ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Αναγνώριση Κίνησης σε Αρχεία Video

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μαυρίδης Μ. Παναγιώτης

Επιβλέπων : Στασινόπουλος Γεώργιος

Καθηγητής Ε.Μ.Π.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Αναγνώριση Κίνησης σε Αρχεία Video

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μαυρίδης Μ. Παναγιώτης

Επιβλέπων : Στασινόπουλος Γεώργιος

Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 17η Σεπτεμβρίου 2013.

.....	.....	.....
Στασινόπουλος Γεώργιος	Συκάς Δ. Ευστάθιος	Θεολόγου Ε. Μιχαήλ
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2013

.....

Μαυρίδης Μ. Παναγιώτης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

1. Περίληψη

Σκοπός αυτής της διπλωματικής εργασίας είναι ο χαρακτηρισμός, η ανάλυση και η καταγραφή της κίνησης που υπάρχει σε ένα βίντεο. Το πρόγραμμα δέχεται ως είσοδο ένα αρχείο οποιουδήποτε τύπου, το μετατρέπει σε επεξεργάσιμο τύπο και αναγνωρίζει και καταγράφει ποιες είναι οι διαφορετικές σκηνές που υπάρχουν καθώς και οι πιθανές σκηνές στο βίντεο αυτό. Οι πιθανές σκηνές εξάγονται από την επικάλυψη των 4 πιθανών σφαλμάτων σε κάθε αναγνώριση σκηνής. Από κάθε σκηνή εξάγονται στατιστικές μετρικές που χαρακτηρίζουν την κίνηση σε κάθε σκηνή και είναι τα απόλυτα μεγέθη του μέσου όρου κίνησης, η απόλυτη μέγιστη μετακίνηση κατά άξονα Χ αλλά και η απόλυτη μέγιστη μετακίνηση κατά τον άξονα Υ. Με αυτά τα δεδομένα μπορεί να έχουμε ως έξοδο κομμάτι από το αποτύπωμα του βίντεο. Αυτό το εξάγουμε σε ένα αρχείο κειμένου για πιθανή περαιτέρω επεξεργασία και για συμπεράσματα.

Λέξεις κλειδιά: βίντεο, αναγνώριση κίνησης, .yuv, στατιστική βίντεο, αναγνώριση σκηνών, μέγιστη κίνηση, μέσος όρος κίνησης, στατιστικό τεστ

2. Abstract

The purpose of this diploma thesis is the characterization, analysis and footprint of the motion that can be detected in a video file. The program takes as input any video file and transforms it in a raw editable file type in order to recognize and record the patterns of different scenes that exist in the video along with the potential and possible scene changes in the video. Potential scenes are extracted from the overlapping 4 error scenes in every scene recognition. From every scene there are some statistic metrics being extracted. That is the absolute average of the motion vector, the absolute max in the x axe and the absolute max in the y axe of the motion vector. With these data we can have the fingerprint of the video file. We export this fingerprint in a text file for possible further processing.

keywords: video, motion recognition, .yuv, video statistics, scene recognition, maximum motion, average motion, statistical test

3. Ευχαριστίες

Με το παρόν τεύχος θα ήθελα να ευχαριστήσω εγγράφως όσους με βοήθησαν τόσο ψυχολογικά όσο και έμπρακτα για να το περατώσω. Η παρούσα έκδοση έγινε με αρκετό κόπο και θυσίες όχι μονό από την πλευρά μου αλλά και από τα άτομα και οντότητες που το έργο τους δεν είναι εμφανές σε όλα τα σημεία.

Πρώτα από ολα θα ήθελα να ευχαριστήσω το Μεγάλο Δημιουργό Θεό. Αν δεν ήταν η Μεγάλη του Χάρη δεν θα είχα καταφέρει τίποτε σαν μικρός άνθρωπος που είμαι. Δεν θα είχα φτάσει σε αυτό το σημείο ούτε θα είχα κάνει αυτές τις επιλογές που έχω κάνει.

Έπειτα με την δίκαιη σειρά θα ήθελα να ευχαριστήσω τον καθηγητή μου κύριο Στασινόπουλο Γεώργιο που μου στάθηκε και με καθοδήγησε με πολύ μεράκι ο ίδιος σε κάθε κομμάτι της δουλειάς που με αφορούσε άμεσα και δεν είχα προηγούμενη εμπλοκή και έπρεπε να γνωρίζω. Ήταν διαθέσιμος εκτός ωραρίου και με εμπιστεύτηκε για να αναλάβω την διπλωματική σε περίοδο όπου η σχολή δεν λειτουργούσε. Αυτό με βοήθησε πολύ να θέσω τους στόχους μου και να προχωρήσω. Επίσης κατάλαβε όλες τις ιδιοτροπίες μου στον τομέα των μαθημάτων που χρωστούσα και το γεγονός ότι έκανα παράλληλα το στρατολογικό μου και έκανε πολύ μεγάλη υπομονή. Το αποτέλεσμα θέλω να πιστεύω ότι θα επιβραβεύσει την υπομονή και την αγάπη του.

Θα ήθελα να ευχαριστήσω τους γονείς μου Μιχαήλ και Φωτεινή που με πολύ αγάπη αυτοί δέχονται όλα αυτά τα χρονια τις ιδιοτροπίες μου και τις όποιες αντιδράσεις μου για να τελειώσω έγκαιρα την σχολή που διάλεξα με τόση αγάπη και μεράκι να τελειώσω. Δεν μου αρνήθηκαν ηθική, υλική και ψυχολογική βοήθεια για ότι θέλησα να κάνω. Η υπομονή και η αγάπη τους είναι ανυπέρβλητη σε σημείο αυτοθυσίας όχι μόνο για την διπλωματική αλλά και για ότι απασχολούσε αυτόν τον καιρό εμένα. Κομμάτι αυτής της βοήθειας ήταν και τα αδέρφια μου φυσικά που μου στάθηκαν και είχαν συμπληρωματικό ρόλο.

Εν κατακλείδι, ελπίζω να μην τους κούρασα παραπάνω από όσο ήταν πρέπον. Αν το έκανα και δεν το ήθελα ας με συγχωρήσουν.

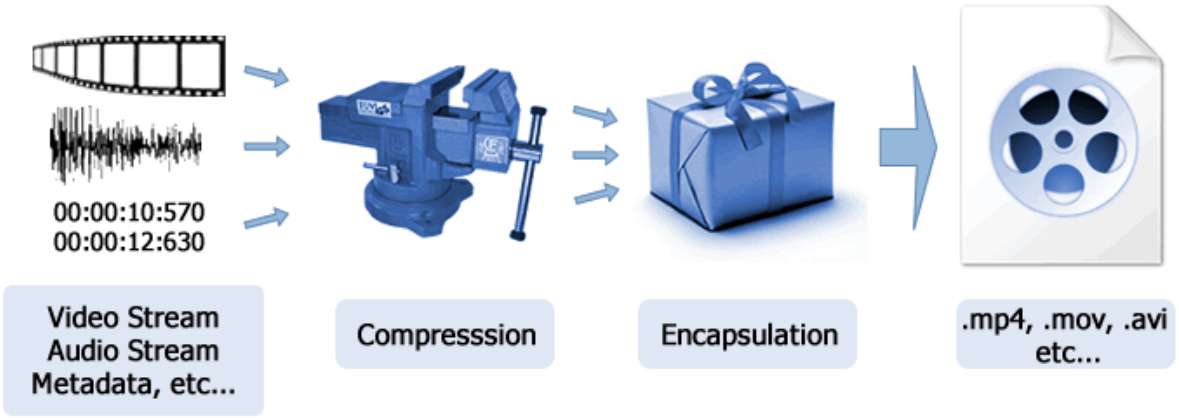
Μαυρίδης Παναγιώτης

ΗΜΜΥ ΕΜΠ

Πίνακας περιεχομένων

4. Εισαγωγή

Ο σκοπός της διπλωματικής είναι η αναγνώριση, ο χαρακτηρισμός και οι υπολογισμοί κίνησης σε αρχεία βίντεο. Το πρόγραμμα δέχεται ως είσοδο ένα αρχείο βίντεο οποιουδήποτε τύπου (.avi, .mpg, .mpeg, .flv, .wmv, .mkv κτλ) τα μετατρέπει σε μια



μορφή raw ασυμπίεστου βίντεο χωρίς header και άλλες ιδιομορφίες που ονομάζεται YUV. Αυτός ο τύπος βίντεο χρησιμοποιείται από το πρόγραμμα για να γίνουν οι υπολογισμοί. Η αναγνώριση των σκηνών και ένα κομμάτι του χαρακτηρισμού κίνησης δεν είναι εργασία της παρούσας διπλωματικής αλλά είναι αναπόσπαστο κομμάτι της εργασίας που έχει γίνει και που χρειάζεται για να γίνει κατανοητός ο αλγόριθμος που πραγματοποιήθηκε και η εργασία που έγινε σε αυτήν την διπλωματική.

