χρησιμοποιήθηκαν. Όλοι τους είναι απλοί αλγόριθμοι που εφαρμόστηκαν με επιτυχία σε όλα τα βίντεο που χρησιμοποιήθηκαν.

Οι αλγόριθμοι που χρησιμοποιήθηκαν περιγράφονται παρακάτω.

1. Η αλλαγή της **φωτεινότητας** (*brightness*) και της **αντίθεσης** (*contrast*) του βίντεο είναι σχετικά απλό να γίνει και μπορεί να οδηγήσει σε μια εύκολη βελτίωση της ποιότητας του βίντεο και να αναδείξει χαρακτηριστικά που πριν δεν ήταν ορατά. Η υλοποίηση του αλγορίθμου ήταν απλή και έχει ως εξής:

$$y[i][j] = (x[i][j]+a)*b$$

όπου y το νέο καρέ που διαμορφώνεται μετά την επεξεργασία, x είναι το τρέχων καρέ και οι μεταβλητές i,j είναι δείκτες που αντικατοπτρίζουν το pixel που επεξεργαζόμαστε. Όσων αφορά τα a,b , είναι μεταβλητές για τη φωτεινότητα και την αντίθεση με εύρος τιμών $[-\infty, +\infty]$ αντίστοιχα. Στην περίπτωση της αντίθεσης το βήμα αλλαγής των τιμών είναι 0.1. Πρέπει να σημειωθεί εδώ ότι ο παραπάνω τύπος εφαρμόζεται και για τα 3 κανάλια χρώματος RGB του κάθε καρέ.

2. Ένας τρόπος για ανάδειξη περιοχών που δεν είναι τόσο ορατές πριν είναι με την τεχνική της **εξισορρόπησης ιστογράμματος** (*histogram equalization*). Το ιστόγραμμα είναι ουσιαστικά μια γραφική αναπαράσταση της έντασης των τιμών των pixels σε μια συγκεκριμένη απόχρωση. Με εξισορρόπηση ιστογράμματος σκοπός μας είναι ο διασκορπισμός των τιμών σε όλο το φάσμα των αποχρώσεων αυξάνοντας την αντίθεση. Είναι μια γρήγορη βελτίωση που μπορεί να εφαρμοστεί αν και χρειάζεται προσοχή μιας και το αποτέλεσμα σε συγκεκριμένες περιπτώσεις μπορεί να μην είναι το επιθυμητό (ειδικά στις έγχρωμες εικόνες). Συνοπτικά, τα βήματα που ακολουθούνται είναι τα εξής:

α. Υπολογισμός ιστογράμματος και για τα 3 κανάλια χρωμάτων RGB

β. Υπολογισμός της πιθανότητας εμφάνισης κάθε απόχρωσης και για τα 3 κανάλια χρωμάτων RGB. Από το εύρος τιμών $[0,255]$ πάμε στο $[0,1]$

γ. Υπολογισμός της αθροιστικής πιθανότητας κάθε απόχρωσης και για τα 3 κανάλια χρωμάτων RGB

δ. Πολλαπλασιασμός της αθροιστικής πιθανότητας κάθε απόχρωσης και για τα 3 κανάλια χρωμάτων RGB με 255 έτσι ώστε να σχηματιστεί το τελικό αποτέλεσμα

3. Μια ακόμα εύκολη και εφαρμόσιμη μέθοδος για την αύξηση (μόνο) της φωτεινότητας είναι ο υπολογισμός της τετραγωνικής ρίζας του κάθε pixel του καρέ. Πρέπει να σημειωθεί ότι πριν την εφαρμογή της τετραγωνικής ρίζας πρέπει το κάθε pixel να πολλαπλασιαστεί με 255 έτσι ώστε να έχουμε ορατό αποτέλεσμα. Ο τύπος φαίνεται παρακάτω:

$$y[i][j] = \text{sqrt}(x[i][j]*255)$$

όπου y το νέο καρέ που διαμορφώνεται μετά την επεξεργασία, x είναι το τρέχων καρέ και οι μεταβλητές i,j είναι δείκτες που αντικατοπτρίζουν το pixel που επεξεργαζόμαστε. Ο παραπάνω τρόπος είναι αρκετά χρήσιμος σε βίντεο που είναι σκοτεινά.

4. Η ανίχνευση ακμών, ή πιο κατανοητά ο διαχωρισμός περιοχών σε κάθε κάρε, σε ένα βίντεο είναι πολύ χρήσιμη όταν θέλουμε να ξεχωρίσουμε κάποιες φιγούρες και να της παρατηρήσουμε καλύτερα. Η λογική του αλγορίθμου είναι να διατηρήσει μόνο τις αλλαγές από το κάθε καρέ δηλαδή να μας δώσει ανοιχτές αποχρώσεις σε σημεία όπου έχουμε έντονη αλλαγή των αποχρώσεων, όπως είναι τα σύνορα των περιοχών μέσα στο εκάστοτε καρέ και σκουρόχρωμες αποχρώσεις στις ομοιόμορφες περιοχές. Στην συγκεκριμένη εργασία ο αλγόριθμος ανίχνευσης ακμών που χρησιμοποιήθηκε είναι αυτός του Sobel. Αρχικά δηλώνουμε 2 πίνακες 3x3 (οι διαστάσεις του πίνακα πρέπει να είναι μονός αριθμός), 1 για κάθε άξονα, όπου τους γεμίζουμε με προκαθορισμένες τιμές.

$$\begin{array}{cc} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \\ \mathbf{wx} = & \mathbf{wy} = \end{array}$$

Χρησιμοποιώντας αυτούς τους 2 πίνακες κάνουμε συνέλιξη του κάθε πίνακα με το κάθε καρέ (με το κάθε κομμάτι κάθε φορά του καρέ μιας και υπολογίζουμε μια ομάδα από pixel κάθε φορά) και στη συνέχεια εφαρμόζω τον παρακάτω τύπο στο κάθε αποτέλεσμα που προκύπτει από τις συνέλιξεις του κάθε άξονα:

$$t = \text{sqrt}(tx*tx + ty*ty)$$

όπου tx, ty είναι τα αποτελέσματα της συνέλιξης για τον άξονα X και Y αντίστοιχα.

5. Τέλος, ένα αλγόριθμος που χρησιμοποιήθηκε κυρίως για βοηθητικούς λόγους είναι αυτός της εγγραφής κειμένου και γενικότερα αλφαριθμητικών χαρακτήρων πάνω στη ροή του βίντεο. Αυτό είναι ιδιαίτερα χρήσιμο, για παράδειγμα στην περίπτωση που θέλουμε να επαληθεύσουμε αν τα καρέ που επιλέχθηκαν από έναν αλγόριθμο είναι τα σωστά, με την έννοια της αλλαγής σκηνής στο συγκεκριμένο καρέ ή γενικότερα της διαφοροποίησης του συγκεκριμένου καρέ από το προηγούμενο του. Ουσιαστικά είναι ένα σετ από συναρτήσεις οι οποίες υπάρχουν ήδη υλοποιημένες στην βιβλιοθήκη που χρησιμοποιήθηκε. Στη συνέχεια θα μπούμε σε λεπτομέρειες ως προς την χρήση αυτών των συναρτήσεων.

`CvScalar black = CV_RGB(0,0,0); // Καθορισμός χρώματος για το κείμενο`

```

CvPoint pt1 = cvPoint(5,10);      // Καθορισμός του σημείου που θα βρίσκεται το κείμενο
int thickness = 0.0;              // Πάχος των χαρακτήρων

// Text variables
char text[20]                    // Το κείμενο
double hscale = 0.4;             // Οριζόντια κλίμακα
double vscale = 0.4;             // Κάθετη κλίμακα
double shear = 0.0;              // Καθορίζει αν θα είναι πλάγιοι οι χαρακτήρες
int line_type = 0;               // Τύπος γραμμής που καθορίζει τον τρόπο γραφής των χαρακτήρων
CvFont font;                     // Δομή που καθορίζει τη μορφοποίηση των γραμμμάτων

// Αρχικοποίηση της δομής για τη γραμματοσειρά
cvInitFont(&font1,CV_FONT_HERSHEY_SIMPLEX,hscale,vscale,shear,thickness,line_type);

// Τοποθέτηση του κειμένου στη καθορισμένη θέση σύμφωνα με τις παραμέτρους που δώσαμε
cvPutText(copy_frame,text1,pt1,&font1,black);

```

[OpenCV Docs]

3.1.2 Μαρκάρισμα Καρέ

Και εδώ περνάμε στην ουσία της εργασίας όπου θα γίνει εκτενής αναφορά στους αλγορίθμους που χρησιμοποιήθηκαν για την επιλογή των καρέ.

Γενικότερα, η επιλογή των καρέ είναι μια καλή μέθοδος για να κατανοήσουμε στα γρήγορα το περιεχόμενο ενός βίντεο. Εφαρμόστηκαν αρκετοί αλγόριθμοι και διάφορα σενάρια για να μπορέσουμε να καταλήξουμε σε κάποια συμπεράσματα σχετικά με την αποδοτικότητα των αλγορίθμων ως προς το μαρκάρισμα των καρέ, το αν είναι δυνατό με αυτόν τον τρόπο να

κατανοήσουμε ένα βίντεο και να κρατήσουμε αυτό που πραγματικά είναι χρήσιμο σε εμάς αλλά και την εξέταση των αλγορίθμων για μελλοντική χρήση σε κάποια άλλη εφαρμογή ανάλογα με τα αποτελέσματα που μας δίνουν.

3.1.2.1 Αλγόριθμοι

Σε αυτή την παράγραφο, θα παρουσιαστούν όλοι οι αλγόριθμοι για την επιλογή των καρτέ που χρησιμοποιήθηκαν αλλά και των διάφορων σεναρίων για κάθε αλγόριθμο.

Οι αλγόριθμοι που χρησιμοποιήθηκαν περιγράφονται παρακάτω.

1. Πρώτα απ' όλα, εφαρμόσαμε έναν απλό αλγόριθμο και στην υλοποίηση αλλά και στη λογική. Υπολογίζουμε τις απόλυτες διαφορές μεταξύ προηγούμενου και επόμενου καρτέ. Με τον όρο απόλυτες διαφορές εννοούμε την αφαίρεση των τιμών των εκάστοτε pixel του προηγούμενου καρτέ με το επόμενο. Με αυτό τον τρόπο, μπορούμε να υπολογίσουμε αν έχει υπάρξει αλλαγή στο συγκεκριμένο σημείο. Μετά την αφαίρεση το αποτέλεσμα που προκύπτει το περνάμε σε ένα νέο πίνακα που είναι ας πούμε το νέο καρτέ και αθροίζουμε όλα τα pixel αυτού του πίνακα. Σε αυτό το σημείο ήρθε η ώρα να επιλέξουμε το κριτήριο της επιλογής των καρτέ. Εδώ χρησιμοποιούμε την **ομοιότητα** (*similarity*) μεταξύ των καρτέ η οποία προκύπτει από τον ακόλουθο τύπο:

$$\text{similarity} = 100 - (\text{sum} / ((\text{double})(\text{width_img} * \text{height_img})) / 2.55)$$

όπου **similarity** είναι ένα ποσοστό ομοιότητας μεταξύ του προηγούμενου και του επόμενου καρτέ, **sum** είναι το άθροισμα των pixel μετά τον υπολογισμό των απόλυτων διαφορών και τα **width_img, height_img** είναι οι διαστάσεις του καρτέ. Οι σταθερές που έχουν μπει είναι για να βγάλουμε το ποσοστό σε κλίμακα $[0, 100]$.

Αυτός ο τύπος εφαρμόζεται και στα 3 κανάλια RGB.

Τέλος, έχοντας υπολογίσει το similarity το μόνο που απομένει είναι να επιλέξουμε ένα κατώφλι για την επιλογή των καρτέ. Εδώ εφαρμόστηκαν 2 σεναρία, ένα όταν το similarity είναι μικρότερο του 90% και ένα όταν το similarity είναι μικρότερο του 95%. Τα συγκεκριμένα ποσοστά φαίνονται μεγάλα αλλά ύστερα από δοκιμές καταλήξαμε ότι είναι τα πιο ρεαλιστικά από την άποψη του

συνόλου των καρέ που κρατάμε. Μικρότερα του 90% ποσοστά δεν μας κρατάνε ουσιαστικά κανένα καρέ τις περισσότερες φορές. [OpenCV Mailing List]

2. Ο επόμενος αλγόριθμος στον οποίο θα αναφερθούμε είναι σχετικά καλύτερος σε σχέση με τον προηγούμενο όσον αφορά την απόδοση. Σε αντίθεση με τις διαφορές μεταξύ των pixels των καρέ, εδώ υπολογίζουμε τις διαφορές στα ιστογράμματα του προηγούμενου καρέ με το επόμενο. Η διαδικασία και εδώ είναι σχετικά απλή. Αρχικά ξεκινάμε με τον υπολογισμό του ιστογράμματος για το τωρινό και το προηγούμενο καρέ. Στη συνέχεια αθροίζουμε τα ιστογράμματα και τα αφαιρώντας τα παίρνουμε μια τιμή η οποία χρησιμοποιείται για τον υπολογισμό του similarity. Ο τύπος που εφαρμόζεται στον αλγόριθμο των απόλυτων διαφορών μεταξύ των pixel χρησιμοποιείται και εδώ. Υπολογίζουμε το ιστόγραμμα και για τα 3 κανάλια RGB.

