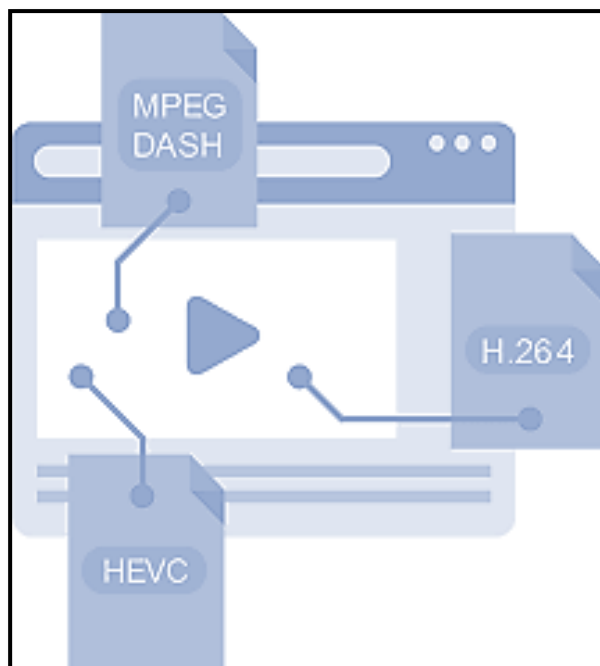




**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**«Πειραματική αξιολόγηση ποιότητας μετάδοσης  
Video, με χρήση προτύπου MPEG DASH».**



**Όνοματεπώνυμο: Φράγκου Φωτεινή**

**A.M. 2013034**

**Επιβλέπων καθηγητής: Μποζαντζής Βασίλειος**

**ΠΕΡΙΕΧΟΜΕΝΑ****ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> Εισαγωγή.**

1.1.Εισαγωγή.	Σελ. 6
1.2.Μετάδοση Κινούμενης Εικόνας-Video Streaming.	Σελ. 7
1.3.Εκτίμηση Ποιότητας Ψηφιακών Μέσων.	Σελ. 9
1.4.Ποιότητα Εικόνας.	Σελ.10
1.5.Προσεγγίσεις Μετρήσεων Ποιότητας Εικόνας.	Σελ.12
1.5.1. Αλγόριθμοι Πλήρους Αναφοράς.	Σελ.12
1.5.2. Αλγόριθμοι Μη Αναφοράς.	Σελ.13
1.5.3. Αλγόριθμοι Μειωμένης Αναφοράς.	Σελ.14

**ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> Μέθοδοι Εκτίμησης Ποιότητας Εικόνων.**

2.1. Μέσος Τετραγωνικού Σφάλματος(MSE).	Σελ.17
2.2. Λόγος Σήματος Προς Θόρυβο(PSNR).	Σελ.17
2.3. Καθολικός Δείκτης Ποιότητας Εικόνας(UIQI).	Σελ.17
2.4. Διαρθρωτικός Δείκτης Μέτρησης Ομοιότητας(SSIM).	Σελ.19
2.5. Πληροφορία Οπτικής Πιστότητας(VIF).	Σελ.19
2.6. Κλίση Απόκλιση Ομοιότητας Μεγέθους(GMSO).	Σελ.20
2.7. Ποιότητα Εμπειρίας Χρήστη(Quality of Experience).	Σελ.21
2.8. Ποιότητα Υπηρεσίας QoS (Quality of Service).	Σελ.22
2.9. Βαθμός Υπηρεσίας GoS και Ποιότητα Προσαρμοστικότητας QoR	Σελ.23
2.10.Βασικές Απαιτήσεις του Χρήστη.	Σελ.25

**ΚΕΦΑΛΑΙΟ 3<sup>ο</sup> Ψηφιακή Επεξεργασία και Ανάλυση Εικόνας.**

3.1. Η Ανάγκη για Ψηφιακή Συμπύεση Δεδομένων.	Σελ.27
3.2. Λόγος Συμπύεσης-Ρυθμός Συμπύεσης.	Σελ.27
3.3. Συμπύεση Δεδομένων.	Σελ.27
3.3.1.Παράγοντες που σχετίζονται με τη Συμπύεση.	Σελ.28
3.4. Γενικές Αρχές Συμπύεσης Video.	Σελ.29
3.4.1.Ποιοτική Ανοχή.	Σελ.29
3.4.2.Πλεονασμός (Redundancy).	Σελ.29
3.5. Δειγματοληψία.	Σελ.30
3.6. Κβαντοποίηση.	Σελ.30
3.7. Κωδικοποίηση.	Σελ.31
3.8. Γνωριμία με την Ψηφιακή Εικόνα.	Σελ.31
3.8.1.Είδη Ψηφιακών Εικόνων.	Σελ.31
3.9. Ανάλυση και Επεξεργασία Ψηφιακής Εικόνας.	Σελ.32
3.9.1.Βήματα για την Ανάλυση της Ψηφιακής Εικόνας.	Σελ.32
3.10. Περί Μετασχηματισμών Σημάτων Διακριτού Χρόνου.	Σελ.36
3.11. Ο Διακριτός Μετασχηματισμός Συνημιτόνου.	Σελ.36
3.11.1. Κωδικοποίηση DCT.	Σελ.37
3.11.2. Δισδιάστατος Διακριτός Μετασχηματισμός Συνημιτόνου(2d-DCT).	Σελ.37