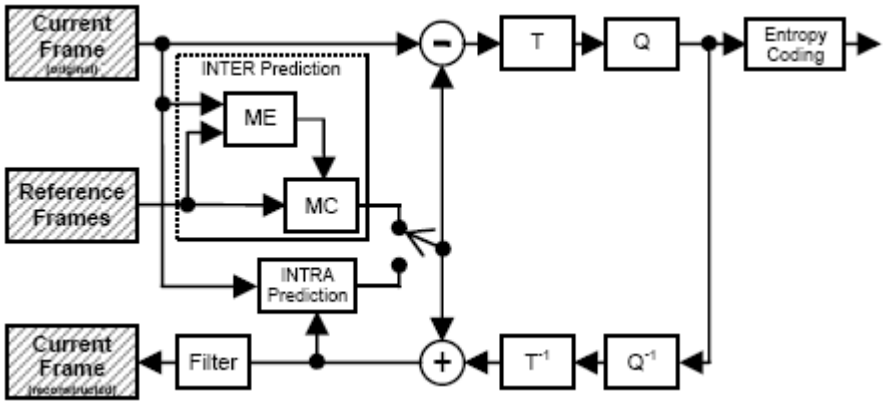
Η βασική επιδίωξη ήταν να εκμεταλλευτούμε κομμάτι της παρούσας έξοδου του προγράμματος για να προσθέσουμε κάποιες νέες λειτουργίες που είναι οι στατιστικές μετρικές για να συμπληρωθεί η έξοδος του προγράμματος. Η κύρια έξοδος του προγράμματος δημιουργεί το αποτυπώματο του βίντεο που επεξεργαζόμαστε. Από αυτό μπορούν να εξαχθούν συμπεράσματα για το βίντεο που αφορούν τις ιδιομορφίες του, την κίνηση που υπάρχει σε αυτό καθώς και τις σκηνές σε αυτό.

Το πρόγραμμα ήδη κάνει κάποια εργασία για την αναγνώριση σκηνών και πιθανών σκηνών πάνω στα βίντεο που επεξεργάζεται. Χαρακτηρίζει ως αλλαγή σκηνής τα σημεία του βίντεο που πληρούν ορισμένες προυποθέσεις και επ’ αυτών θεωρώντας ότι υπάρχουν σφάλματα σε αυτές τις προυποθέσεις επιτρέψαμε να μπορεί κανείς να συνυπολογίσει ως 4 σφάλματα ανά σκηνή.

Με αυτόν τον τρόπο μειώνονται οι πιθανότητες το πρόγραμμα να κάνει κάποιο λάθος στην εργασία του και έτσι να του ξεφύγει κάποια αλλαγή σκηνής. Οι σκηνές αυτές είναι

επικαλυπτόμενες και για αυτό παρουσιάζει ιδιαίτερο ενδιαφέρον πως θα της εμφανίσει και υπολογίσει κανείς αλλά και το πως θα κάνει υπολογισμούς σε αυτές. Κάθε σκηνή αποτελείται από μια αλληλουχία εικόνων. Σε κάθε πιθανή σκηνή αυτό που προσθέσαμε και ήταν και η εργασία για την διπλωματική ήταν ο υπολογισμός των μετρικών που αφορούν ένα τροποποιημένο και απλουστευμένο μέσο όρο που υπολογίζεται γρήγορα καθώς και το μέγιστο κατά την μετακίνηση στον X άξονα αλλά και το μέγιστο στην μετακίνηση κατά τον Y άξονα.

5. Λογική .H264, MPEG4 – Αλγόριθμος



Block διάγραμμα h264,mpeg encoding, decoding

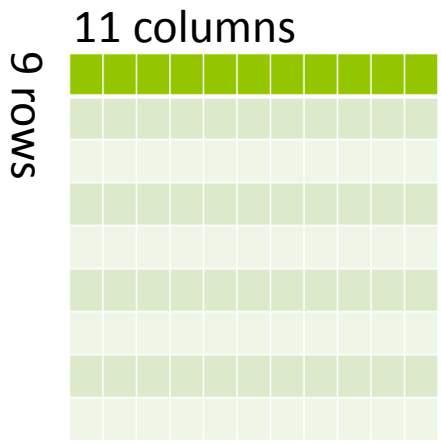
Η κεντρική και βασική ιδέα του προγράμματος είναι η χρήση του ανοικτού βρόχου αλγορίθμου .h264 και mpeg για κωδικοποίηση βίντεο. Για να γίνει αυτό κατανοητό θα εξηγήσουμε τον αρχικό τρόπου λειτουργίας του προγράμματος.

Κάθε βίντεο αποτελείται από μια αλληλουχία εικόνων που ονομάζονται frames. Κάναμε την παραδοχή ότι κάθε frame αποτελείται από 99 macroblocks (9x11). Έχοντας πρόσβαση αναφοράς σε κάθε macroblock χρησιμοποιούμε την κεντρική ιδέα video encoding για να ανακαστασκευάσουμε ένα frame από το προηγούμενο του ή από το εσωτερικό του. Συμβαίνει για εξοικονόμηση μεταφοράς περιττής πληροφορίας και επιτυγχάνεται κυρίως με 2 διαδικασίες: 1) το intra coding και το 2) inter coding.

5.1 Intra coding-prediction

Εκμεταλεύεται την εσωτερική δομή κάθε frame και προσπαθεί να ανακατασκευάσει τα macroblocks του frame με εσωτερική πληροφορία. Αυτό δικαιολογείται από την εσωτερική συνοχή μη παθολογικών σκηνών στα χρώματα(χρωματική συνέχεια), στα σχήματα, στις συμμετρίες που μπορεί να υπάρχουν σε μια εικόνα το λεγόμενο spatial locality. Τα blocks που υπολογίζονται από την διαδικασία αυτή ονομάζονται intra blocks ή i_blocks. Τα blocks αυτά αναφέρονται στο κάθε frame.

5.2 Inter coding-prediction



Αναπαράσταση Macroblock

Με αυτό ασχολούμαστε περισσότερο και αφορά την ομοιότητα μετά από μεταφορά macroblock μεταξύ των 2 frames. Αν υποθέσουμε ότι έχουμε ένα πρόσωπο που μιλάει και μετακινεί τα χείλη του για να το πετύχει τότε η εικόνα από το ένα frame στο άλλο πρακτικά θα περιέχει αρκετές μετακινήσεις macroblock πράγμα που σημαίνει ότι μπορει να ανακατασκευαστεί από τον μετασχηματισμό της μετακίνησής τους. (μεταφορά κατα τον άξονα χ των macroblock και κατά ψ των macroblock).

Σε αυτό υπάρχει μια μηχανή motion estimation που ελέγχει αν τα macroblock που προαναφέραμε μετακινούνται σε κάποιο άλλο κομμάτι της επόμενης σκηνής. Υποννοείται ότι κάθε macroblock ονομάζεται με κάποιο σημείο αναφοράς και η ταύτιση του με κάποιο από επόμενο block σηματοδοτεί ότι υπάρχει κίνηση από το ένα frame στο άλλο. Τα blocks που υπολογίζονται με αυτόν τον τρόπο ονομάζονται inter blocks ή me_blocks. Τα blocks αυτά αναφέρονται στο κάθε frame.

Επειδή υπάρχει θέμα σε πολλά video που έχουν μαύρο πλαίσιο από σκηνή σε σκηνή που είτε είναι σταθερό είτε αλλάζει γίνεται μια κανονικοποιήση και δεν μετριούνται αυτά σαν i_blocks.

Ένα frame για να χαρακτηριστεί ως αλλαγή σκηνής θα πρέπει να ξεπερνάει ένα εμπειρικό κατώφλι που έχουμε θέσει ως μέγιστο για ποσοστό i_blocks σε σχέση με me_blocks. Αν αυτό το κατώφλι το ξεπερνάμε θεωρούμε ότι έχουμε (πιθανή) αλλαγή σκηνής.

Και λέμε πιθανή αλλαγή σκηνής γιατί ουσιαστικά τα κριτήρια για την αλλαγή σκηνής που έχουμε θέσει δεν είναι ούτε ικανά ούτε αναγκαία. Δεν αρκούν από μόνα τους για να θεσπίσουν αλλαγή σκηνής. Μπορεί κάποια σκηνή να μην αναγνωριστεί και να είναι διαφορετική αλλά και κάποια σκηνή να μην είναι και να μαρκαριστεί ως αλλαγή σκηνής. Όμως για τα πλαίσια της ταχύτητας και των εφαρμογών που προορίζεται το συγκεκριμένο μοντέλο και ο αλγόριθμος είναι άριστο το αποτέλεσμα. Μάλιστα επειδή έχουμε μια ασφάλεια 4 σφαλμάτων για τις αλλαγές σκηνής υπάρχει αρκετά καλή εγγύηση ότι μπορούμε να βγάλουμε το χαρακτηριστικό αποτύπωμα σκηνών για ένα βίντεο.