Τα σενάρια που εφαρμόζονται και εδώ είναι ίδια με παραπάνω έχοντας 2 κατώφλια, ένα 90 % και ένα 95%. Και εδώ συναντάμε κάποιες φορές παρόμοια ποσοστά με τον προηγούμενο αλγόριθμο παρόλα αυτά η αναγνώριση περιοχών δεν είναι πάντα η επιθυμητή. Για παράδειγμα, όταν έχουμε ένα καρέ που αναπαριστά μια παραλία και ένα άλλο όπου αναπαριστά ένα χωράφι με σιτάρι και γαλάζιο ουρανό και σχηματίζουμε τα ιστογράμματα τους τότε θα παρατηρήσουμε ότι είναι σχεδόν όμοια. Σε κάθε περίπτωση μια τέτοια κατάσταση μπορεί να είναι θετική ή αρνητική ανάλογα με το τι ψάχνουμε. [Rathod et al 2013]

3. Ένας ακόμα αλγόριθμος που ακολουθεί τη παραπάνω μεθοδολογία των διαφορών είναι και αυτός που θα αναλυθεί παρακάτω. Όπως ήδη γνωρίζουμε, ένα βίντεο είναι κωδικοποιημένο και συμπίεσμένο σε μια συγκεκριμένη μορφή και με ένα συγκεκριμένο τρόπο. Σε αυτό τον αλγόριθμο ασχολούμαστε ξεχωριστά με κάθε καρέ του βίντεο και την κωδικοποίηση αυτών και όχι τόσο της ροής του βίντεο. Πιο συγκεκριμένα, κωδικοποιούμε το κάθε καρέ (το οποίο είναι μια εικόνα) σε μορφή JPEG ή PNG και στη συνέχεια υπολογίζουμε τις διαφορές σε Bytes μεταξύ προηγούμενου και επόμενου καρέ. Όση περισσότερη λεπτομέρεια περιέχεται στο καρέ τόσο μεγαλύτερο το μέγεθος του. Αν η διαφορά του επόμενου από το προηγούμενο είναι σχετικά μεγάλη τότε βγαίνει το συμπέρασμα ότι υπάρχει κάποια αλλαγή στο συγκεκριμένο σημείο. Η κωδικοποίηση του καρέ σε JPEG ή PNG γίνεται με την παρακάτω συνάρτηση που μας προσφέρει η βιβλιοθήκη που χρησιμοποιήσαμε:

```
encoded_frame = cvEncodeImage("frame.JPEG",decoded_frame,0)
```

όπου **encoded_frame** είναι μια δομή τύπου **CvMat** (ουσιαστικά μπορούμε να το φανταστούμε σαν ένα πίνακα) όπου περιέχει το καρέ κωδικοποιημένο σε μορφή JPEG. Τα 3 ορίσματα της συνάρτησης **cvEncodeImage** είναι τα εξής:

- Το πρώτο είναι ένα όνομα που δίνουμε στο καρέ (δεν έχει κάποια ιδιαίτερη σημασία. Το πιο σημαντικό εδώ είναι η κατάληξη του αρχείου που καθορίζει και τον τρόπο συμπίεσης του καρέ).

- Η μεταβλητή **decoded_frame** είναι μια δομή τύπου **IplImage** η οποία περιέχει το καρέ όπως εξάχθηκε από τη ροή του βίντεο.
- Η τελευταία παράμετρος καθορίζει την ποιότητα του καρέ (αν μιλάμε για συμπίεση *JPEG*) ή το επίπεδο συμπίεσης (αν μιλάμε για συμπίεση *PNG*). [Boreczky et Rowe 1996]

4. Σε αυτό το σημείο θα αναφερθούμε σε έναν διαφορετικό αλγόριθμο ο οποίος ακολουθεί μια διαφορετική φιλοσοφία όσων αφορά τη σύγκριση των καρέ. Το **Edge Change Ratio (ECR)** είναι ένας αλγόριθμος ο οποίος ουσιαστικά προσπαθεί να συγκρίνει το πραγματικό περιεχόμενο μεταξύ 2 καρέ. Είναι πολύπλοκος από την άποψη προεπεξεργασίας που επιδέχονται τα καρέ προτού συγκριθούν μεταξύ τους και έτσι τον καθιστά “βαρύ” (*processor intensive*) ως προς την υπολογιστική ισχύ. Είναι ιδιαίτερα ευαίσθητος στις έντονες αλλαγές (*hard cuts*) μεταξύ των καρέ αλλά κάποιες φορές αποτυγχάνει να εντοπίσει αλλαγές από κάποιο καρέ σε ένα άλλο που πραγματοποιούνται σταδιακά (*fade in, fade out, dissolve*). Αυτά είναι, θα μπορούσαμε να πούμε, είδη εναλλαγών των καρέ (ή καλύτερα σκηνών μιας και αποτελούνται από πολλά καρέ για να επιτευχθεί το αποτέλεσμα) στο βίντεο. Όταν έχουμε **fade in/out** τότε η σκηνή εμφανίζεται σταδιακά στην οθόνη ενώ στο **dissolve** έχουμε την σταδιακή εξαφάνιση μιας σκηνής όπου μέσα από αυτή εμφανίζεται σταδιακά η επόμενη. Βίντεο με πολλές εναλλαγές σκηνών είναι καλοί υποψήφιοι για αυτόν τον αλγόριθμο. Τα βήματα υλοποίησης είναι αρκετά και θα αναλυθούν παρακάτω:

α. Αρχικά μετατρέπουμε το καρέ από έγχρωμο σε ασπρόμαυρο (*αποχρώσεις στην κλίμακα του γκρι*)

β. Εφαρμόζουμε έναν αλγόριθμο ανίχνευσης ακμών π.χ. **Sobel, Canny** στο καρέ ο οποίος θα μας ξεχωρίσει τις περιοχές που υπάρχουν στο καρέ.

γ. Μετατρέπουμε αυτή τη φορά το ασπρόμαυρο καρέ σε δυαδική μορφής δηλαδή να αποτελείται μόνο από 2 αποχρώσεις, άσπρο και μαύρο. Αυτό γίνεται με τη χρήση κάποιου κατωφλιού το οποίο μπορεί να είναι προκαθορισμένο ή αυτόματο. Αν η απόχρωση ενός pixel είναι μεγαλύτερη του κατωφλιού τότε παίρνει την τιμή **255** (άσπρο) αλλιώς **0** (μαύρο).

δ. Μέτρηση του συνόλου των pixels σ^n, σ^{n-k} που αποτελούν τις ακμές των περιοχών του προηγούμενου και του επόμενου καρέ.

ε. Χρήση της μεθόδου της **διαστολής** (*dilation*) των ακμών έτσι ώστε να είναι πιο αποδοτική η σύγκριση μεταξύ προηγούμενου και επόμενου καρέ.

ζ. Αντιστροφή του καρέ, κάνοντάς το δηλαδή αρνητικό.

η. Σε αυτό το σημείο, κάνουμε μια πράξη **AND** μεταξύ του προηγούμενου καρέ που επεξεργαστήκαμε στο βήμα (γ) και του επόμενου καρέ το οποίο επεξεργαστήκαμε μέχρι το βήμα (ζ).

θ. Μέτρηση του συνόλου των pixels εξόδου $Xin^n, Xout^{n-k}$ από το προηγούμενο και το

επόμενο καρέ.

1. Διαίρεση των **pixels** εξόδου με τα **pixels** των ακμών για το προηγούμενο και το επόμενο καρέ αντίστοιχα. Τέλος γίνεται έλεγχος για την εύρεση της μεγαλύτερης τιμής εκ των 2 η οποία είναι και το ECR μεταξύ των 2 καρέ. [Nasreen et Shobha 2013][Miene et al 2004]

5. Τέλος θα αναφερθούμε σε μια διαφορετική προσέγγιση επιλογής των καρέ. Αυτό γίνεται με χρήση των wavelets και πιο συγκεκριμένα του **Haar wavelet**. Η χρήση των wavelets στη ψηφιακή επεξεργασία βίντεο είναι σχετικά πρόσφατη και έχει να προσφέρει πολλά όσον αφορά την απόδοσή και την ποιότητα. Στον συγκεκριμένο αλγόριθμο, χρησιμοποιούμε τις ενέργειες ως μέτρο επιλογής των καρέ. Αρχικά υπολογίζουμε τους συντελεστές για κάθε **επίπεδο αποσύνθεσης** (*decomposition level*) του καρέ (στην εργασία φτάνουμε μέχρι **επίπεδο 3** (*level 3*)). Στη συνέχεια αθροίζουμε τους συντελεστές κάθε επιπέδου και υψώνουμε στο τετράγωνο το άθροισμα τους. Έτσι στην περίπτωση μας έχουμε 9 τιμές για τις ενέργειες για 3 επίπεδα αποσύνθεσης. Επιπλέον τιμές που κρατάμε είναι η συνολική ενέργεια για όλο το καρέ η οποία χρησιμοποιείται για την επιλογή των καρέ. [Tint et Soe. 2013]

3.1.2.2 Παράμετροι – Σενάρια

Σε αυτή τη παράγραφο θα γίνει αναφορά στα σενάρια πάνω στα οποία εφαρμόσαμε τους αλγορίθμους επιλογής των καρέ αλλά και τις παραμέτρους που δώσαμε για να λειτουργήσουν απρόσκοπτα.

Αρχικά η πρώτη μας σκέψη ήταν να εφαρμόσουμε μια λογική τυχαίων περιοχών πάνω στα καρέ του βίντεο από τα οποία θα παίρναμε τις μετρήσεις για κάθε αλγόριθμο και στη συνέχεια θα επιλέγαμε τα καρέ με βάση τα αποτελέσματα που πήραμε. Μετά από σειρά πειραμάτων καταλήξαμε ότι τα αποτελέσματα δεν είναι συμπαγή με την έννοια ότι διαφορετικά παράθυρα θα μας δώσουν διαφορετικές μετρήσεις και αυτές με τη σειρά τους διαφορετικά μαρκαρισμένα καρέ. Η επιλογή των θέσεων των παραθύρων είναι τυχαία κάθε φορά οπότε είναι λογικό αυτό. Ένα βασικό μειονέκτημα της χρήσης των παραθύρων είναι το μέγεθος τους το οποίο εξαρτάται από την ανάλυση του βίντεο και πρέπει να υπάρχει μια αναλογία αυτών έτσι ώστε να έχουμε και λογικά αποτελέσματα. Με τη χρήση των παραθύρων έχουμε τεράστιο κέρδος σε υπολογιστική ισχύ μιας και η εφαρμογή του αλγορίθμου γίνεται σε ένα κομμάτι του καρέ και όχι σε όλο.

Αντίθετα από τις αρχικές μας προβλέψεις αποφασίσαμε να εφαρμόσουμε τους αλγορίθμους σε ολόκληρη την επιφάνεια του καρέ. Ουσιαστικά εδώ εκμεταλλευόμαστε όλα τα χαρακτηριστικά του καρέ και τα αποτελέσματα είναι ταυτόσημα κάθε φορά που τρέχουμε τον ίδιο αλγόριθμο με τις ίδιες παραμέτρους. Ένα μειονέκτημα εδώ είναι η ύπαρξη “άχρηστων” περιοχών που μπορεί να

υπάρχουν στα κάθε καρέ ενός βίντεο. Στην εργασία μας αυτό το πρόβλημα συναντάται σε διάφορα ιατρικά βίντεο όπου υπάρχει ένα μαύρο κομμάτι που καλύπτει μεγάλο μέρος του καρέ και περιέχει διάφορα στοιχεία για την εκάστοτε εξέταση και για το γιατρό που τη διεξάγει. Επιπρόσθετα καλό είναι να αναφερθούμε και στην ανάλυση των βίντεο που έχει σημαντική επίπτωση στο χρόνο επεξεργασίας και την υπολογιστική ισχύ που διαθέτουμε. Το μεγαλύτερο μέρος των αλγορίθμων περιλαμβάνει υπολογισμούς σε επίπεδο pixel των καρέ του βίντεο που σημαίνει ότι όσα περισσότερα τα pixel τόσοι περισσότεροι χρόνος απαιτείται για την ολοκλήρωση της επεξεργασίας του εκάστοτε βίντεο. Ας δώσουμε ένα παράδειγμα για να γίνει πιο κατανοητό: έστω ένα βίντεο ανάλυσης **1920x1080 pixels (Full HD)**. Αν θεωρήσουμε ότι είναι έγχρωμο τότε έχουμε $1920 * 1080 * 3$ (τα κανάλια RGB) = 6.220.800 pixels. Αρκετά μεγάλος αριθμός αν φανταστούμε το καρέ σαν έναν πίνακα με 6.000.000+ τιμές.

Παρακάτω θα αναφερθούμε στους αλγορίθμους που χρησιμοποιήσαμε για την επιλογή των καρέ ως προς τον τρόπο επιλογής των καρέ (*κατώφλια*) αλλά και όλα τα διαφορετικά σενάρια πάνω στα οποία εφαρμόστηκαν οι αλγόριθμοι για την παραγωγή περισσότερων αποτελεσμάτων δείχνοντας έτσι την δυναμική των αλγορίθμων χρησιμοποιώντας διαφορετικά κατώφλια και διαφορετικά βίντεο. Αυτή η ποικιλομορφία δίνει πιο ορθά και πιο αξιόπιστα αποτελέσματα.