5.3 (4-πλη) ασφάλεια σφάλματος

Η τετραπλή ασφάλεια σφαλμάτων που έχουμε χρησιμοποιήσει μπορεί να εγγυηθεί μεγάλη πιθανότητα σωστών αποτελεσμάτων για εύρεση των σκηνών. Στην ουσία μια σκηνή αναγνωρίζεται ως αλλαγή αλλά και η επόμενη που θεωρείται ως αλλαγή θεωρούμε ότι μπορεί να είναι στην θέση της προηγούμενης και ούτω κάθε εξής για 5 σκηνές. Με αποτέλεσμα να δημιουργούνται 5 διαφορετικά διαστήματα σκηνών για κάθε σκηνή με επικαλύψεις για τις κάθε αναγνωρίσεις. Αυτό μπορεί εύκολα να φανεί στο εξής παράδειγμα όταν έχουμε αναγνώριση αλλαγής σκηνής με αρίθμηση όπως φαίνεται στα παρακάτω frames. Οι σκηνές που κατασκευάζονται είναι αυτές που δεν υπολογίζονται από τον αλγόριθμο με το κατώφλι των intra_blocks αλλά κατασκευάζονται από τον αλγόριθμο των πιθανών σκηνών με το σύστημα 4απλής ασφάλεια σφάλματος που γίνεται.

Αναγνωρισμένες αλλαγές σκηνών:

0, 34, 56, 78, 97, 102, 156, 187, 214, 260

Πιθανό σετ σκηνών
0-34
0-56 πιθανή σκηνή (κατασκευάζεται)
0-78 πιθανή σκηνή (κατασκευάζεται)
0-97 πιθανή σκηνή (κατασκευάζεται)

0-102 πιθανή σκηνή (κατασκευάζεται)

Πιθανό σετ σκηνών
34-56
34-78 πιθανή σκηνή (κατασκευάζεται)
34-97 πιθανή σκηνή (κατασκευάζεται)
34-102 πιθανή σκηνή (κατασκευάζεται)
34-156 πιθανή σκηνή (κατασκευάζεται)

Πιθανό σετ σκηνών
56-78
56-97 πιθανή σκηνή (κατασκευάζεται)
56-102 πιθανή σκηνή (κατασκευάζεται)
56-156 πιθανή σκηνή (κατασκευάζεται)
56-187 πιθανή σκηνή (κατασκευάζεται)

Πιθανό σετ σκηνών
78-97
78-102 πιθανή σκηνή (κατασκευάζεται)
78-156 πιθανή σκηνή (κατασκευάζεται)
78-187 πιθανή σκηνή (κατασκευάζεται)
78-214 πιθανή σκηνή (κατασκευάζεται)

Πιθανό σετ σκηνών
97-102
97-156 πιθανή σκηνή (κατασκευάζεται)
97-187 πιθανή σκηνή (κατασκευάζεται)
97-214 πιθανή σκηνή (κατασκευάζεται)
97-260 πιθανή σκηνή (κατασκευάζεται)

Πιθανό σετ σκηνών
102-156
102-187 πιθανή σκηνή (κατασκευάζεται)
102-214 πιθανή σκηνή (κατασκευάζεται)
102-260 πιθανή σκηνή (κατασκευάζεται)
102-…… πιθανή σκηνή (κατασκευάζεται)

Πιθανό σετ σκηνών
156-187
156-214 πιθανή σκηνή (κατασκευάζεται)
156-260 πιθανή σκηνή (κατασκευάζεται)
156-….. πιθανή σκηνή (κατασκευάζεται)
156-….. πιθανή σκηνή (κατασκευάζεται)

Αυτές οι πεντάδες των πιθανών σετ σκηνών σχηματίζονται αυτόματα από τον αλγόριθμο με την βοήθεια μιας δομής στοίβας που δημιουργείται από τις αναγνωρισμένες σκηνές.

6. Υπολογισμός μετρικών για τις σκηνές και τις εναλλακτικές-πιθανές σκηνές

Εμεις ενδιαφερόμαστε να υπολογίσουμε σε πιθανό σετ σκηνών ορισμένες μετρικές που τις χαρακτηρίζουν. Οι μετρικές αυτές είναι μια παραλλαγή του απόλυτου μέσου όρου, είναι το απόλυτο μέγιστο κατά X και το απόλυτο μέγιστο κατά Y. Αυτό υπολογίζεται σε 3 επίπεδα:

1) Επίπεδο frame (μετρική βγαλμένη από τα macroblokcs)
2) Επίπεδο σκηνής (μετρική βγαλμένη από τις σκηνές)
3) Επίπεδο πιθανής σκηνής (μετρική βγαλμένη από τη μεταποίηση των αποθηκευμένων μετρικών σε επίπεδο σκηνής)

6.1 Παραλλαγή απόλυτου μέσου όρου

Στο επίπεδο frame υπολογίζεται από το άθροισμα των απολύτων τιμών της μετακίνησης κατά x και κατά y του macroblock η παραλλαγή του μέσου όρου.
Δηλαδή:
ΜέσοςΌρος = |ΜετακίνησηΚατάΧστοΕπόμενοFrame|+|
ΜετακίνησηΚατάΥστοΕπόμενοFrame|
Έτσι γλιτώνουμε τις πράξεις που θα χρειάζονται από το να λάβουμε καρτεσιανό μέτρο ως απόλυτο τιμή μετακίνησης και απλοποιείται και ο αλγόριθμος και μικραίνει ο χρόνος απόδοσης των αποτελεσμάτων.

Σε επίπεδο σκηνής είναι πολύ πιο απλό γιατί δεν μας απασχολεί το x και το y. Έχει υπολογιστεί η παραλλαγή μέσου όρου σε επίπεδο σκηνής και έτσι ο Μέσος όρος δίνεται από τον κλασσικό υπολογισμό:
Μέσος όρος = [Άθροισμα Μετρικών]/[Πλήθος]

Σε επίπεδο πιθανής σκηνής τα πράγματα γίνονται πιο περίπλοκα γιατί θα πρέπει να καταφύγουμε σε χρήση της μεθόδου του δυναμικού προγραμματισμού και για ευκολία να λάβουμε υπόψιν και το ακριβές μέγεθος των σφαλμάτων (4πλη ασφάλεια) που επιτρέπουμε για τις απόλυτες σκηνές. Αυτά θα εξηγηθούν αναλυτικότερα (στον αλγόριθμο) και με κώδικα για το πως έγιναν και για εξήγηση των επιλογών που έγιναν. Αλλά η απλή εξήγηση είναι ότι πρακτικά ο μέσος όρος για μια πιθανή σκηνή προκύπτει από τον βασικό τύπο:

ΜέσοςΟροςΠιθανήςΣκηνής =
[Μέσος όρος σκηνής1+ Μέσος όρος σκηνής2]/[ΠλήθοςFrameΣκηνής1 +ΠλήθοςFrameΣκηνής2]

Αν χρησιμοποιήσουμε για το σετ των σκηνών με κοινή αρχή των παραπάνω τύπο με υπολογισμό του επόμενου από τον προηγούμενο και την νέα πληροφορία έχουμε τον αναδρομικό τύπο για την μέθοδο του δυναμικού προγραμματισμού.

Αυτό γίνεται για κάθε πιθανή σκηνή όπως εξηγήσαμε παραπάνω από αυτές που κατασκευάζονται και όχι για αυτές που υπολογίζονται από τον αλγόριθμο με το κατώφλι.

6.2 Απόλυτο μέγιστο X και Y

Σε επίπεδο frame υπολογίζουμε την μέγιστη απόλυτη μετακίνηση είτε κατά X είτε κατά Y από όλα τα macroblock που υπάρχουν στο frame.

Σε επίπεδο σκηνής υπολογίζουμε από τα μέγιστα ανά frame (είτε X ή Y) το μεγαλύτερο από αυτά υπολογίζεται ως το μέγιστο σε επίπεδο σκηνής (X ή Y αντίστοιχα) με το γνωστό αλγόριθμο.

Σε επίπεδο πιθανής σκηνής αφού έχουμε υπολογισμένους το Απόλυτο μέγιστο για κάθε σκηνή (X ή Y) υπολογίζουμε μεταξύ των σκηνών πιο είναι το μεγαλύτερο και κρατάμε αυτό για την ένωση των 2 σκηνών που δημιουργεί την πιθανή σκηνή.

7. Κώδικας και αλγόριθμος με εξήγηση των μετρικών και του αλγορίθμου υπολογισμού τους

7.1 Κώδικας για Υπολογισμό μετρικών σε πρώτο επίπεδο (macroblock, frame)

Στο πρόγραμμα έχουμε τον υπολογισμό για κάθε μετρική σε 3 επίπεδα όπως αναφέραμε. Στο επίπεδο του frame φαίνεται παρακάτω ο κώδικας για κάθε υπολογισμό που γίνεται στην διαδικασία:

```
(void Cavc_encoderView::FrameSeqProcess(int frame_counter, uint8_t* currFrame , uint8_t* prevFrame))

if (pDoc->i_ME_Blocks_counted >0)
                pDoc->AbsoluteMeanOfMbs/=pDoc->i_ME_Blocks_counted;
        else
                pDoc->AbsoluteMeanOfMbs = 0;
        pDoc->SceneFrameCounter++;
        pDoc->AbsoluteMeanofScene+=pDoc->AbsoluteMeanOfMbs;
        //ypologismos maxX kai maxY se epipedo scene
```

```
        if (pDoc->MaxXofMbs>pDoc->MaxXofScene)
                pDoc->MaxXofScene= pDoc->MaxXofMbs;
        if (pDoc->MaxYofMbs>pDoc->MaxYofScene)
                pDoc->MaxYofScene= pDoc->MaxYofMbs;
        pDoc->MaxXofMbs=0;
        pDoc->MaxYofMbs=0;
```

Για να φτάσουμε όμως στο επίπεδο του frame πρέπει πρώτα να περάσουμε από το επίπεδο του macroblock αθροίζοντας ή συγκρίνοντας ανάλογα το τι θέλουμε να υπολογίσουμε, μέσο όρο, MaxX ή MaxY σε κάθε macroblock τις ανάλογες τιμές. Αυτό φαίνεται παρακάτω πολύ καθαρά. Είναι φανερό ότι έχουμε επιλέξει να πάρουμε τις απόλυτες τιμές για κάθε διάνυσμα X, Y και έτσι δεν έχουμε να κάνουμε με αρνητικές τιμές. Αυτό φαίνεται ξεκάθαρα παρακάτω:

```
(void Cavc_encoderDoc::FrameSeq_MB_Process(uint8_t* currFrame , uint8_t* prevFrame))

this->AbsoluteMeanOfMbs = this->AbsoluteMeanOfMbs +
abs(this->inter_coding->current_frame_mv->x)+
abs(this->inter_coding->current_frame_mv->y);
        //printf("%d \n",i_ME_Blocks_counted);
        //ypologismos maxX kai maxY se epipedo frame. dld metaksy twn macroblocks
        if (abs(this->inter_coding->current_frame_mv->x) > this->MaxXofMbs)
                this->MaxXofMbs=abs(this->inter_coding->current_frame_mv->x);
        if (abs(this->inter_coding->current_frame_mv->y) > this->MaxYofMbs)
                this->MaxYofMbs=abs(this->inter_coding->current_frame_mv->y);
```

Ο αλγόριθμος για τον υπολογισμό των πιθανών σκηνών έχει τα εξής βήματα και τις εξής δομές δεδομένων:

7.2 Δομές δεδομένων

Οι 2 κυριότερες δομές που αφορούν και τον αλγόριθμο υπολογισμού πιθανών σκηνών:

1) **Πίνακας (Μονοδιάστατος) SceneChangeXXXXMatrix[5]:** Κρατάει την εκάστοστε μετρική για τα πρώτα 5 διαστήματα σκηνών. Όταν αυτά συμπληρωθούν πρέπει να φτιαχνεί και ο αντίστοιχος MnemonicMatrix (αντίστοιχος εννοούμε για την εκάστοτε μετρική)

2) **Πίνακες MnemonicMatrix (διδιάστατος 5x5):** Για τον αλγόριθμο δυναμικού προγραμματισμού που χρησιμοποιούμε είναι ο κυριότερος πίνακας του αλγορίθμου. Κρατάει μεταξύ των πιθανών επικαλυπτόμενων και μη σκηνών την εκάστοτε μετρική και με την χρήση του τύπου σύνθεσης των μέσων όρων που έχουμε αναφέρει ή σύνθεσης των μεγίστων υπολογίζεται για την υποτιθέμενη σκηνή η αντίστοιχη μετρική. Πρακτικά είναι κάτω τριγωνικός γιατί γράφουμε από την κύρια διαγώνιο και κάτω μόνο στοιχεία. Τα υπόλοιπα δεν χρειάζονται στον αλγόριθμο που χρησιμοποιούμε

Αυτές είναι και οι κυριότερες δομές που χρησιμοποιούνται για τον αλγόριθμο. Από την $1^{η}$ δομή φτιάχνουμε 4 πίνακες, ένα για κάθε μετρική και ένα για το μέγεθος της κάθε

σκηνής σε frames. Από την 2η δομή ομοίως φτιάχνουμε πάλι άλλους 4 διδιάστατους πίνακες όπου ένας είναι για την κάθε μετρική και ένας κρατάει τα πλήθη (διδιάστατος) των frames στην σκηνή που αντιπροσωπεύει κάθε θέση του πίνακα.

Όλες οι δομές δεδομένων με τους τύπους τους όπως ορίστηκαν στο περιβάλλον που χρησιμοποιήθηκε:

```
long int AbsoluteMeanOfMbs;
int MaxXofMbs;       //same way as mesos oros for X,Y
int MaxYofMbs;
long int metric_counter; //thelei mhdenismo se kathe scene change
long int AbsoluteMeanofScene;
int MaxXofScene; //same way as mesos oros for X,Y
int MaxYofScene;
long int SceneFrameCounter;
long int SceneChangeMeanMatrix[5]; //krataei ta floats me kathe meso oro ana scene
changes, metaksy twn skinwn dld
long int MeanMnemonicMatrix[5][5]; //krataei ta floats me kathe meso oro ana
epikalyptomeno scene change, etsi wste na exei kai ta redundant kai ta paragwmena
scene changes gia to eventfiles
int MaxXMnemonicMatrix[5][5];            //antistoixa me meso oro gia maxX kai maxY
int MaxYMnemonicMatrix[5][5];
int SceneChangeMaxXMatrix[5];            //antistoixa me meso oro gia maxX kai maxY
int SceneChangeMaxYMatrix[5];
long int Frame_no_Matrix[5]; //krataei posa frames diarkei ena scene apo ta 5 pou
apothikevontai
long int Frame_no_Matrix2[5][5];
```

7.3 Αφηρημένη δομή αλγορίθμου

Βήμα1: Επεξεργασία current frame (Προετοιμασία βίντεο, Εισαγωγή βίντεο, μετατροπή βίντεο, έναρξη διαδικασιών encoding)

Βήμα2: Υπολογισμός μετρικών (Mean, MaxX,MaxY) σε επίπεδο macroblock

Βήμα3: Υπολογισμός μετρικών (Mean, MaxX,MaxY) σε επίπεδο frame

- Για κάθε macroblock κρατάω το άθροισμα των απολύτων για κάθε ευρεθέν διάνυσμα (AbsoluteMeanOfMbs)
- Συγκρίνω μεταξύ τους για τον υπολογισμό του MaxXofMbs και MaxYofMbs κρατάω με τον γνωστό αλγόριθμο αυτόν που πληρεί τα κριτήρια

Βήμα4: Υπολογισμός μετρικών σε επίπεδο Scene

- Για κάθε frame αθροίζω στο AbsoluteMeanofScene
- όταν γίνει η αλλαγή σκηνής διαιρώ με το μέγεθος της σκηνής για να πάρω το μέσο όρο
- κρατάω το ακέραιο μέρος του μέσου όρου (int)
- Για κάθε frame κάνω σύγκριση μεταξύ τους και έχω στα MaxXofScene και MaxYofScene τους μέγιστους X και Y για κάθε σκηνή

- Στον υπολογισμό κάθε scene τυπώνω στο fingerprint αρχείο μετά τα 5 πρώτα events(scene changes-τα 5 πρώτα μπαίνουν μαζί)

Βήμα5: Στους 5 θέσιους πίνακες SceneChangeXXXXMatrix για κάθε μετρική κρατάω για τις τελευταίες 5 σκηνές τις αντίστοιχες μετρικές σε κάθε έναν και στον Frame_no_Matrix τα πλήθη των frame ανά σκηνή.

- Βάζω για το current scene τις μετρικές που υπολογίστηκαν
- Αυξάνω το μετρητή (counter) για την επόμενη σκηνή
- Έπειτα κάνω το ίδιο μέχρι οι σκηνές να φτάσουν τις 5

Βήμα 6: Όταν φτιαχτούν οι 5 σκηνές ετοιμάζομαι να φτιάξω τον κάτω-τριγωνικό πίνακα XXXXMnemonicMatrix. Δυναμικός προγραμματισμός

- Στην διαγώνιο μπαίνουν οι τιμές του SceneChangeXXXXMatrix
- Στις υπόλοιπες τιμές κάτω από την διαγώνιο μπαίνουν τα αποτελέσματα των μετρικών υπολογίζοντας όλες τις πιθανές σκηνές από τα δεδομένα που έχουμε.
- Έτσι η πρώτη στήλη παίρνει την σύνθεση μετρικής από την $1^{η}$ με την $2^{η}$, την $1^{η}$ με $2^{η}$ και $3^{η}$, την $1^{η}$, $2^{η}$, $3^{η}$, την $1^{η}$, $2^{η}$, $3^{η}$,$4^{η}$. Αλλά ουσιαστικά κάθε φορά στην στήλη συνθέτει το i+1-στοιχείο του πίνακα SceneChangeXXXXMatrix με το i-στοιχείο της τρέχουσας στήλης του XXXXMnemonicMatrix.
- Ομοίως και για την $2^{η}$ στήλη με τις υπόλοιπες από τις σκηνές ($3^{η}$ και κάτω) καθώς και για την $3^{η}$ στήλη μέχρι την $4^{η}$.
- Η $5^{η}$ είναι το στοιχείο της διαγωνίου
- Στον πίνακα Frame_no_Matrix2 κρατάμε τα μεγέθη κάθε σκηνής όπως χρειάζονται για τους υπολογισμούς του XXXXMnemonicMatrix.