Ο αλγόριθμος των απόλυτων διαφορών μεταξύ των pixel του προηγούμενου με το επόμενο καρέ θα είναι ο πρώτος που θα αναλυθεί. Ο συγκεκριμένος αλγόριθμος δεν έχει κάποια δυσκολία στην υλοποίησή του και η επιλογή των καρέ έγινε με βάση ένα απλό **προ-τοποθετημένο κατώφλι** (*fixed threshold*) κατώφλι, μια σταθερή τιμή η οποία προέκυψε από αρχικά πειράματα που έγιναν έτσι ώστε να κρίνουμε τη κλίμακα που κυμαίνονται οι τιμές μας. Η χρήση μιας σταθεράς για την επιλογή των καρέ είναι μια απλή λύση για την απόκτηση γρήγορων αποτελεσμάτων. Η επιλογή μιας και μεταβαλλόμενης τιμής είναι δύσκολο κριτήριο μιας και προϋποθέτει αρκετή προεπεξεργασία και ένα διαφορετικό υπόβαθρο πάνω στο οποίο βασιζόμαστε για την επιλογή των καρέ. Η κάθε μέθοδος έχει τα πλεονεκτήματα και τα μειονεκτήματα της. Στην δική μας περίπτωση κερδίζουμε σε απόδοση αλλά χάνουμε σε αξιοπιστία μιας και μας περιορίζει το δείγμα σε γενικές γραμμές. Η επιλογή της μεθόδου του υπολογισμού των τελικών τιμών έγινε με κριτήριο τη δημιουργία κάποιου ποσοστού έτσι ώστε το αποτέλεσμα της επεξεργασίας κάθε καρέ να είναι πιο εύκολα αντιληπτό. Τα κατώφλια που χρησιμοποιήθηκαν είναι το **90** και **95** (σαν ποσοστά).

Στη συνέχεια έχουμε τον αλγόριθμό διαφορών μεταξύ ιστογραμμάτων όπου ξεφεύγουμε από το επίπεδο της σύγκρισης των pixel ένα προς ένα και περνάμε στο επίπεδο των αποχρώσεων των καρέ. Στην συγκεκριμένη περίπτωση εφαρμόζουμε τα ίδια κριτήρια όπως και προηγουμένως χρησιμοποιώντας το ποσοστό της ομοιότητας των ιστογραμμάτων για την επιλογή των καρέ. Πρέπει να σημειωθεί εδώ ότι όπως και παραπάνω χρησιμοποιούμε την ίδια μέθοδο για την επιλογή των καρέ. Επίσης, στην εργασία έχει γίνει χρήση μόνο έγχρωμων βίντεο οπότε οι τιμές οι οποίες παράγονται από το πρόγραμμα αφορούν ξεχωριστά το κάθε κανάλι χρώματος (RGB). Με λίγα λόγια, για κάθε καρέ έχουμε και μια τιμή ομοιότητας για το κάθε κανάλι ξεχωριστά. Εδώ έγγυται το γεγονός ότι πρέπει να βρούμε κάποιο τρόπο για να γίνει η επιλογή των καρέ μιας και

έχουμε 3 τιμές να διαχειριστούμε. Εφαρμόζουμε ένα **λογικό Ή (OR)** μεταξύ των τριών τιμών των καναλιών που σημαίνει ότι αν έστω μια τιμή είναι χαμηλότερη από το κατώφλι μας τότε επιλέγεται το καρέ. Ουσιαστικά εδώ σημασία έχουν μόνο ένα από τα 3 κανάλια μιας και ένα από αυτά θα κρίνει την επιλογή. Αυτό είναι χρήσιμο ειδικά αν σε κάποια βίντεο κυριαρχεί μια συγκεκριμένη απόχρωση οπότε και θα παρατηρούνται οι έντονες αλλαγές. Τα κατώφλια που χρησιμοποιήθηκαν είναι το **90** και **95** (σαν ποσοστά).

Όπως προαναφέρθηκε, το κάθε καρέ είναι μια εικόνα αποτελούμενη από σημεία (*pixels*). Προγραμματιστικά μιλώντας, το κάθε σημείο ισοδυναμεί με 1 Byte. Συνεπώς αν έχουμε ένα έγχρωμο καρέ διαστάσεων 1920x1080 τότε το συνολικό του μέγεθος είναι: $1920 * 1080 * 3 = 6.220.800 \text{ Bytes} \approx 5,9 \text{ MBytes}$. Ο αλγόριθμος που θα αναφερθούμε ασχολείται με τις διαφορές στο μέγεθος σε Bytes μεταξύ προηγούμενου και επόμενου καρέ. Βέβαια επειδή στην πραγματικότητα δεν πρόκειται να δούμε μεγέθη εικόνων σαν το παραπάνω κωδικοποιούμε τα καρέ σε μορφή JPEG προτού προχωρήσουμε σε περαιτέρω επεξεργασία. Είναι το πιο λογικό καθώς η πλειοψηφία των βίντεο που υπάρχουν είναι κωδικοποιημένα σε κάποια μορφή και όχι ασυμπίεστα. Άρα το μέγεθος σε Bytes αυτή τη φορά είναι το κωδικοποιημένο καρέ. Το μέγεθος του καρέ είναι ένας παράγοντας που μπορεί να δείξει αν έχει υπάρξει αλλαγή περιεχομένου από το προηγούμενο στο επόμενο καρέ. Όταν το περιεχόμενο είναι παραπλήσιο, το μέγεθος δεν έχει μεγάλες μεταβολές. Αντίθετα όταν υπάρχουν εναλλαγές προφανώς το μέγεθος θα αλλάξει σε κάποιο βαθμό καθώς αλλάζουν οι τιμές των pixel. Σαν κριτήριο επιλογής εδώ έχουμε το ποσοστό του μεγέθους του καρέ σε Bytes. Αυτό σημαίνει ότι υπολογίζεται η διαφορά μεταξύ των καρέ και στη συνέχεια ελέγχεται αν είναι μικρότερη από αυτό το ποσοστό. Ο υπολογισμός του κατωφλιού γίνεται ως εξής: επιλογή του ποσοστού που θέλουμε να έχουμε και πολλαπλασιασμός αυτού με το μέγεθος του καρέ (σε Bytes). Αν η διαφορά σε Bytes είναι μεγαλύτερη από το κατώφλι τότε γίνεται επιλογή του καρέ. Οι παράμετροι που χρησιμοποιήθηκαν για τον υπολογισμό του κατωφλιού είναι το **0,5%**, **1%** και **5%**.

Περνάμε σε έναν διαφορετικό αλγόριθμο που ξεφεύγει από το επίπεδο των pixel και ακολουθεί μια διαφορετική προσέγγιση. Με τον **Edge Change Ratio (ECR)** αλγόριθμο ουσιαστικά ελέγχουμε αν ένα καρέ μοιάζει με ένα άλλο κρίνοντας από το περιεχόμενο τους. Αυτό γίνεται εντοπίζοντας τα σχήματα στο κάθε καρέ χρησιμοποιώντας και άλλους αλγορίθμους για αυτή τη διαδικασία. Ο συγκεκριμένος αλγόριθμος είναι ο πιο χρονοβόρος και ο πιο απαιτητικός από άποψη πόρων. Ένα βασικό μειονέκτημα του είναι ότι όσο μεγαλώνει η ανάλυση του βίντεο τόσο μεγαλώνει και ο χρόνος ολοκλήρωσης της επεξεργασίας. Αποτελείται από πολλά βήματα και το καρέ δέχεται πολλαπλές επεξεργασίες μέχρι να πάρουμε τις επιθυμητές τιμές. Μια παρατήρηση εδώ είναι ότι στα βίντεο υψηλής ανάλυσης κυρίως από 1920x1080 και μεγαλύτερα παρατηρούνται μεγάλες τιμές σε κάποια καρέ. Αυτό συμβαίνει γιατί ο αλγόριθμος κρίνει ότι στο συγκεκριμένο σημείο υπάρχει μεγάλη αλλαγή μεταξύ προηγούμενου και επόμενου καρέ. Σαν παράμετρο επιλογής των καρέ, κρατάμε τη μέγιστη τιμή ECR που προκύπτει από όλο το βίντεο. Η διαφορά εδώ είναι ότι για να μην μας κρατήσει μεγάλες τιμές βάζουμε μια σταθερή τιμή, το 10 για παράδειγμα. Αυτή προέκυψε από μετρήσεις που πάρθηκαν από όλα τα βίντεο και παρατηρώντας τι τιμές παράγονται.

Για να διατηρήσουμε την ομοιομορφία του αλγόριθμου έτσι ώστε να λειτουργεί καλύτερα και για τις μεγάλες αναλύσεις, πριν εφαρμόσουμε το κατώφλι ελέγχουμε αν η μέγιστη τιμή ECR είναι μεγαλύτερη από τη σταθερά που έχουμε θέσει. Στο συγκεκριμένο αλγόριθμο εφαρμόζουμε 2 σενάρια για την επιλογή των καρέ. Αρχικά, πρώτο κατώφλι είναι αν το ECR του καρέ είναι μεγαλύτερο από το $[0.0 (0\%) , 1.0 (100\%)]$ επί της μέγιστης τιμής ECR ολόκληρου του βίντεο. Χρησιμοποιούμε 3 τιμές: **30%**, **50%** και **90%** οι οποίες πολλαπλασιάζονται με το μέγιστο ECR. Σαν δεύτερο σενάριο υπολογίζουμε τη διαφορά των ECR τιμών μεταξύ προηγούμενου και επόμενου καρέ. Σαν κατώφλια χρησιμοποιούμε τις τιμές **0,5** , **2** και **4**. Αν το η διαφορά είναι μεγαλύτερη από αυτές τιμές τότε επιλέγεται το καρέ. Με τον τελευταίο τρόπο δεν αντιμετωπίζουμε κάποιο πρόβλημα σχετικά με τη διαφορά των αναλύσεων αλλά ούτε μας επηρεάζει κάποιος άλλος παράγοντας.

Τελευταίος αλγόριθμος που θα αναφερθούμε είναι αυτός των wavelets. Γίνετε χρήση του **Haar wavelet** το οποίο είναι σχετικά απλό στην εφαρμογή. Και εδώ ακολουθείται μια διαφορετική φιλοσοφία ως προς την ανίχνευση αλλαγών. Υπολογίζοντας τις ενέργειες για κάθε επίπεδο αποσύνθεσης τις χρησιμοποιούμε στη συνέχεια για την επιλογή των καρέ. Ουσιαστικά υπολογίζουμε το άθροισμα των ενεργειών για το κάθε καρέ και παράλληλα κρατάμε τη μέγιστη τιμή που προκύπτει από όλο το βίντεο. Με αυτή τελικά διαμορφώνουμε το κατώφλι μας. Χρησιμοποιούμε την ίδια λογική με παραπάνω όπου πολλαπλασιάζουμε με ένα ποσοστό τη μέγιστη τιμή που προκύπτει. Στο συγκεκριμένο αλγόριθμο πολλαπλασιάζουμε με **30%**, **50%** και **90%**.

3.1.3 Μετάδοση Καρέ

Ήρθε η ώρα να περάσουμε στο δεύτερο μέρος αυτής της εργασίας όπου έχει ολοκληρωθεί η επεξεργασία του βίντεο και η επιλογή των καρέ βάση του αλγορίθμου που επιλέχθηκε. Το παραγόμενο βίντεο είναι ουσιαστικά μια σύνοψη του αρχικού με λιγότερα καρέ και έχει διάρκεια όση και το αρχικό βίντεο. Αυτό σημαίνει ότι όταν δεν υπάρχουν επιλεγμένα καρέ σε κάποιο σημείο του βίντεο τότε επαναλαμβάνεται το τελευταίο μαρκαρισμένο μέχρι να υπάρξει αλλαγή. Οπτικά το αποτέλεσμα είναι αρκετά καλό δίνοντας άμεσα μια πρώτη γεύση για την κατανόηση του. Όσον αφορά την μετάδοση του βίντεο, έχουμε ακολουθήσει μια διαφορετική λογική που εξοικονομεί σε αρκετά μεγάλο βαθμό το διαθέσιμο εύρος ζώνης για όσο χρονικό διάστημα διαρκεί η μετάδοση. Στην περίπτωση μας δεν μεταδίδουμε ολόκληρο το βίντεο αλλά μόνο τα επιλεγμένα καρέ αυτού. Συνεπώς ο φόρτος του δικτύου είναι χαμηλός μιας και στις περισσότερες περιπτώσεις μεταδίδεται λιγότερο από το 50% των καρέ (πάντα με γνώμονα τον αλγόριθμο που χρησιμοποιήθηκε).

3.1.3.1 Πρόγραμμα Εξυπηρετητή (Server)

Μπαίνοντας σε ποιο τεχνικά θέματα θα αναφερθούμε ξεχωριστά για τα προγράμματα που χρησιμοποιήθηκαν για τη μετάδοση των καρτέ αλλά και τις μεθόδους που χρησιμοποιήθηκαν.