Βήμα 7: Στην $6^{η}$ σκηνή και τις επόμενες θα πρέπει να γίνει:

- Μεταφορά των στοιχείων σε κάθε SceneChangeXXXXMatrix μια θέση πάνω για να μπει το νέο στοιχείο στην τελευταία θέση.
- Έπειτα θα πρέπει να γίνει και μετακίνηση μια θέση πάνω και αριστερά στους πίνακες XXXXMnemonicMatrix(μνημονικός πίνακας) για να εισαχθεί η νέα γραμμή.

Βήμα 8: Ομοίως με το βήμα 6 εισάγεται η καινούρια γραμμή στον XXXXMnemonicMatrix (Μνημονικό πίνακα) και υπολογίζονται τα πλήθη για το Frame_no_Matrix2.

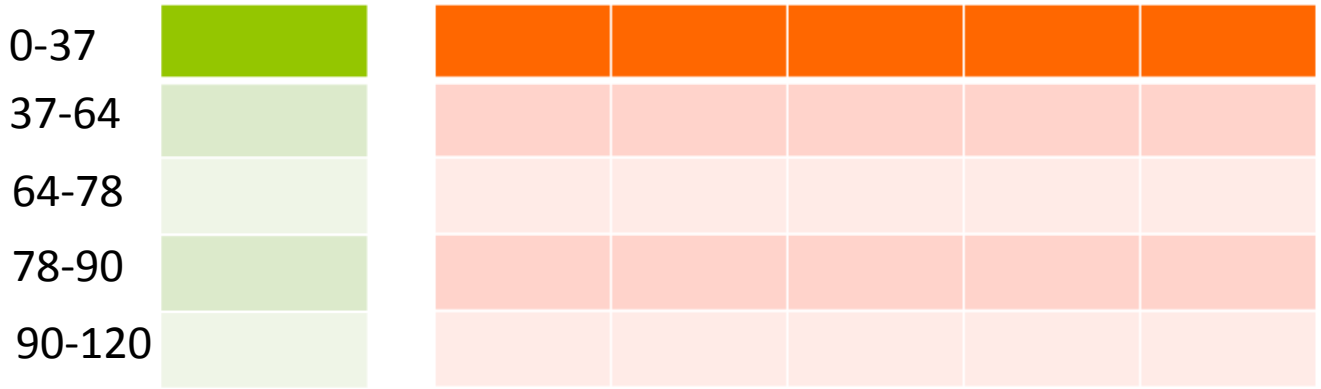
Βήμα 9: Επαναλαμβάνεται η παραπάνω διαδικασία μέχρι να μην υπάρχουν άλλα frames και σκηνές.

7.4 Παράδειγμα εκτέλεσης αλγορίθμου για πιθανές σκηνές

Συμπλήρωση μνημονικού πίνακα ανά στήλη με χρήση του SceneChangeXXXXMatrix

- Για τον Μέσο όρο θα δούμε το Χτίσιμο του μνημονικού πίνακα

Υποθετικές αλλαγές σκηνών: 0, 37, 64, 78, 90, 120, 145, 160



SceneChangeXXXXMatrix XXXXMnemonicMatrix

Σε κάθε βήμα γίνεται ο υπολογισμός κάθε στήλης, και στο new event γίνεται η μετακίνηση μια θέση αριστερά και πάνω των στοιχείων του πίνακας, για εισαγωγή της νέας γραμμής των πινάκων.

Βήμα 1

0-37	10	10	n/a	n/a	n/a	n/a
37-64	2	7	2	n/a	n/a	n/a
64-78	3	8	n/a	3	n/a	n/a
78-90	45	15	n/a	n/a	45	n/a
90-120	23	17	n/a	n/a	n/a	n/a

0-37	37	37	n/a	n/a	n/a	n/a
37-64	28	55	28	n/a	n/a	n/a
64-78	15	50	n/a	15	n/a	n/a
78-90	13	63	n/a	n/a	13	n/a
90-120	31	94	n/a	n/a	n/a	31

Βήμα 2

0-37	10	10	n/a	n/a	n/a	n/a
37-64	2	7	2	n/a	n/a	n/a
64-78	3	8	2	3	n/a	n/a
78-90	45	15	11	n/a	45	n/a
90-120	23	17	15	n/a	n/a	23

0-37	37	37	n/a	n/a	n/a	n/a
37-64	28	55	28	n/a	n/a	n/a
64-78	15	50	43	15	n/a	n/a
78-90	13	63	56	n/a	13	n/a
90-120	31	94	87	n/a	n/a	31

Βήμα 3

0-37	10	10	n/a	n/a	n/a	n/a
37-64	2	7	2	n/a	n/a	n/a
64-78	3	8	2	3	n/a	n/a
78-90	45	15	11	22	45	n/a
90-120	23	17	15	22	n/a	23

0-37	37	37	n/a	n/a	n/a	n/a
37-64	28	55	28	n/a	n/a	n/a
64-78	15	50	43	15	n/a	n/a
78-90	13	63	56	28	13	n/a
90-120	31	94	87	59	n/a	31

Βήμα 4

0-37	10
37-64	2
64-78	3
78-90	45
90-120	23

10	n/a	n/a	n/a	n/a
7	2	n/a	n/a	n/a
8	2	3	n/a	n/a
15	11	22	45	n/a
17	15	22	29	23

0-37	37
37-64	28
64-78	15
78-90	13
90-120	31

37	n/a	n/a	n/a	n/a
55	28	n/a	n/a	n/a
50	43	15	n/a	n/a
63	56	28	13	n/a
94	87	59	44	31

New event

~~0-37~~	~~10~~
37-64	2
64-78	3
78-90	45
90-120	23
120-145	34

2	n/a	n/a	n/a	n/a
2	3	n/a	n/a	n/a
11	22	45	n/a	n/a
15	22	29	23	n/a
…	…	…	…	34

~~0-37~~	~~37~~
37-64	28
64-78	15
78-90	13
90-120	31
120-145	26

28	n/a	n/a	n/a	28
43	15	n/a	n/a	43
56	28	13	n/a	56
87	59	44	31	87
…	…	…	…	26

7.5 Τμήματα κώδικα με εξήγηση και σχόλια

```
	//arxikopoihsseis twn metavlitwn metrikwn
	pDoc->MaxXofScene=0;
	pDoc->MaxYofScene=0;
	pDoc->MaxYofMbs=0;
	pDoc->MaxXofMbs=0;
	int counter;
	for (counter=0; counter<5;counter++){
		pDoc->SceneChangeMeanMatrix[counter]=0;
		pDoc->SceneChangeMaxXMatrix[counter]=0;
		pDoc->SceneChangeMaxYMatrix[counter]=0;
	}
	counter=0;
```

Αυτό που κάνουμε παραπάνω είναι η αρχικοποίηση των μετρικών που θα υπολογίζουμε παρακάτω στο πρόγραμμα και αφορούν το frame, και την σκηνή.

Ο πίνακας SceneChangeXXXXMatrix όπου XXXX (MaxX,MaxY, Mean) η μετρική την εκάστοτε φορά είναι ο πίνακας που αποθηκεύει την μετρική σε κάθε αλλαγή σκηνής με βάση τον υπολογισμό κάθε φορά από το κατώφλι. Αυτός είναι μια από τις βασικές δομές που χρησιμοποιούνται για να υπολογίσουμε τις πιθανές αλλαγές σκηνής για να αποφύγουμε τα σφάλματα. Σε αυτόν αποθηκεύονται οι τιμές για τις μετρικές τις 5 τελευταίες σκηνές. Από αυτές τις 5 σκηνές με το που έρθει η επόμενη σκηνή (η $6^{η}$) μπορούμε να αρχίσουμε να υπολογίζουμε τις επιθυμητές σκηνές.

```
	int FistTime = 1; //to vazoume arxika true, mexri na ginei to prwto ftiaksimo
tou MnemonicMatrix
```