Για να λειτουργήσει η μέθοδος που χρησιμοποιήσαμε απαιτούνται 2 προγράμματα που αντιστοιχούν στον πελάτη και τον εξυπηρετητή αντίστοιχα. Το ένα στέλνει δεδομένα και το άλλο λαμβάνει τα δεδομένα και τα αναπαράγει με προκαθορισμένο τρόπο. Στη συγκεκριμένη παράγραφο θα ασχοληθούμε με το πρόγραμμα του εξυπηρετητή το οποίο είναι υπεύθυνο για την μετάδοση του επεξεργασμένου βίντεο. Το πρώτο πράγμα που κάνουμε είναι να περάσουμε στο πρόγραμμα μας την πληροφορία σχετικά με το πόσα και ποια είναι τα επιλεγμένα καρτέ από τα οποία αποτελείται και το τελικό βίντεο. Καθώς διαβάζουμε το αρχείο βίντεο κατευθείαν από το δίσκο, χρησιμοποιώντας τις δομές που μας παρέχει η βιβλιοθήκη παίρνουμε κάποια δεδομένα από το αρχείο τα οποία θα χρειαστούμε στη συνέχεια όπως είναι το **FPS**, οι **διαστάσεις των καρτέ** και το **σύνολο των καρτέ του βίντεο**. Αυτά τα περνάμε κατευθείαν σε ένα πίνακα ο οποίος θα μεταδοθεί στον πελάτη για να μπορέσει να αρχικοποιήσει κάποιες τιμές βασικές για την ανακατασκευή του βίντεο στη μεριά του. Επόμενο βήμα είναι η αρχικοποίηση κάποιων μεταβλητών σχετικά με τον τρόπο μετάδοσης. Κάνοντας χρήση της συνάρτησης **socket()** δημιουργούμε μια **υποδοχή (socket)** μέσω της οποίας διεξάγεται η επικοινωνία μεταξύ πελάτη και εξυπηρετητή. Αυτό αφορά το πρωτόκολλο που θα χρησιμοποιηθεί αλλά και σε ποια **θύρα (port)** θα “ακούει” για εισερχόμενες αιτήσεις ο εξυπηρετητής. Στην περίπτωση μας χρησιμοποιήθηκε το πρωτόκολλο TCP για λόγους αξιοπιστίας μιας και δεν στέλνουμε ολόκληρο το βίντεο αλλά κομμάτια αυτού υπό τη μορφή των καρτέ. Η θύρα που δίνουμε μπορεί να είναι κάθε φορά διαφορετική αλλά πάντα μεγαλύτερη από 1024. Είναι σημαντικό να σημειωθεί εδώ ότι η συγκεκριμένη θύρα πρέπει να πρώτα να ανοιχτεί στο modem που χρησιμοποιούμε για να συνδεθούμε στο Internet, **προωθώντας την εξερχόμενη/εισερχόμενη κίνηση προς/από τον εξυπηρετητή σε αυτή τη θύρα (port forwarding)**. Έχοντας έτοιμες τις προϋποθέσεις για τη δημιουργία σύνδεσης, χρησιμοποιώντας τη συνάρτηση **bind()** αναθέτουμε τη θύρα και την τοπική διεύθυνση IP που δηλώσαμε παραπάνω και επίσημα στο πρόγραμμα μας έτσι ώστε να **μεταδίδει/λαμβάνει (transmit/receive)** μέσω αυτής. Μια ακόμα συνάρτηση είναι η **listen()** που τοποθετεί τον εξυπηρετητή σε παθητική κατάσταση και καθορίζει τον μέγιστο αριθμό συνδέσεων. Έχοντας ρυθμίσει όλες τις δικτυακές παραμέτρους είμαστε έτοιμοι να δεχθούμε αιτήσεις από απομακρυσμένους πελάτες. Η συνάρτηση **accept()** είναι αυτή η οποία καλείται για να διαχειριστεί τις αιτήσεις του πελάτη και είναι υπεύθυνη για τη δημιουργία της σύνδεσης. Μόλις εγκαθιδρυθεί επικοινωνία με ένα πελάτη το πρώτο βήμα που κάνει ο εξυπηρετητής είναι να στείλει τα δεδομένα του βίντεο μας έτσι ώστε να μπορέσει ο πελάτης να προετοιμάσει τις δομές πάνω στις οποίες θα αποθηκευτούν τα καρτέ που θα αποσταλούν. Η λογική που ακολουθούμε έχει ένα και μοναδικό στόχο: τη μείωση του φόρτου του δικτύου όσο διαρκεί η επικοινωνία. Αυτό το πετυχαίνουμε με τη μετάδοση των καρτέ που έχουν επιλεχθεί και όχι ολόκληρου του βίντεο. Η εγγραφή των δεδομένων στην υποδοχή (μπορούμε να το φανταστούμε σαν ένα χώρο μνήμης ή έναν πίνακα) γίνεται με τη συνάρτηση **write()**. Έχοντας σαν πληροφορία το

ποια καρτέ έχουν μαρκαριστεί, στέλνουμε μόνο αυτά. Έτσι η μετάδοση είναι σχετικά σύντομη μιας και δεν αποστέλλονται όλα τα καρτέ. Στην επόμενη παράγραφο θα αναφερθούμε στο πρόγραμμα του πελάτη.

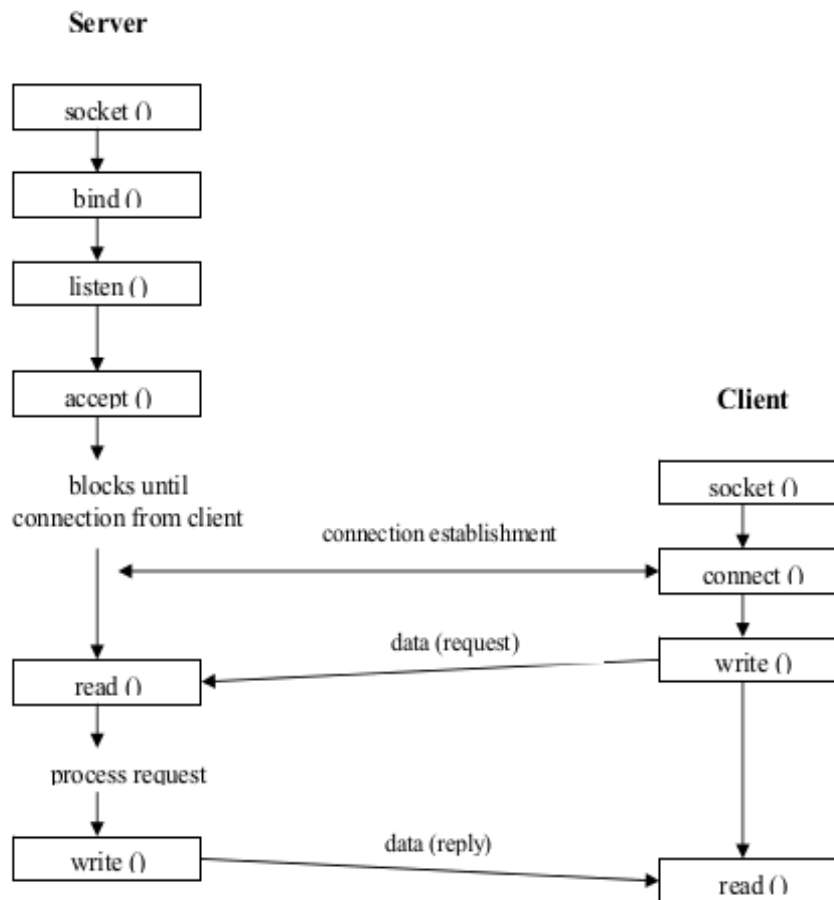
3.1.3.2 Πρόγραμμα Πελάτη (Client)

Από τη μεριά του πελάτη, ακολουθούμε παρόμοια λογική όσον αφορά το δικτυακό κομμάτι χρησιμοποιώντας τις παραμέτρους που λαμβάνουμε από τον εξυπηρετητή. Η διαφορά εδώ έγκειται στον τρόπο αναπαραγωγής του βίντεο.

Και εδώ το πρώτο πράγμα που δημιουργείται είναι μια **υποδοχή (socket)** με τη χρήση της **socket()** η οποία είναι ουσιαστικά η πύλη που θα δεχτεί δεδομένα από το κανάλι επικοινωνίας που θα έχει ανοιχτεί με τον εξυπηρετητή. Σε αυτό το σημείο παίρνουμε πληροφορίες σχετικά με τη διεύθυνση IP του εξυπηρετητή αλλά και την πόρτα στην οποία αναμένει αιτήσεις. Στην ίδια πόρτα πρέπει να αρχικοποιηθεί και ο πελάτης έτσι ώστε να είναι δυνατή η επικοινωνία. Το επόμενο βήμα είναι η δημιουργία της σύνδεσης χρησιμοποιώντας την **connect()** η οποία χρησιμοποιεί τη διεύθυνση του εξυπηρετητή καθώς και την θύρα επικοινωνίας για την έναρξη της λήψης δεδομένων. Τα δεδομένα που λαμβάνονται στην αρχή έχουν σχέση με την δομή του αρχείου βίντεο που θα λάβουμε και είναι απαραίτητα για την αρχικοποίηση κάποιων μεταβλητών που είναι υπεύθυνες για την ανακατασκευή του βίντεο. Η λήψη δεδομένων στη μεριά του πελάτη γίνεται με τη συνάρτηση **read()** η οποία διαβάζει τα δεδομένα που λαμβάνονται στην υποδοχή και τα αποθηκεύει όπου έχουμε καθορίσει εμείς. Στην προκειμένη περίπτωση, τις αρχικές τιμές τις αποθηκεύουμε σε έναν πίνακα και στη συνέχεια σε άλλες μεταβλητές για την καλύτερη διαχείριση τους ενώ τα δεδομένα των καρτέ αποθηκεύονται απευθείας στις δομές δεδομένων που μας διαθέτει η βιβλιοθήκη. Μόλις πραγματοποιηθεί η αποθήκευση των δεδομένων των καρτέ κατευθείαν διεξάγεται η αναπαραγωγή τους. Με λίγα λόγια αμέσως μετά τη λήψη ο πελάτης βλέπει άμεσα τα επιλεγμένα καρτέ που διαδέχονται το ένα το άλλο μόνο όταν υπάρχει αλλαγή. Εδώ είναι και το σημείο ουσιαστικά όπου έχουμε επέμβει σε μεγάλο βαθμό στον τρόπο αναπαραγωγής του βίντεο καθώς το τελικό βίντεο το οποίο παρακολουθεί ο πελάτης είναι μόνο τα επιλεγμένα καρτέ στην διάρκεια του αρχικού βίντεο. Αυτό το πετυχαίνουμε αποστέλλοντας τα καρτέ που έχουν μαρκαριστεί με τον εκάστοτε αλγόριθμο και όσο δεν έχουμε νέα καρτέ προς λήψη, αυτό το καρτέ επαναλαμβάνεται μέχρι να υπάρξει αλλαγή. Αυτή η διαδικασία συνεχίζεται μέχρι να συμπληρώσουμε τον συνολικό αριθμό των καρτέ του αρχικού βίντεο. Έτσι με αυτή τη τεχνική δεν έχουμε φόρτο στο δίκτυο όσο δεν υπάρχουν νέα καρτέ προς λήψη αυξάνοντας την απόδοση και τη λειτουργικότητα των προγραμμάτων.

Παρακάτω φαίνεται μια σχηματική αναπαράσταση του μοντέλου **πελάτη-εξυπηρετητή** για **επικοινωνία με σύνδεση (connection-oriented communication)** όπως χρησιμοποιεί το πρωτόκολλο

TCP καθώς και των σχέσεων μεταξύ **πελάτη** (*client*) και **εξυπηρετητή** (*server*).



3.2 Υλικό (βίντεο)

Σε αυτή τη παράγραφο θα αναφερθούμε στα βίντεο που χρησιμοποιήθηκαν στα πλαίσια της εργασίας. Η πλειοψηφία αυτών είναι ιατρικά βίντεο κωδικοποιημένα με **MPEG-1** κατάληξης **mpeg** ή **mpg**. Η επιλογή τους έγινε αρχικά με τη σκέψη να ασχοληθούμε περισσότερο με ιατρικά βίντεο που έχουν ένα διαφορετικό ενδιαφέρον από πλευράς χρησιμότητας. Στο συγκεκριμένο είδος βίντεο είναι δύσκολο να εντοπίσουμε αλλαγές μεταξύ των καρέ για το λόγο ότι δεν έχει μεγάλες αλλαγές στις σκηνές. Η ανάλυση τους είναι 352x240 αν και η περιοχή που μας είναι χρήσιμη είναι μικρότερη. Στη συνέχεια έχουμε τα *iriver.avi*, *foreman.avi*, *carphone.avi*. Το πρώτο είναι κωδικοποιημένο με **MPEG-4** (*Simple profile, DIVX*) με ανάλυση 320x240 ενώ τα άλλα 2 είναι **ασυμπίεστα AVI** (*raw,I420*) με ανάλυση 176x144. Όλα είναι κατάληξης **avi**. Τα μέχρι στιγμής βίντεο που αναφέρθηκαν ανήκουν στην κατηγορία των **SD βίντεο** (*Standard Definition videos*)

Περνάμε τώρα σε μεγαλύτερες αναλύσεις που ανήκουν στην κατηγορία των **HD βίντεο** (*High Definition videos*). Συνολικά 21 βίντεο χρησιμοποιούνται που ανήκουν σε αυτή την κατηγορία αλλά είναι διαφορετικών αναλύσεων μερικά από αυτά. Τα 6 από αυτά είναι ανάλυσης

1280x720, άλλα 3 είναι ανάλυσης 1920x1080 ενώ τα υπόλοιπα 12 είναι ανάλυσης 3840x2160 (εξαίρεση αποτελεί ένα βίντεο ανάλυσης 4096x1714). Όλα τα βίντεο είναι κωδικοποιημένα με **H.264/MPEG-4 Part 10 (High profile)** ή αλλιώς **AVC** (*Advanced Video Coding*).

Στο τέλος της εργασίας, στο **Παράρτημα**, παρατίθεται σχετικός πίνακας με όλα τα βίντεο που χρησιμοποιήθηκαν και τεχνικές λεπτομέρειες για το καθένα.

4. ΠΕΙΡΑΜΑΤΑ

Στο συγκεκριμένο κεφάλαιο θα γίνει εκτενής αναφορά στα αποτελέσματα τα οποία πήραμε χρησιμοποιώντας τους αλγορίθμους που αναφέραμε καθώς και όλα τα βίντεο. Χάρη σε αυτά καταλήξαμε σε χρήσιμα συμπεράσματα όσων αφορά τους αλγορίθμους που χρησιμοποιήσαμε τα οποία θα παρουσιαστούν στο επόμενο κεφάλαιο.

Η καταγραφή των αποτελεσμάτων ήταν μια επίπονη διαδικασία λόγω και της ποσότητας αυτών. Όπως έχει ήδη αναφερθεί πήραμε μετρήσεις με βάση διαφορετικά σενάρια για κάθε αλγόριθμο τα οποία θα παρουσιαστούν στη συνέχεια μέσα από πίνακες με τις τιμές που πήραμε όπως είναι το ποσοστό των μαρκαρισμένων καρέ αλλά και με τη μορφή διαγραμμάτων. Οι πίνακες περιέχουν τα συνολικά καρέ, τα επιλεγμένα καρέ και τα επιλεγμένα καρέ υπό μορφή ποσοστού. Τα διαγράμματα κατασκευάστηκαν με γνώμονα την καλύτερη παρουσίαση των αποτελεσμάτων που πήραμε έτσι ώστε άμεσα ο αναγνώστης να πάρει μια πρώτη ιδέα για την δυναμική του κάθε αλγόριθμου.

Η σειρά παρουσίασης των μετρήσεων είναι η εξής:

- Διαφορές στα **pixel** (*absolute differences*) μεταξύ των καρέ.
- Διαφορές στα **ιστογράμματα** (*histogram differences*) μεταξύ των καρέ.
- Διαφορές στη **συμπίεση** (*compression differences*) μεταξύ των καρέ.
- **Αναλογία Αλλαγής Ακμών** (*Edge Change Ratio, ECR*) μεταξύ των καρέ.
- Διαφορές στις **ενέργειες** μεταξύ των καρέ κάνοντας χρήση του **Haar Wavelet**.

Τα συστήματα πάνω στα οποία πάρθηκαν οι μετρήσεις είναι τα εξής:

Desktop

- Μητρική πλακέτα : Gigabyte P31-DS3L
- Επεξεργαστής : Intel Core 2 Duo E7200 @ 2.53 GHz (3 MB L3 cache)
- Μνήμη RAM : 3 GB @ 800 Mhz DDR2
- Κάρτα γραφικών : NVIDIA GeForce 210, ATI Mobility FireGL9000/Radeon 9000
- Σκληρός δίσκος : Seagate 400 GB @ 7200 rpm

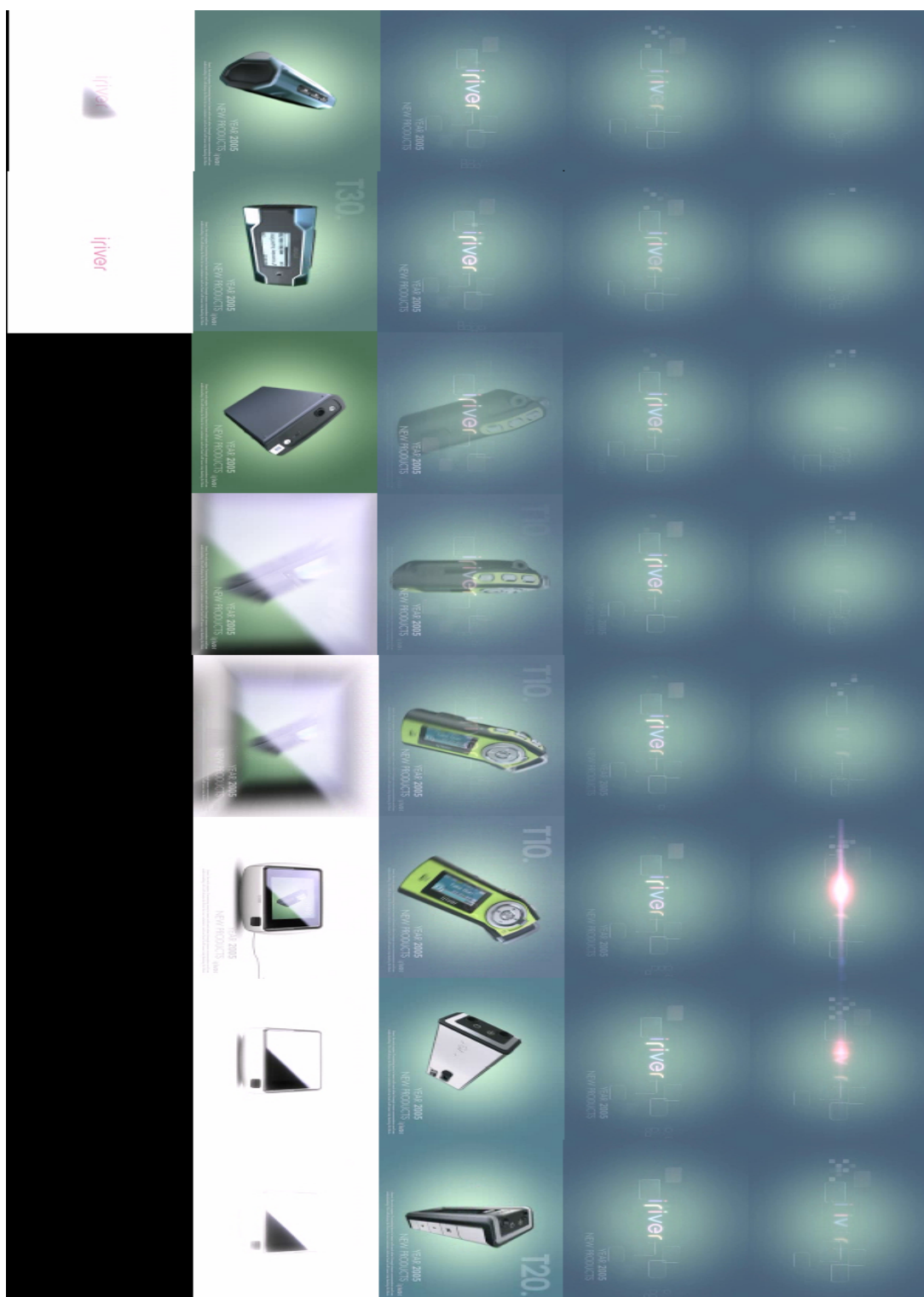
- Λειτουργικό σύστημα : Linux Lite 2.0 32bit

Laptop

- Μοντέλο : TOSHIBA Satellite C660-1X8
- Επεξεργαστής : Intel® Pentium® B940 @ 2.00 GHz (2 MB L3 cache)
- Μνήμη RAM : 3 GB @ 1333 Mhz DDR3
- Κάρτα γραφικών : NVIDIA® GeForce® 315M
- Σκληρός δίσκος : 320 GB @ 5400 rpm
- Λειτουργικό σύστημα : Ubuntu 12.04 32bit

Τα διαγράμματα βρίσκονται συγκεντρωμένα στο **Παράρτημα** στο τέλος του βιβλίου.

Παρακάτω φαίνεται ένα δείγμα από την επιλογή των καρέ τα οποία έχουν εξαχθεί σαν εικόνες και έχουν ενσωματωθεί όλα σε μία για την καλύτερη προβολή τους. Τα βίντεο αυτά έχουν δημιουργηθεί με τον αλγόριθμο των διαφορών του ECR μεταξύ του προηγούμενου και του επόμενου καρέ. Σαν κατώφλι επιλογής των καρέ έχει χρησιμοποιηθεί το 0.5. Δηλαδή αν η διαφορά είναι μεγαλύτερη της τιμής αυτής τότε επιλέγεται το εκάστοτε καρέ.



iriver.avi

Μέγεθος: 887,7 KB (887718 bytes)

Διαστάσεις: 320x240

Διάρκεια: 00:27.07 Δευτερόλεπτα



senna_crash.mp4

Μέγεθος: 1,4 MB (1375691 bytes)

Διαστάσεις: 1280x720

Διάρκεια: 00:13.00 Δευτερόλεπτα



challenger_explosion.mp4

Μέγεθος: 10,4 MB (10393322 bytes)

Διαστάσεις: 1280x720

Διάρκεια: 01:10.00 Δευτερόλεπτα

5. ΑΠΟΤΕΛΕΣΜΑΤΑ – ΣΥΜΠΕΡΑΣΜΑΤΑ

Συνοψίζοντας όσα έχουμε πει μέχρι στιγμής στα πλαίσια αυτής της εργασίας, φτάσαμε στο τελευταίο κεφάλαιο που αφορά την παρουσίαση των αποτελεσμάτων και των συμπερασμάτων που αποκομίσαμε από όλες αυτές τις δοκιμές με πληθώρα βίντεο και αλγορίθμων. Εφαρμόζοντας όλους αυτούς τους αλγορίθμους αποκτήσαμε τη δυνατότητα να εξετάσουμε και να κατανοήσουμε τη διαφορετικότητα τους αλλά και τις ικανότητες του καθενός ανάλογα τις συνθήκες αλλά και το περιβάλλον στο οποίο χρησιμοποιείται. Συνεπώς μερικοί αλγόριθμοι δίνουν καλύτερα αποτελέσματα (από την άποψη αριθμού μαρκαρισμένων καρέ) σε συγκεκριμένα βίντεο από ότι κάποιοι άλλοι. Πέρα από τα αποτελέσματα που πήραμε, δεν ήταν δυνατό να αντιμετωπίσουμε και τις σχετικές δυσκολίες και προβλήματα που παρόλα αυτά αντιμετωπίστηκαν με επιτυχία. Κυριότερος παράγοντας αποτυχίας εφαρμογής των αλγορίθμων είναι η κωδικοποίηση που χρησιμοποιείται από τα εκάστοτε βίντεο. Αυτό παρατηρήθηκε περισσότερο στα ιατρικά βίντεο. Το βασικό μειονέκτημα κατά τη συγγραφή των προγραμμάτων που χρησιμοποιήθηκαν είναι η μερική έλλειψη υποστήριξης, σε κάποια σημεία κυρίως όσον αφορά την κωδικοποίηση των βίντεο, της βιβλιοθήκης OpenCV όσον αφορά το λειτουργικό σύστημα (*Linux*) πάνω στο οποίο ήταν εγκατεστημένη. Βέβαια αυτό δεν μας εμπόδισε να βρούμε εναλλακτικές λύσεις για να αντιμετωπίσουμε σχετικά προβλήματα.

Περνώντας τώρα στην ουσία του κεφαλαίου ήρθε η ώρα να αναλύσουμε κάποια βασικά συμπεράσματα που εξάγαμε από τα αποτελέσματα. Αυτά προκύπτουν και είναι άρρηκτα συνδεδεμένα με τα βίντεο που έχουμε επιλέξει για επεξεργασία. Αυτό σημαίνει ότι ένας διαφορετικός συνδυασμός από βίντεο θα είχε διαφορετικά αποτελέσματα. Τα περισσότερα βίντεο επιλέχθηκαν ειδικά για αυτό το σκοπό, την ποικιλομορφία των αποτελεσμάτων. Αυτό που φαίνεται πρώτα από όλα από όλους τους αλγορίθμους είναι διαφορετική συμπεριφορά που έχουν σε αργές και γρήγορες καταστάσεις αντίστοιχα (*slow motion*, *fast motion*). Θα μπορούσαμε να πούμε ότι όταν έχουμε αργή κίνηση περισσότερα καρέ επιλέγονται από το συνηθισμένο ειδικά στην περιοχή που εμφανίζεται αυτό το φαινόμενο καθώς φαίνεται να αλλάζει σιγά σιγά και όταν τελικά γίνει μεγαλύτερη η διαφορά επιλέγεται. Αντίθετα στην γρήγορη κίνηση βλέπουμε ότι έχουμε απώλειες στα καρέ που επιλέγονται με την έννοια ότι θα μπορούσαν να είχαν μαρκαριστεί περισσότερα από το αναμενόμενο. Αυτό δεν σημαίνει ότι ο αλγόριθμος είναι λάθος. Έχει να κάνει με πολλές παραμέτρους όπως είναι ο τρόπος υπολογισμού της μέτρησης αλλά και το όριο το οποίο χρησιμοποιούμε για να επιλέξουμε τα καρέ. Διαφορετικές παράμετροι μας δίνουν και διαφορετικά αποτελέσματα. Επίσης παίζει ρόλο και το αν υπάρχουν αλλαγές σκηνών στο βίντεο με την έννοια ότι η θεματική αυτών μεταβάλλεται σε σύντομο χρονικό διάστημα και δεν μιλάμε για μια σκηνή όπου το περιεχόμενο αλλάζει σταδιακά όσο διαρκεί το βίντεο. Αυτό παρατηρείται περισσότερο στα ιατρικά βίντεο όπου έχουμε μια κάμερα κινείται στο εσωτερικό του σώματος. Παρόλο που αλλάζει το περιεχόμενο υπάρχει μία μοναδική σκηνή η οποία αλλάζει σταδιακά. Αυτό μπερδεύει κάποιους αλγορίθμους και φτάνουν στο σημείο να μαρκάρουν συνεχόμενα καρέ νομίζοντας ότι υπάρχει μεγάλη διαφορά μεταξύ αυτών. Επιπλέον, στα περισσότερα βίντεο υπάρχουν και διαφορετικά εφέ

τα οποία επηρεάζουν την απόδοση των αλγορίθμων ως προς τον αριθμό των μαρκαρισμένων καρέ. Στο σημείο που εμφανίζονται αυτά τα εφέ, όπως είναι το **fade in/out**, **dissolve** τα οποία αναφέρθηκαν σε προηγούμενο κεφάλαιο, ανάλογα με τον αλγόριθμο που χρησιμοποιείται κάθε φορά παρατηρούνται προβλήματα (μερικά από τα οποία αναφέρθηκαν παραπάνω) τα οποία δυσκολεύουν το έργο του αλγορίθμου.