Το FirstTime χρησιμοποιείται ως μεταβλητή Boolean για να ξέρουμε την πρώτη φορά που κατασκευάζουμε τον MnemonicMatrix. Έτσι δεν χρειάζεται να κάνουμε υπολογισμό ολόκληρου του πίνακα MnemonicMatrix (διδιάστατος) δεύτερη φορά, παρά μόνο της τελευταίας του γραμμής για περισσότερη ταχύτητα.

```
if (event_counter > 4) {
//FistTime=0;
pDoc->SceneChangeMeanMatrix[counter]=pDoc->AbsoluteMeanofScene;
pDoc->SceneChangeMaxXMatrix[counter]=pDoc->MaxXofScene;
pDoc->SceneChangeMaxYMatrix[counter]=pDoc->MaxYofScene;
pDoc->Frame_no_Matrix[counter]=pDoc->SceneFrameCounter;
counter++;
//construct SceneChangeMeanMatrix
//also construct Frame_no_Matrix
//shift tous pinakes tous kanoume sto ekto event
```

Σε αυτό το σημείο σημαίνει ότι καταφέραμε και τα events (scene changes) είναι τόσα όσο το πιθανό σφάλμα που έχουμε ορίσει οπότε πρέπει να αρχίσει η κατάστρωση των πιθανών scene changes. Προστίθεται επίσης η τελευταία μετρική σκηνής που υπολογίστηκε μόλις πριν από τις συναρτήσεις που δουλεύουν ανά macroblock και ανά frame.

```
if (event_counter>5){
for (counter=0;counter<4;counter++){
pDoc->SceneChangeMeanMatrix[counter]= pDoc->SceneChangeMeanMatrix[counter+1];
pDoc->Frame_no_Matrix[counter]=pDoc->Frame_no_Matrix[counter+1];
pDoc->SceneChangeMaxXMatrix[counter]= pDoc->SceneChangeMaxXMatrix[counter+1];
pDoc->SceneChangeMaxYMatrix[counter]= pDoc->SceneChangeMaxYMatrix[counter+1];
}
pDoc->SceneChangeMeanMatrix[counter]=pDoc->AbsoluteMeanofScene;
pDoc->SceneChangeMaxXMatrix[counter]=pDoc->MaxXofScene;
pDoc->SceneChangeMaxYMatrix[counter]=pDoc->MaxYofScene;
pDoc->Frame_no_Matrix[counter]=pDoc->SceneFrameCounter;
```

Εδώ κάνουμε ολίσθηση των τιμών του πίνακα και τοποθετούμε τις τελευταίες τιμές στο τελευταίο στοιχείο του κάθε πίνακα SceneChangeXXXMatrix με την τελευταία έγκυρη τιμή.

Με αυτόν τον τρόπο θα μπορούμε να κατασκευάσουμε τον πίνακα MnemonicMatrix για κάθε μετρική.

Η αρχική κατασκευή του MnemonicMatrix με την βοήθεια της Boolean first time:

```
if (FistTime) {
FistTime=0;
for (int i=0;i<5;i++){
//h diagwnios pairnei tis times apo ta etoima metric scene changes
pDoc->MeanMnemonicMatrix[i][i]= pDoc->SceneChangeMeanMatrix[i];
pDoc->MaxXMnemonicMatrix[i][i]=pDoc->SceneChangeMaxXMatrix[i];
pDoc->MaxYMnemonicMatrix[i][i]=pDoc->SceneChangeMaxYMatrix[i];
//h diagwnios pairnei tis times apo ta etoima synola frames pou periexei kathe scene--> xreiazetai
gia tous mesous orous
pDoc->Frame_no_Matrix2[i][i]= pDoc->Frame_no_Matrix[i];
}
```

Παραπάνω γίνεται το εξής, η διαγώνιος του πίνακα MnemonicMatrix παίρνει τις τιμές των SceneChangeXXXXMatrix. Για κάθε μετρική ενημερώνεται ο αντίστοιχος διδιάστατος πίνακας μετρικών φυσικά.

Στην συνέχεια έχουμε:

```
for (int j=0;j<5;j++){
for(int i=j+1;i<5;i++){
//tha prepei na exw kai mnemonic gia to plithos matrix...tha xreiastei an thelw na eimai etoimos
apo edw pou etsi opws to exw kanei apo edw prepei na eimai afou den mplekw me thn lista
//meta prepei na ftiaksoume tis ypoloipes times gia prwth fora tha einai me vassh ton typo tou
mesou orou tou miktou mo= (plithos1/plithos)*MO1 +(plithos2/plithos)*MO2
pDoc->Frame_no_Matrix2[i][j]=pDoc->Frame_no_Matrix[i]+pDoc->Frame_no_Matrix2[i-1][j];
pDoc->MeanMnemonicMatrix[i][j]=(pDoc->SceneChangeMeanMatrix[i])*(pDoc->Frame_no_Matrix[i])/
(pDoc->Frame_no_Matrix2[i][j])+(pDoc->MeanMnemonicMatrix[i-1][j])*
```

```
(pDoc->Frame_no_Matrix2[i-1][j])/pDoc->Frame_no_Matrix2[i][j];
pDoc->MaxXMnemonicMatrix[i][j]=pDoc->MaxXMnemonicMatrix[i-1][j];
pDoc->MaxYMnemonicMatrix[i][j]=pDoc->MaxYMnemonicMatrix[i-1][j];
if (pDoc->MaxXMnemonicMatrix[i-1][j]<pDoc->SceneChangeMaxXMatrix[i])
pDoc->MaxXMnemonicMatrix[i][j]=pDoc->SceneChangeMaxXMatrix[i];
if (pDoc->MaxYMnemonicMatrix[i-1][j]<pDoc->SceneChangeMaxYMatrix[i])
pDoc->MaxYMnemonicMatrix[i][j]=pDoc->SceneChangeMaxYMatrix[i];
```

Εδώ παραπάνω γίνεται ο υπολογισμός του XXXXMnemonicMatrix για κάθε μετρική και τον υπολογισμό των στοιχείων εντός του πίνακα στο κάτω τριγωνικό κομμάτι, αφού όπως αναφέραμε ο πίνακας είναι κάτω τριγωνικός.

Ο υπολογισμός για τον μέσο όρο γίνεται με τον τύπο σύνθεσης 2 μέσων όρων με βαρύτητα για κάθε ένα το πλήθος και συνολικό πλήθος το άθροισμα πλήθους frames των 2 σκηνών. Πρακτικά κάθε στοιχείο του μνημονικού πίνακα/μνημονικών πινάκων υπολογίζεται με κατάλληλη σύνθεση (σύγκριση για max, άθροιση και διαίρεση για μέσο όρο) του στοιχείου i του μνημονικού πίνακα, στην τρέχουσα στήλη, με το στοιχείο i-1 του SceneChangeXXXXMatrix.

```
for (int j=1;j<5;j++){
        for(int i=j;i<5;i++){
        pDoc->MeanMnemonicMatrix[i-1][j-1]=pDoc->MeanMnemonicMatrix[i][j];
        pDoc->MaxXMnemonicMatrix[i-1][j-1]=pDoc->MaxXMnemonicMatrix[i][j];
        pDoc->MaxYMnemonicMatrix[i-1][j-1]=pDoc->MaxYMnemonicMatrix[i][j];
        pDoc->Frame_no_Matrix2[i-1][j-1]=pDoc->Frame_no_Matrix2[i][j];
        }
}
```

Παραπάνω στον κώδικα χρησιμοποιούμε ένα τρικ μετατόπισης ώστε να μετακινήσουμε μια θέση πάνω και μια θέση αριστερά τα στοιχεία του μνημονικού πίνακα(XXXXMnemonicMatrix). Με αυτόν τον τρόπο μπορούμε να καταφέρουμε να εισάγουμε την νέα γραμμή για τον πίνακα όταν θα έρθει το νέο event χωρίς να ξανακάνουμε τους υπολογισμούς από την αρχή και να εκμεταλλευτούμε και τις προηγούμενες πληροφορίες που είχαμε αποθηκεύσει. Πρακτικά ο μνημονικός πίνανας είναι σαν να έχουμε 5 παράλληλες κατακόρυφες λίστες να τρέχουν καθώς σχηματίζονται τα γεγονότα και το μοτίβο των πιθανών σκηνών με τις μετρικές τους, μόνο που δουλεύει ποιο ομαλά και γρήγορα από τις λίστες.

```
for (int j=0;j<4;j++){
pDoc->Frame_no_Matrix2[4][j]=pDoc->Frame_no_Matrix[4]+pDoc->Frame_no_Matrix2[3][j];
pDoc->MeanMnemonicMatrix[4][j]=(pDoc->SceneChangeMeanMatrix[4])*(pDoc->Frame_no_Matrix[4])/(pDoc-
>Frame_no_Matrix2[4][j])+(pDoc->MeanMnemonicMatrix[3][j])*(pDoc->Frame_no_Matrix2[3][j])/(pDoc-
>Frame_no_Matrix2[4][j]);
pDoc->MaxXMnemonicMatrix[4][j] = pDoc->MaxXMnemonicMatrix[3][j];
pDoc->MaxYMnemonicMatrix[4][j]   = pDoc->MaxYMnemonicMatrix[3][j];
if (pDoc->MaxXMnemonicMatrix[4][j]< pDoc->SceneChangeMaxXMatrix[4])
pDoc->MaxXMnemonicMatrix[4][j]=pDoc->SceneChangeMaxXMatrix[4];
if (pDoc->MaxYMnemonicMatrix[4][j]< pDoc->SceneChangeMaxYMatrix[4])
pDoc->MaxYMnemonicMatrix[4][j]=pDoc->SceneChangeMaxYMatrix[4];
}
pDoc->Frame_no_Matrix2[4][4]=pDoc->Frame_no_Matrix[4];
pDoc->MeanMnemonicMatrix[4][4]=pDoc->SceneChangeMeanMatrix[4];
pDoc->MaxXMnemonicMatrix[4][4]=pDoc->SceneChangeMaxXMatrix[4];
pDoc->MaxYMnemonicMatrix[4][4]=pDoc->SceneChangeMaxYMatrix[4];
```