Για να συγκεκριμενοποιήσουμε τις παρατηρήσεις μας θα μιλήσουμε για ένα φαινόμενο που παρατηρήθηκε μόνο στον αλγόριθμο υπολογισμού διαφορών με βάση τη συμπίεση του κάθε καρέ. Τα περισσότερα βίντεο που χρησιμοποιήθηκαν ήταν μικρής ανάλυσης και έτσι δεν είναι τόσο ορατό το φαινόμενο αυτό το οποίο εμφανίζεται μόνο στα υψηλής ανάλυσης βίντεο (*1280x720 και άνω με κωδικοποίηση H.264/MPEG-4 Part 10 (High profile)*). Ουσιαστικά σε αυτά τα βίντεο ο αριθμός των επιλεγμένων καρέ είναι μικρός σε σχέση με τη διάρκεια του βίντεο αλλά και του περιεχομένου του καθενός σε σχέση με τους άλλους αλγορίθμους όπου τα ίδια βίντεο μας δίνουν μεγαλύτερα νούμερα. Εδώ έχουμε να κάνουμε με καρέ μεγαλύτερης ανάλυσης, συνεπώς μεγαλύτερου μεγέθους, και καλύτερης ποιότητας. Όπως παρατηρείται και από τις μετρήσεις οι διαφορές μεταξύ των καρέ τείνουν να είναι μεγαλύτερες όταν υπάρχουν αλλαγές μεταξύ των καρέ και αυτό αποτυπώνεται καλύτερα σε αυτά τα βίντεο καθώς η ποιότητα τους είναι καλύτερη. Για να το κατανοήσουμε καλύτερα πρέπει να δούμε ότι η συμπίεση που εφαρμόζεται στο κάθε καρέ είναι άρρηκτα συνδεδεμένη με την τιμή των pixel αυτού. Καθώς στα HD βίντεο έχουμε μεγαλύτερη πιστότητα στο εκάστοτε καρέ η τιμή που θα έχει μετά τη συμπίεση θα είναι πιο ακριβής σε σχέση με ένα SD βίντεο όπου έχει μικρότερη ανάλυση και έχει κωδικοποιηθεί με έναν όχι και τόσο αποδοτικό τρόπο.

Σε αλγορίθμους όπως είναι ο υπολογισμός των διαφορών μεταξύ των pixel αλλά και των ιστογραμμάτων, επειδή οι υπολογισμοί γίνονται ουσιαστικά πάνω στις ίδιες τις τιμές που αποτελούν το καρέ, είναι πολύ βασική η ποιότητα αυτών έτσι ώστε να μην επηρεάζεται πάρα πολύ ο αλγόριθμος από αλλοιωμένα pixel ή περιοχές όπου οι αποχρώσεις είναι παρόμοιες ή ακόμα και ίδιες. Σε αυτό βοηθάει η κωδικοποίηση που εφαρμόζεται σε κάποια βίντεο όπου αποτελεί ένα βασικό βήμα για καλύτερα αποτελέσματα. Γενικότερα, όσο καλύτερη είναι η ποιότητα τόσο καλύτερα θα είναι και τα αποτελέσματα.

Πέρα από όσα προαναφέρθηκαν, τα οποία είναι κυρίως θέματα της υλοποίησης, μιλώντας πιο γενικά και βασικά αυτή είναι η ουσία της εργασίας και το αποτέλεσμα όλων των αλγορίθμων, τα επιλεγμένα καρέ είναι αυτά τα οποία θα πρέπει να κρατήσουμε και αποτελούν μια περίληψη του αρχικού μας βίντεο. Με λίγα λόγια αν κάποιος δει ένα τέτοιο επεξεργασμένο βίντεο να μπορέσει να καταλάβει σε γενικές γραμμές και στο μέτρο του δυνατού τι παρουσιάζει το αρχικό. Αυτό είναι πάρα πολύ χρήσιμο σε διάφορες καταστάσεις όπου είναι δύσκολο και από θέμα χρόνου αλλά και απόδοσης (αν θεωρήσουμε ότι μεταδίδουμε ένα τέτοιο βίντεο μέσω δικτύου) να παρουσιάσουμε το αρχικό βίντεο στην κανονική του διάρκεια, με όλες τις λεπτομέρειες που περιείχε οι οποίες κάποιες φορές είναι περιττές.

6. ΒΙΒΛΙΟΓΡΑΦΙΑ

Ιστοσελίδες

- [OpenCV], <http://opencv.org/>, *OpenCV (Open source computer vision)*
- [OpenCV Docs], <http://docs.opencv.org/>, *OpenCV Documentation*
- [OpenCV Mailing List], <http://opencv-users.1802565.n2.nabble.com/>, *OpenCV Mailing List*
- [GStreamer], <http://gstreamer.freedesktop.org/>, *GStreamer - open source multimedia framework*
- [FFmpeg], <http://ffmpeg.org/>, *FFmpeg*
- [LinuxCommand.org], <http://linuxcommand.org/>, *LinuxCommand.org*
- [YouTube], <http://www.youtube.com/>, *YouTube*
- [Ultra Video Group], <http://ultravideo.cs.tut.fi/>, *Ultra Video Group*
- [XhmikosR's Builds], <http://xhmikosr.1f0.de/>, *XhmikosR's Builds*
- [**Pokemon World Championships 2012**], <http://www.pokemonworldchampionships.com/2012/videos>, *Pokemon World Championships 2012*

Δημοσιεύσεις

- [**Ζεϊναλιπούρ 2014**]. Δημήτρης Ζεϊναλιπούρ. Διάλεξη 8, Το Κέλυφος Bash (**2014**). *ΕΠΛ 371 - Προγραμματισμός Συστημάτων*. Τμήμα Πληροφορικής. Πανεπιστήμιο Κύπρου.
- [**Ωραιοζήλη 2011**]. Καργοπούλου Ωραιοζήλη (**2011**), Διπλωματική εργασία. *Τεχνικές επεξεργασίας για βελτιστοποίηση υπερηχογραφικών εικόνων και εξαγωγή χαρακτηριστικών, με χρήση του Matlab*. Πανεπιστήμιο Μακεδονίας, Τμήμα Εφαρμοσμένης Πληροφορικής.
- [**Sullivan et al. 2012**]. Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand (**2012**). *Overview of the High Efficiency Video Coding (HEVC) Standard*. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 12, 1649-1668.
- [**Boreczky et Rowe. 1996**]. John S. Boreczky, Lawrence A. Rowe (**1996**). *Comparison of video shot boundary detection techniques*. Journal of Electronic Imaging 5(2), 122 – 128.
- [**Nasreen et Shobha. 2013**]. Azra Nasreen, Dr Shobha G. (**2013**). *Key Frame Extraction*

using Edge Change Ratio for Shot Segmentation. International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 11.

[**Rathod et al. 2013**]. Ganesh. I. Rathod, Dipali. A. Nikam. (2013). *An Algorithm for Shot Boundary Detection and Key Frame Extraction Using Histogram Difference*. International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 8.

[**Miene et al. 2004**]. A. Jacobs, A. Miene, G. T. Ioannidis, and O. Herzog (2004). *Automatic shot boundary detection combining color, edge, and motion features of adjacent frames*. TREC 2004.

[**Cerf et Kahn. 1974**]. Vinton G. Cerf, Robert E. Kahn (1974). *A Protocol for Packet Network Intercommunication*. IEEE Transactions on Communications, Vol 22, No 5 May 1974.

[**Tint et Soe. 2013**]. Khin Thandar Tint, Dr. Kyi Soe (2013). *Key Frame Extraction for Video Summarization Using DWT Wavelet Statistics*. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 2, No 5, May 2013

Βιβλία

[**Jack 2007**]. Keith Jack (2007). *Video Demystified*. Newnes.

[**Παπαμάρκος 2010**]. Νικόλαος Η. Παπαμάρκος (2010). *Ψηφιακή Επεξεργασία & Ανάλυση Εικόνων (Έκδοση 2η)*. Νικόλαος Η. Παπαμάρκος.

[**Bradski et Kaehler 2008**]. Gary Bradski, Adrian Kaehler (2008). *Learning OpenCV*. O'REILLY.

[**Χατζηγιαννάκης 2010**]. Νίκος Μ. Χατζηγιαννάκης (2010). *Η γλώσσα C σε βάθος (Έκδοση 3η)*. Εκδόσεις Κλειδάριθμος.

[**Comer et Stevens 2003**]. Douglas E. Comer, David L. Stevens (2003). *Δικτυακός Προγραμματισμός στο TCP/IP – Linux/POSIX Sockets (Έκδοση 3η)*. Εκδόσεις ΙΩΝ

[**Kurose et Ross 2012**]. James F. Kurose, Keith W. Ross (2012). *Δικτύωση Υπολογιστών – Προσέγγιση από Πάνω προς τα Κάτω. (Έκδοση 6η)*. Εκδόσεις Μ. Γκιούρδας.

ΠΑΡΑΡΤΗΜΑ

SD (Standard Definition) videos

Filename	Width	Height	FPS	Compression
apendiceproctroid.mpg	352	240	30	MPEG-1
argonpoliposis1.mpg	352	240	30	MPEG-1
argonpoliposis2.mpg	352	240	30	MPEG-1
Candididiasis_Marion1.mpg	352	240	24	MPEG-1
Candididiasis_Marion2.mpg	352	240	24	MPEG-1
Cappolipoiss1.mpg	352	240	24	MPEG-1
carphone.avi	176	144	12	MPEG-1
cecum2.mpg	352	240	30	MPEG-1
cecum3.mpg	352	240	30	MPEG-1
cecum.mpg	352	240	30	MPEG-1
ColonJaundince1.mpg	352	240	30	MPEG-1
ColonJaundince2.mpg	352	240	30	MPEG-1
ColonNBI1.mpg	352	240	24	MPEG-1
ColonNBI2.mpg	352	240	24	MPEG-1
ColonNBI3.mpg	352	240	24	MPEG-1
Colonoscopy2.mpg	352	240	30	MPEG-1
Colon_tattoo1.mpg	352	240	24	MPEG-1
Colon_tattoo2.mpg	352	240	24	MPEG-1
Colon_tattoo3.mpg	352	240	24	MPEG-1
Colon_tattoo4.mpg	352	240	24	MPEG-1
Degarro10.mpg	352	240	30	MPEG-1

Filename	Width	Height	FPS	Compression
Degarro11.mpg	352	240	30	MPEG-1
Degarro12.mpg	352	240	30	MPEG-1
Degarro13.mpg	352	240	30	MPEG-1
Degarro14.mpg	352	240	30	MPEG-1
Degarro1.mpg	352	240	30	MPEG-1
Degarro2.mpg	352	240	30	MPEG-1
Degarro3.mpg	352	240	30	MPEG-1
Degarro4.mpg	352	240	30	MPEG-1
Degarro5.mpg	352	240	30	MPEG-1
Degarro6.mpg	352	240	30	MPEG-1
Degarro7.mpg	352	240	30	MPEG-1
Degarro8.mpg	352	240	30	MPEG-1
Degarro9.mpg	352	240	30	MPEG-1
Desgarrox1.mpg	352	240	24	MPEG-1
Enteroscopyzx1.mpg	352	240	24	MPEG-1
FamilalGastricPolip1.mpg	352	240	24	MPEG-1
FamilalGastricPolip3.mpg	352	240	24	MPEG-1
famliaranastomosis2.mpg	352	240	30	MPEG-1
famliaranastomosis.mpg	352	240	30	MPEG-1
foreman.avi	176	144	12	MPEG-1
GastrocFolders.mpg	352	240	30	MPEG-1
GistRectal1.mpg	352	240	24	MPEG-1
guava1.mpg	352	240	30	MPEG-1
guava2.mpg	352	240	30	MPEG-1