Στο παραπάνω κομμάτι κώδικα υπάρχει η αναγκαιότητα να υπολογίσουμε την τελευταία γραμμή του μνημονικού πίνακα (XXXXMnemonicMatrix) και το τελευταίο στοιχείο (5,5) που είναι το $5^{ο}$ στην σειρά από το SceneChangeXXXXMatrix. Είναι το στοιχείο της διαγωνίου εξάλλου που πάντα περνάμε αυτούσιο από το μονοδιάστατο στον διδιάστατο πίνακα. Αυτό όταν τα events είναι πάνω από 5 και δεν είναι η πρώτη φορά που υπολογίζουμε τον πίνακα.

```
else { //when even_counter <= 4
pDoc->SceneChangeMeanMatrix[counter]=pDoc->AbsoluteMeanofScene;
pDoc->SceneChangeMaxXMatrix[counter]=pDoc->MaxXofScene;
pDoc->SceneChangeMaxYMatrix[counter]=pDoc->MaxYofScene;
pDoc->Frame_no_Matrix[counter]=pDoc->SceneFrameCounter;
counter++;
EventFileFillInitial(pDoc);
}
```

Παραπάνω βλέπουμε τον κώδικα που εκτελείται για να αποθηκεύει τις μετρικές στους μονοδιάστατους πίνακες SceneChangeXXXXMatrix. Αυτό γίνεται όσο το event_counter είναι μικρότερο από 5 μόλις φτάσει 5 πρέπει να ξεκινήσουν να γίνονται και οι υπόλοιποι υπολογισμοί.

```
if (scene_change)
{
scene_change = 0;
pDoc->AbsoluteMeanofScene = 0;
pDoc->MaxXofScene=0;
pDoc->MaxYofScene=0;
event_counter++;
}
```

Στην αλλαγή σκηνής πρέπει να επιστρέψει η μεταβλητή στην αρχική κατάσταση όπου δεν έχει αναγνωριστεί σκηνή και να μηδενιστούν οι μετρικές ανά σκηνή. Με αυτόν τον τρόπο δεν θα έχουμε υπερχειλίσεις και θα έχουμε σωστές μετρήσεις.

8. Παράδειγμα από fingerprint file

Κατά την εκτέλεση του προγράμματος για ένα συγκεκριμένο βίντεο οι πρώτες 3 σελίδες του αρχείου αποτυπώματος (fingerprint) που αποτελούν την ταυτότητα του βίντεο φαίνονται παρακάτω. Η πρώτη γραμμή που εισάγεται στο αρχείο ξεκινάει από το 0 και φτάνει ως το τελευταίο Α της τέταρτης γραμμής. Οι πρώτες γραμμές έχουν χωριστεί χρωματικά για να γίνει κατανοητό πως αποτελεί η κάθε μια οντότητα. Κάθε στοιχείο χωρίζεται από ένα tab με το άλλο και έχει την δική του σημασία. Οι μετρικές μας Mean, MaxX, MaxY βρίσκονται από τα αριστερά μετρώντας στην 8η στήλη, 9η στήλη και 10η στήλη αντίστοιχα. Επίσης η 3η και η 4η στήλη ορίζουν την πιθανή σκηνή στην οποία και καταγράφεται το αποτύπωμα των μετρικών και των υπόλοιπων στοιχείων για την σκηνή.

```
0	52	0	309	309	0	5	0	0	17	-1	-1
	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

0	52	0	310	310	1	10	0	3	17	-1	-1
	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAx

0	52	0	358	358	2	15	0	3	17	-1	-1
	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	254;258;263;265;

0	52	0	359	359	3	20	0	5	17	-1	-1
	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

0	52	0	411	411	4	25	0	5	17	-1	-1
	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	254;257;

0	52	309	310	1	0	10	0	3	3	-6	-2


0	52	309	358	49	1	15	0	3	3	-6	-2
	xDebAAAADDDAAbbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	0;3;8;10;

0	52	309	359	50	2	20	0	5	7	-6	-2
	xDebAAAADDDAAbbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxE	0;3;8;10;

0	52	309	411	102	3	25	0	5	7	-6	-2
	xDebAAAADDDAAbbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxExDEbAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAA
AAAAAAAAAAAAAAAEE	0;3;8;10;49;54;
```

0 52 309 454 145 4 30 0 5 7 -6 -2
xDebAAAADDDAAbbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxExDEbAAAAAAAAAAAAAAAAAAAAAAbAAAAAAAAA
AAAAAAAAAAAAAAAEExBAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 0;3;8;10;49;54;102;105;

0 52 310 358 48 0 15 0 3 3 -2 -4
DebAAAADDDAAbbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 0;2;7;9;

0 52 310 359 49 1 20 0 5 7 -2 -4
DebAAAADDDAAbbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxE 0;2;7;9;

0 52 310 411 101 2 25 0 5 7 -2 -4
DebAAAADDDAAbbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxExDEbAAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAA
AAAAAAAAAAAAAAEE 0;2;7;9;48;53;

0 52 310 454 144 3 30 0 5 7 -2 -4
DebAAAADDDAAbbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxExDEbAAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAA
AAAAAAAAAAAAAAEExBAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 0;2;7;9;48;53;101;104;

0 52 310 455 145 4 35 0 5 7 -2 -4
DebAAAADDDAAbbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxExDEbAAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAA
AAAAAAAAAAAAAAEExBAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxE 0;2;7;9;48;53;101;104;

0 52 358 359 1 0 20 0 5 7 -5 0 E

0 52 358 411 53 1 25 0 5 7 -5 0
ExDEbAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAAAAAAAAAAAAAAAAEE 0;4;

0 52 358 454 96 2 30 0 5 7 -5 0
ExDEbAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAAAAAAAAAAAAAAAAAEExBAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAA 0;4;52;55;

0 52 358 455 97 3 35 0 5 7 -5 0
ExDEbAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAAAAAAAAAAAAAAAAAEExBAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAxE 0;4;52;55;

0 52 358 456 98 4 40 0 5 7 -5 0
ExDEbAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAAAAAAAAAAAAAAAAAEExBAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAxExe 0;4;52;55;97;100;

0 52 359 411 52 0 25 0 5 6 -2 -2
DEbAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAAAAAAAAAAAAAAAAEE 0;2;

0 52 359 454 95 1 30 0 5 6 -2 -2
DEbAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAAAAAAAAAAAAAAAAAEExBAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAA 0;2;50;53;

0 52 359 455 96 2 35 0 5 6 -2 -2
DEbAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAAAAAAAAAAAAAAAAAEExBAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAxE 0;2;50;53;

0 52 359 456 97 3 40 0 5 6 -2 -2
DEbAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAAAAAAAAAAAAAAAAAEExBAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAxExe 0;2;50;53;95;98;

0 52 359 537 178 4 45 0 17 21 -2 -2
DEbAAAAAAAAAAAAAAAAAAAAbAAAAAAAAAAAAAAAAAAAAAAAAAAEExBAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAxExexCAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0;2;50;53;95;100;104;109;

0 52 411 454 43 0 30 0 3 3 -1 -3
BAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

0 52 411 455 44 1 35 0 3 3 -1 -3
BAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxE

0 52 411 456 45 2 40 0 3 3 -1 -3
BAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxExe 42;45;

0 52 411 537 126 3 45 0 17 21 -1 -3
BAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxExexCAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 42;47;51;56;

0 52 411 573 162 4 50 0 17 21 -1 -3
BAAAACdAAbAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxExexCAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxeAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 42;47;51;56;

0 52 454 455 1 0 35 0 1 1 -3 0 E

0 52 454 456 2 1 40 0 1 1 -3 0
Exe 0;2;

0 52 454 537 83 2 45 0 17 21 -3 0
ExexCAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0;4;8;13;

0 52 454 573 119 3 50 1 17 21 -3 0
ExexCAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxeAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 0;4;8;13;

0 52 454 614 160 4 55 1 17 21 -3 0
ExexCAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxeAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 0;4;8;13;

0 52 455 456 1 0 40 0 0 0 -2 -2 e

0 52 455 537 82 1 45 0 17 21 -2 -2
exCAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0;2;6;11;