Filename	Width	Height	FPS	Compression
guava3.mpg	352	240	30	MPEG-1
iriver.avi	320	240	15	MPEG-1
Minorpapilapoliposis.mpg	352	240	30	MPEG-1
perforacion1.mpg	352	240	30	MPEG-1
perforacion2.mpg	352	240	30	MPEG-1
perforacion3.mpg	352	240	30	MPEG-1
perforacion4.mpg	352	240	30	MPEG-1
PJuvenil10.mpg	352	240	30	MPEG-1
PJuvenil11.mpg	352	240	30	MPEG-1
PJuvenil12.mpg	352	240	30	MPEG-1
PJuvenil13.mpg	352	240	30	MPEG-1
PJuvenil14.mpg	352	240	30	MPEG-1
PJuvenil15.mpg	352	240	30	MPEG-1
PJuvenil16.mpg	352	240	30	MPEG-1
PJuvenil17.mpg	352	240	30	MPEG-1
PJuvenil18.mpg	352	240	30	MPEG-1
PJuvenil19.mpg	352	240	30	MPEG-1
PJuvenil1.mpg	352	240	30	MPEG-1
PJuvenil20.mpg	352	240	30	MPEG-1
PJuvenil21.mpg	352	240	30	MPEG-1
PJuvenil22.mpg	352	240	30	MPEG-1
PJuvenil24.mpg	352	240	30	MPEG-1
PJuvenil2.mpg	352	240	30	MPEG-1
PJuvenil3.mpg	352	240	30	MPEG-1

Filename	Width	Height	FPS	Compression
PJuvenil4.mpg	352	240	30	MPEG-1
PJuvenil5.mpg	352	240	30	MPEG-1
PJuvenil6.mpg	352	240	30	MPEG-1
PJuvenil7.mpg	352	240	30	MPEG-1
PJuvenil8.mpg	352	240	30	MPEG-1
PJuvenil9.mpg	352	240	30	MPEG-1
postafpfollowup.mpg	352	240	30	MPEG-1
RectalIdiopathicUlcer1.mpg	352	240	30	MPEG-1
RectalIdiopathicUlcer2.mpg	352	240	30	MPEG-1
RectalIdiopathicUlcer3.mpg	352	240	30	MPEG-1
RectalIdiopathicUlcer4.mpg	352	240	30	MPEG-1
rectaltrauma1.mpg	352	240	30	MPEG-1
rectaltrauma2.mpg	352	240	30	MPEG-1
rectalvarix2.mpg	352	240	30	MPEG-1
rectalvarix.mpg	352	240	30	MPEG-1
rectoflexionsigmoidesrecto.mpg	352	240	30	MPEG-1
Rectovaginal1.mpg	352	240	30	MPEG-1
Rectovaginal2.mpg	352	240	30	MPEG-1
Rociox1.mpg	352	240	30	MPEG-1
Rociox2.mpg	352	240	30	MPEG-1
Rociox3.mpg	352	240	30	MPEG-1
Splenicflexurecolon.mpg	352	240	30	MPEG-1
TABLETAX1.mpg	352	240	30	MPEG-1
TABLETAX2.mpg	352	240	30	MPEG-1

Filename	Width	Height	FPS	Compression
Terminalileon1.mpg	352	240	30	MPEG-1
Terminalileon2.mpg	352	240	30	MPEG-1
TerminalIliumQ1.mpg	352	240	24	MPEG-1
TerminalIliumQ2.mpg	352	240	24	MPEG-1
TerminalIliumQ3.mpg	352	240	24	MPEG-1
Thecappolyps1.mpg	352	240	30	MPEG-1
Thecappolyps2-a.mpg	352	240	30	MPEG-1
Thecappolyps2.mpg	352	240	30	MPEG-1
videocolonoscopy2.mpg	352	240	30	MPEG-1
videocolonoscopy3.mpg	352	240	30	MPEG-1
videocolonoscopy4.mpg	352	240	30	MPEG-1

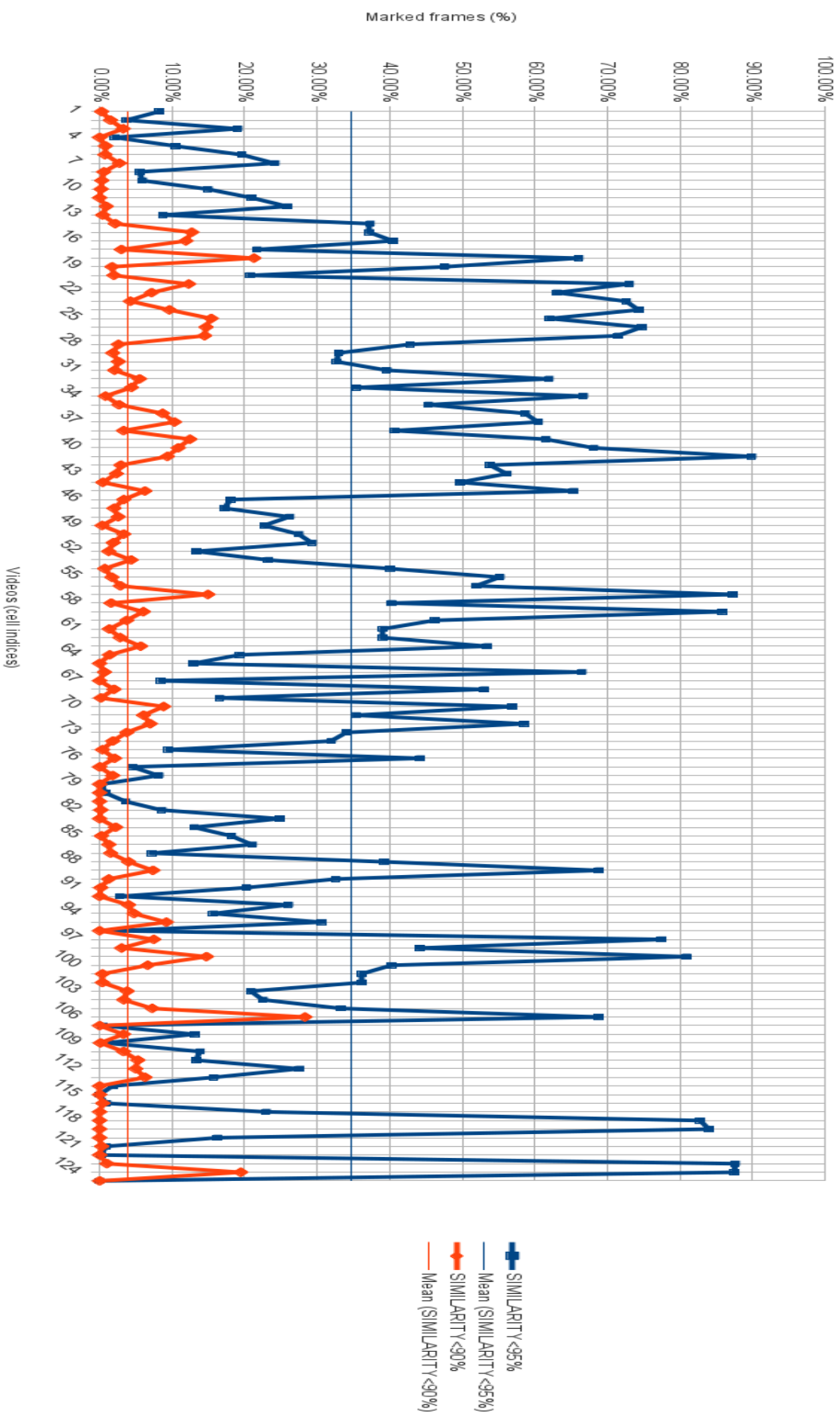
HD (High Definition) videos

Filename	Width	Height	FPS	Compression
2012_Pokemon_World_Championships- _Aloha_from_Hawaii!.mp4	1280	720	23.98	H.264/MPEG-4 Part 10 (High profile)
AMD_FX_Bulldozer_Guinness_World_Record.mp4	1920	1080	29.97	H.264/MPEG-4 Part 10 (High profile)
A_supercell_near_Booker,_Texas.mp4	1920	1080	25	H.264/MPEG-4 Part 10 (High profile)
Bosphorus_3840x2160_30fps_420_8bit_AVC_MP4.mp4	3840	2160	30	H.264/MPEG-4 Part 10 (High profile)
challenger_explosion.mp4	1280	720	30	H.264/MPEG-4 Part 10 (High profile)
Coastguard_H264.mp4	3840	2160	23.98	H.264/MPEG-4 Part 10 (High profile)
Cold_Winter_Night_Time-Lapse_4k_Resolution.mp4	4096	1714	25	H.264/MPEG-4 Part 10 (High profile)
CrowdRun_2160p50.x264.CRF25.mkv	3840	2160	50	H.264/MPEG-4 Part 10 (High profile)
DucksTakeOff_2160p50.x264.CRF25.mkv	3840	2160	50	H.264/MPEG-4 Part 10 (High profile)
InToTree_2160p50.x264.CRF25.mkv	3840	2160	50	H.264/MPEG-4 Part 10 (High profile)
ksc_051611_sts134_launch_1080.wmv	1920	1080	24	H.264/MPEG-4 Part 10 (High profile)
Loeb_testing_before_the_Rallye_Monte-Carlo_-_Best-of-RallyLivecom.mp4	1280	720	25	H.264/MPEG-4 Part 10 (High profile)
OldTownCross_2160p50.x264.CRF25.mkv	3840	2160	50	H.264/MPEG-4 Part 10 (High profile)

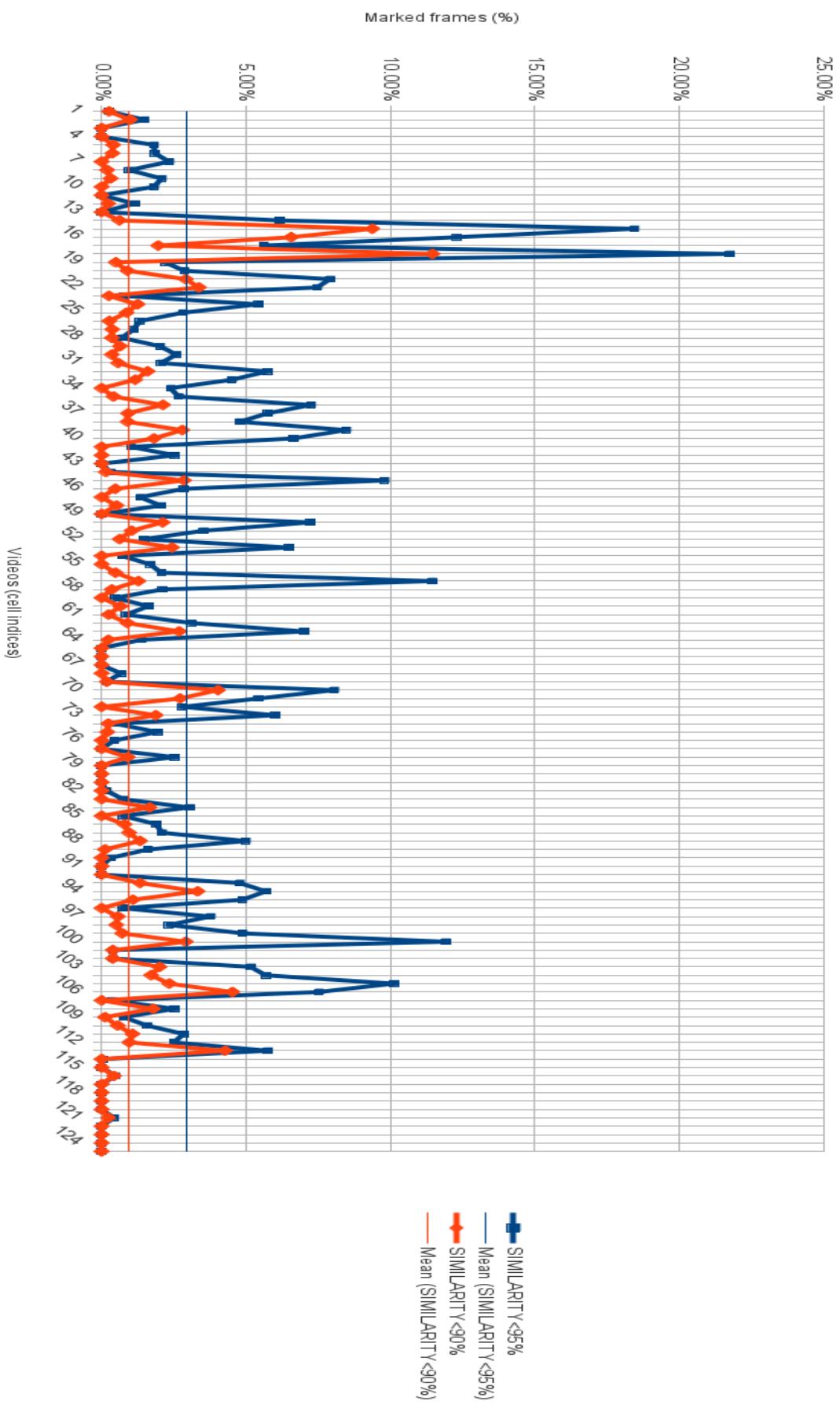
Filename	Width	Height	FPS	Compression
ParkJoy_2160p50.x264.CRF25.mkv	3840	2160	50	H.264/MPEG-4 Part 10 (High profile)
Pokemon_-_The_Origin_-_Trailer.mp4	1280	720	29.97	H.264/MPEG-4 Part 10 (High profile)
ReadySetGo_3840x2160_30fps_420_8bit_AVC_MP4.mp4	3840	2160	30	H.264/MPEG-4 Part 10 (High profile)
senna_crash.mp4	1280	720	30	H.264/MPEG-4 Part 10 (High profile)
Shakedown_-_2012_Acropolis_Rally_of_Greece_-_Best-of-RallyLivecom.mp4	1280	720	30	H.264/MPEG-4 Part 10 (High profile)
SlowMotion_Special_-_WRC_Acropolis_Rally_2013.mp4	1280	720	25	H.264/MPEG-4 Part 10 (High profile)
X-51A_WaveRider_hypersonic_scramjet_testbed.mp4	1920	1080	29.92	H.264/MPEG-4 Part 10 (High profile)

Παρακάτω φαίνονται σε διαγραμματική μορφή όλες οι μετρήσεις που πήραμε από κάθε αλγόριθμο καθώς και τα διάφορα σενάρια που εφαρμόστηκαν.