0 52 455 573 118 2 50 1 17 21 -2 -2
exCAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxeAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA0;2;6;11;

0 52 455 614 159 3 55 1 17 21 -2 -2
exCAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxeAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 0;2;6;11;

0 52 455 653 198 4 60 6 64 64 -2 -2
exCAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxeAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxBCDDedcBBBBAAAAAAAbBbb
BBBBBCBBCCdcCccb 0;2;6;11;159;170;178;197;

0 52 456 537 81 0 45 0 17 21 -2 -4
CAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
4;9;

0 52 456 573 117 1 50 1 17 21 -2 -4
CAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxeAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA 4;9;

0 52 456 614 158 2 55 1 17 21 -2 -4
CAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxeAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAxAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 4;9;

0 52 456 653 197 3 60 6 64 64 -2 -4
CAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxeAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAxAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxBCDDedcBBBAAAAAAAbBbbBBB
BBCBBCCdcCccb 4;9;157;168;176;195;

0 52 456 689 233 4 65 5 64 64 -2 -4
CAAAbeEBCBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxeAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAxAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxBCDDedcBBBAAAAAAAbBbbBBB
BBCBBCCdcCccbxAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 4;9;157;168;176;196;

Το αρχείο fingerprint συνεχίζεται αλλά για λόγους οικονομίας και ουσίας δεν έχει νόημα να το παρέχουμε ολόκληρο εδώ γιατί ξεφεύγει από την εξήγηση του μηχανισμού λειτουργίας και καταγραφής του αλγορίθμου. Η ουσία και ο τρόπος λειτουργίας και καταγραφής είναι πανομοιότυπος οπότε έχει νόημα μονάχα η επίδειξη του μηχανισμού.

9. Τυπικές Εφαρμογές

Οι εφαρμογές μιας κομψής και ισχυρής μηχανής αναγνώρισης κίνησης σε βίντεο δεν περιορίζονται τόσο από τις δυνατότητες του μοντέλου της και του τρόπου υλοποίησης όσο από τις εφαρμογές τις ίδιες και από την φαντασία που έχει ο κάθε μηχανικός. Μερικές οφθαλμοφανείς κατηγορίες εφαρμογών φαίνονται παρακάτω με σαφή παραδείγματα τους.

9.1 Γρήγορη Αναγνώριση σκηνών σε βίντεο με μεγάλη ακρίβεια

Με τις μεθόδους που χρησιμοποιούνται για αναγνώριση σκηνών με πολύ μεγάλη ταχύτητα και ακρίβεια μπορεί κανείς να έχει με key frames που εξάγονται ως jpeg τις χαρακτηριστικές εικόνες της κάθε σκηνής. Ο αλγόριθμος είναι γρήγορος και απλός και βασίζεται στην .H264 κωδικοποίηση που χρησιμοποιούνται ευρέως. Μπορεί κανείς να φτιάξει είτε για το σινεμά είτε για μια αναπαράσταση ενός σημαντικού γεγονότος χαρακτηριστικές εικόνες που περιγράφουν περιληπτικά το γεγονός. Επίσης σε ένα απλό βίντεο μπορεί να γίνει και χαρακτηρισμός της ταχύτητας εξέλιξης κάθε σκηνής και των αλλαγών που έχει από το μέγιστο και τον μέσο όρο κίνησης.

9.2 Στατιστικός Διαγνωστικός έλεγχος μηχανής που έχει κίνηση (από έλεγχο μετρικών)

Μια μηχανή που εκτελεί κάποια κίνηση, συνήθως υποκύπτει στους νόμους της στατιστικής ανάλυσης σχετικά με την ποιότητα παροχής των υπηρεσιών της. Η επαναλαμβανόμενη κίνηση της έχει μεγέθη που απαντούν σε κάποιο μέσο όρο σε κάποιο μέγιστο και σε αποκλίσεις που αν ξεφύγουν από κάποια κατανομή υποδεικνύουν ότι αυτή η μηχανή μπορεί να έχει χαλάσει ή να θέλει επισκευή. Η χρήση ελέγχου με την αναγνώριση κίνησης σε βίντεο που κάνουμε μπορεί να προβλέψει αυτήν την ανάγκη για επισκευή της μηχανής.

Αυτό μπορεί να επιτευχθεί, μέσα από την απόκλιση των μετρικών που υπολογίζουμε συνυπολογίζοντας τις τιμές των μετρικών όταν η μηχανή λειτουργούσε σωστά ή επιθυμητά ή μόλις είχε βγει από το εργοστάσιο και άρα ήταν καινούρια.

Παραδείγματα τέτοιων χώρων με μηχανήματα ακριβά ή και ακριβείας είναι οι βιομηχανικοί και βιοτεχνικοί χώροι που περιέχουν μηχανήματα τύπου ρομποτικού βραχίονα, βιομηχανικά ρομπότ και ταινίες παραγωγής.

9.3 Έλεγχος ασφαλείας

Εκτός από τις βιομηχανικές εφαρμογές υπάρχουν και εφαρμογές κοινωνικής ασφάλειας που μπορεί να χρησιμοποιηθεί ο αλγόριθμος που αναπτύξαμε με ιδιαίτερη χρήση των μετρικών που υπολογίζονται. Ένα παράδειγμα είναι η κίνηση στις κυλιόμενες σκάλες που απόκλιση από το σύνηθες μπορεί να σημαίνει ότι κάτι συνέβη σε κάποιο χρήστη τους που μπορεί να είναι κάποιος ηλικιωμένος ή κάποιος άνθρωπος που χτύπησε. Σε κάθε περίπτωση μπορεί να ενημερωθεί για την ανωμαλία κάποιος υπεύθυνος και έγκαιρα να επιληφθεί η κατάσταση.

Ταυτόχρονα σε διαδρόμους κυκλοφορίας μπορεί να υπάρχει παρόμοια χρήση ασφάλειας για την κίνηση που υπάρχει και παράλληλη ενημέρωση. Μια κλασσική χρήση είναι η χρήση με κάμερες ασφαλείας όπου η αναγνώριση κίνησης και του τύπου της κίνησης που γίνεται μπορεί να σημαίνει κλοπή ή απομάκρυνση αντικειμένου φύλαξης από ένα χώρο και αυτό να συνεπάγεται άμεση ενεργοποίηση αυτοματισμών ασφαλείας και αρμόδιου προσωπικού.

10. Σύντομες τεχνικές λεπτομέρειες

10.1 Γλώσσα-Περιβάλλον Ανάπτυξης

Χρησιμοποιήθηκε ως περιβάλλον ανάπτυξης το Microsoft Visual Studio 2008 και στην συνέχεια έγινε και migration στα καινούρια Visual Studio της ίδιας εταιρείας με ελάχιστες διαφορές που διαπιστώθηκαν με αναζήτηση στα manual της εταιρίας και στο διαδίκτυο. Η Γλώσσα που επιλέχτηκε ήταν η Visual C++ .Net με επιλογή του Console project που έχει ως επιλογή για τα project η MS. Το Microsoft Visual Studio είναι ένα Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE) που βοηθάει ιδιαίτερα την ανάπτυξη εφαρμογών σε περιβάλλον Windows.

10.2 Βιβλιοθήκες

Χρησιμοποιήθηκαν κλασσικές βιβλιοθήκες που χρησιμοποιούνται για τα μαθηματικά math.h, η open source βιβλιοθήκη για μετατροπή βίντεο και αναφορά στα βίντεο με πάρα πολλές δυνατότητες ffmpeg και η βιβλιοθήκη standard input και standard output

της visual c++. Αυτές οι βιβλιοθήκες μπορούν να τρέξουν σε κάθε λειτουργικό σύστημα χωρίς περιορισμούς πέραν από αυτούς που δίνει το λειτουργικό και το hardware.

10.3 Φορητότητα

Ο κώδικας που γράφτηκε μπορεί με ελάχιστες αλλαγές στα header file να τρέξει και να κάνει compile αφού μιλάμε για εφαρμογή κονσόλας (console application) τόσο σε linux όσο και οποιοδήποτε άλλο unix λειτουργικό που έχει συμβατότητα με τις βασικές βιβλιοθήκες της c, την ffmpeg και το standard input and output.

11. Βιβλιογραφία

1) «H.264 and MPEG-4 Video Compression», Iain E. G. Richardson, Wiley Edition

2) «Implementing H.264 video compression algorithms on a software configurable processor**»**
(http://www.embedded.com/design/mcus-processors-and-socs/4006615/Implementing-H-264-video-compression-algorithms-on-a-software-configurable-processor**)**

3) «Enhanced Mode Selection Algorithm for H.264 encoder for Application in Low Computational power devices»
(http://arxiv.org/ftp/arxiv/papers/0909/0909.0245.pdf)

4) «H.264/MPEG-4 AVC» (http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC)

5) Motion Estimation and Intra Frame Prediction in H.264/AVC Encoder, Rahul Vanam, University of Washington,
(http://www.cs.washington.edu/education/courses/csep590a/07au/lectures/rahullarge.pdf**)**