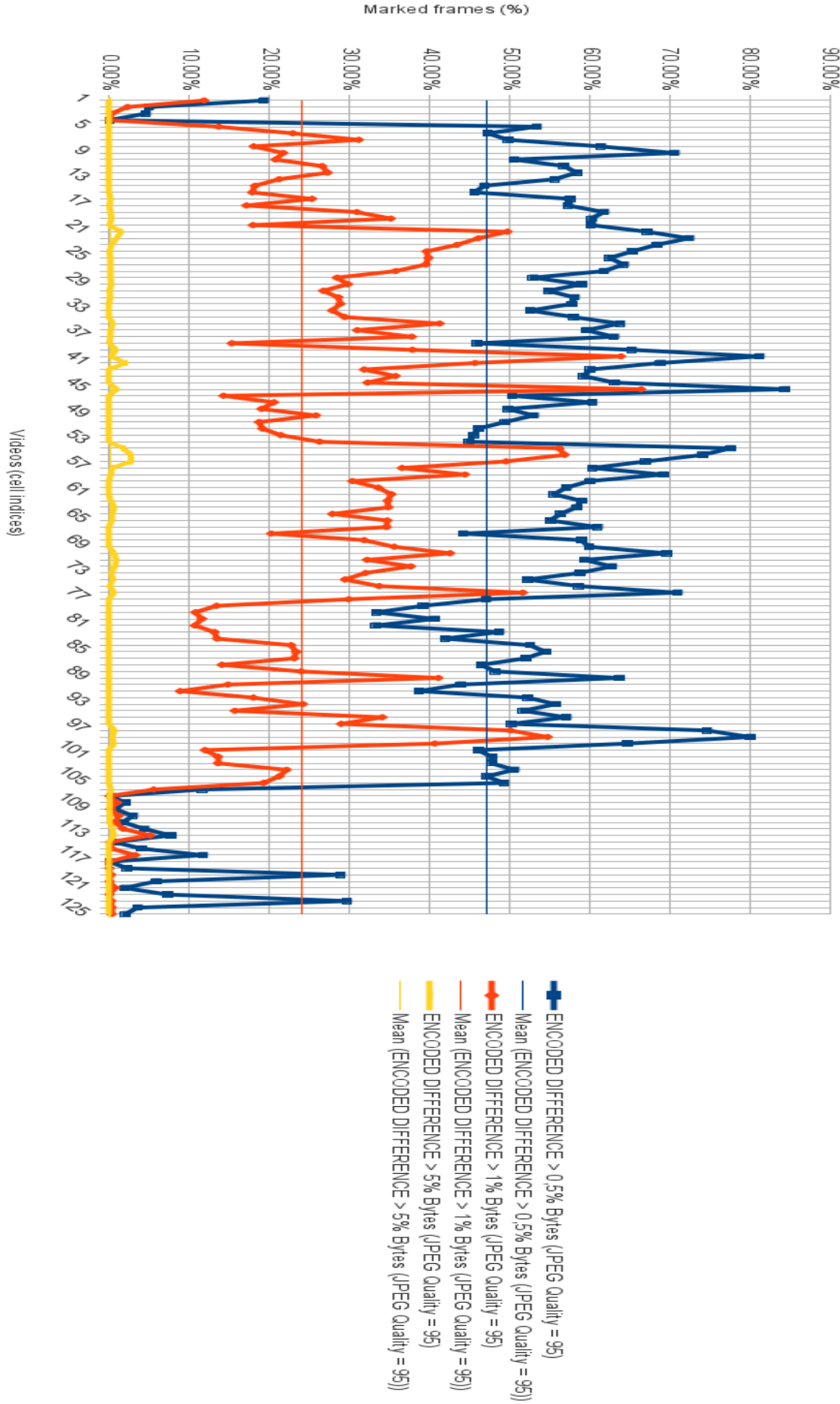
Absolute Differences



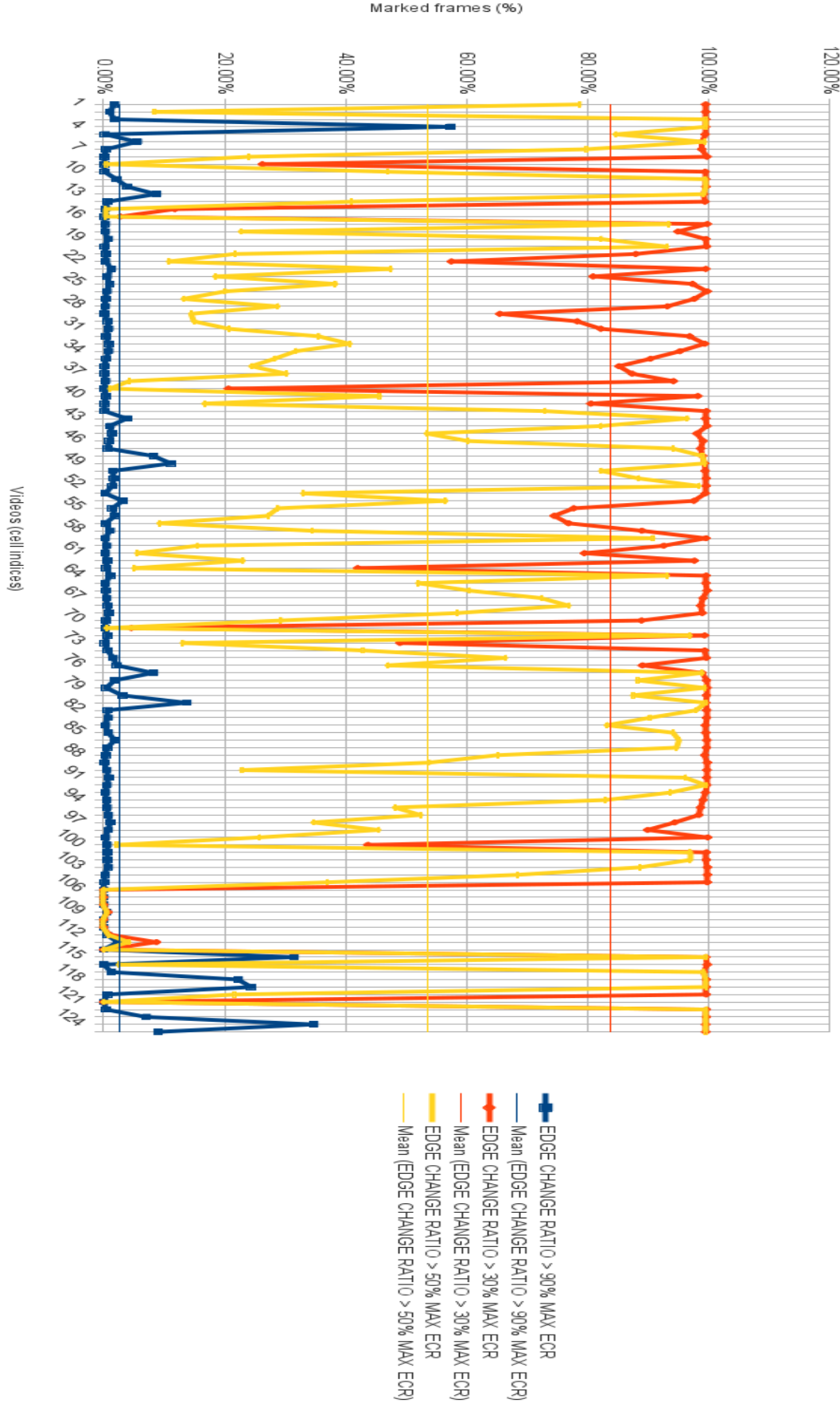
Histogram Differences



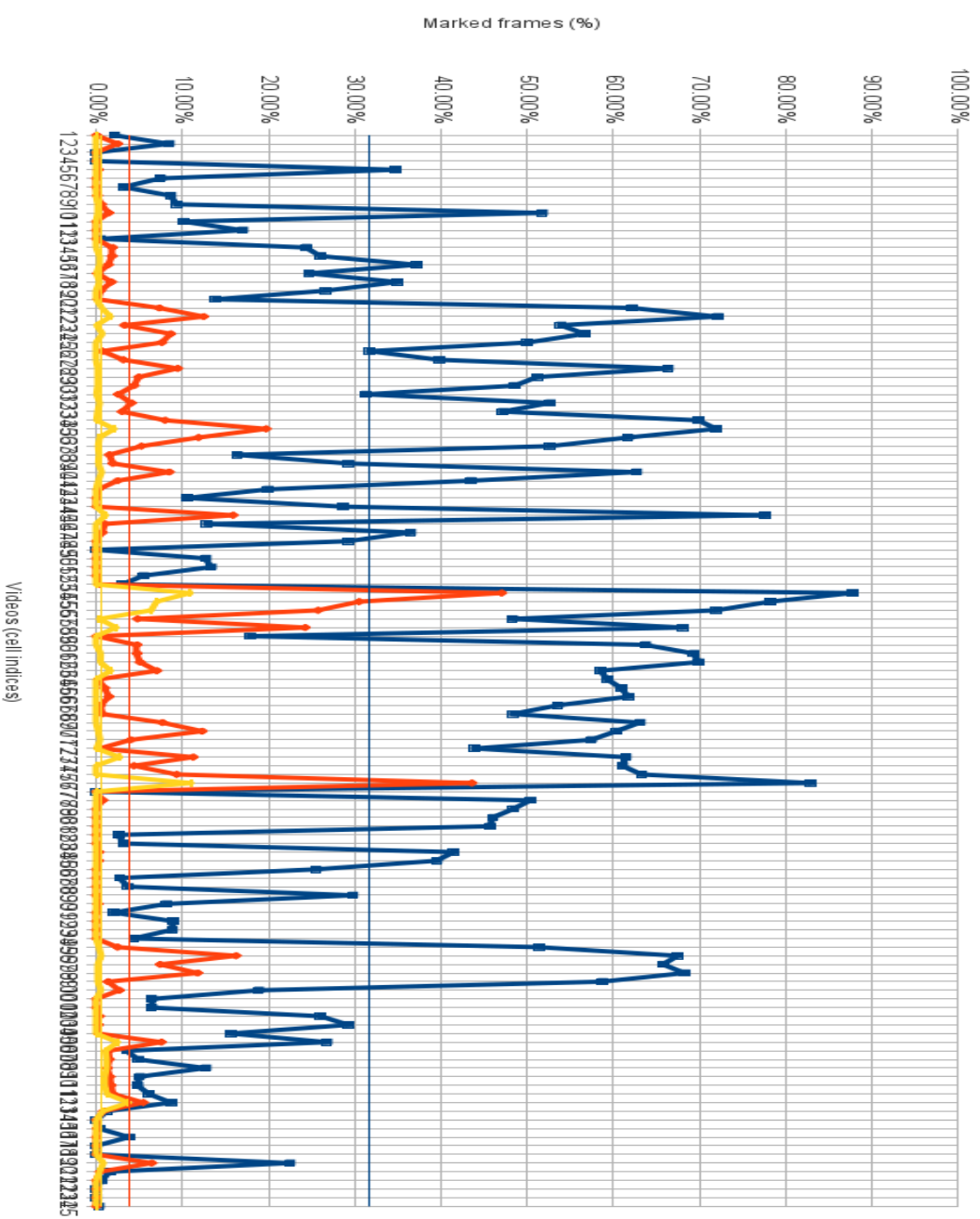
Compression Differences



ECR (Edge Change Ratio)



ECR Difference (Edge Change Ratio Difference)



Videos (cell indices)

- EDGE CHANGE RATIO DIFFERENCE > 0.5
- Mean (EDGE CHANGE RATIO DIFFERENCE > 0.5)
- EDGE CHANGE RATIO DIFFERENCE > 2
- Mean (EDGE CHANGE RATIO DIFFERENCE > 2)
- EDGE CHANGE RATIO DIFFERENCE > 4
- Mean (EDGE CHANGE RATIO DIFFERENCE > 4)

Wavelet