**Κεφάλαιο 4<sup>ο</sup> Το Πρότυπο Συμπύεσης MPEG-4.**

4.1. Εισαγωγή.	Σελ.40
4.2. Ανατομία Σήματος MPEG	Σελ.40
4.3. Βασικοί Τρόποι Συμπύεσης MPEG.	Σελ.41
4.4. Είδη Πλαισίων στο MPEG	Σελ.42
4.5. Ομάδα Πλαισίων (Group of Pictures).	Σελ.44
4.5.1. Κομμάτια (Slices).	Σελ.44
4.5.2. Τμήματα-Μακροτμήματα (Blocks-Macroblocks).	Σελ.45

**ΠΕΡΙΕΧΟΜΕΝΑ****Κεφάλαιο 4<sup>ο</sup> Το Πρότυπο Συμπίεσης MPEG-4.**

4.6. Διαδικασία Συμπίεσης κατά MPEG.	Σελ.45
4.6.1. Το Πρότυπο Κωδικοποίησης MPEG-1.	Σελ.46
4.6.2. Το Πρότυπο Κωδικοποίησης MPEG-2.	Σελ.46
4.6.2.1. Οργάνωση Προτύπου.	Σελ.47
4.6.2.2. Επίπεδα και Προφίλ του MPEG-2.	Σελ.47
4.6.2.3. Κλιμακωσιμότητα στο MPEG-2.	Σελ.48
4.6.2.4. Ατέλειες του MPEG-2.	Σελ.48
4.6.3. Το Πρότυπο Κωδικοποίησης MPEG-4.	Σελ.49
4.6.3.1. Επανάσταση στη Λειτουργικότητα.	Σελ.50
4.6.3.2. Αποδοχή του MPEG-4.	Σελ.51
4.6.3.3. Προφίλ και Επίπεδα του MPEG-4.	Σελ.51
4.6.3.4. Τα Βασικά Μέρη του MPEG-4.	Σελ.51
4.6.3.5. Πλεονεκτήματα του MPEG-4.	Σελ.52

**Κεφάλαιο 5<sup>ο</sup> Πειραματικό Μέρος και Εφαρμογές.**

5.1.Εισαγωγή στο Matlab.	Σελ.54
5.2. Χρήση Παραδειγμάτων και Επεξήγησή τους.	Σελ.54
5.3. Ανάλυση Παραδειγμάτων και Εφαρμογές στην Καθημερινότητα.	Σελ.55

**Βιβλιογραφία**

Πηγές	Σελ. 74
-------	---------

**ΕΥΧΑΡΙΣΤΙΕΣ**

Με την περάτωση της πτυχιακής μου εργασίας, αρχικά, θα ήθελα να ευχαριστήσω τους επιβλέποντες καθηγητές μου, κύριο Σάββα Αργυρόπουλο, και κύριο Βασίλειο Μποζαντζή, για τις γνώσεις που μου παρείχαν, την ευκαιρία που μου έδωσαν να ασχοληθώ με αυτή την εργασία, αλλά και την πολύτιμη συνεργασία και καθοδήγησή τους, ως προς την υλοποίησή της.

Παράλληλα, θέλω να ευχαριστήσω και τους καθηγητές του τμήματός μου, για τις γνώσεις που μου παρείχαν, αλλά και την άριστη συνεργασία που μου προσέφεραν.

Τέλος, ένα μεγάλο ευχαριστώ στην οικογένειά μου, τόσο για την υλική, όσο και για την ψυχολογική στήριξη που μου προσέφεραν, κατά τη διάρκεια των σπουδών μου.

**ΠΕΡΙΛΗΨΗ**

Στις μέρες μας, είναι κοινώς αποδεκτό πως η ψηφιακή τεχνολογία έχει αντικαταστήσει την αναλογική, με μοναδικά κίνητρα, την καλύτερη και γρηγορότερη μετάδοση σήματος. Παρόλα αυτά όμως η παρουσία λαθών στην εικόνα είναι ακόμα εφικτή, είτε από λάθη του σήματος, είτε από παραμόρφωση είτε από συμπίεση. Στην παρούσα πτυχιακή εργασία, περιγράφονται και αναπτύσσονται τρόποι που μας δείχνουν την υποκειμενική και αντικειμενική αξιολόγηση της ποιότητας ψηφιακής εικόνας, κατά τη μετάδοσή της. Στο πρώτο κεφάλαιο, γίνεται μία εισαγωγή στη διαδικασία μετάδοσης της κινούμενης εικόνας, αλλά και στο πως γίνεται αντιληπτή η έννοια της ποιότητας. Παράλληλα, εμβαθύνουμε στο πως αξιολογούμε την ποιότητα, είτε με υποκειμενικό τρόπο, είτε με αντικειμενικό. Εν συνεχεία, στο δεύτερο κεφάλαιο, συνδυάζονται δύο ζητήματα. Το πρώτο, αφορά τις μετρικές της αντιλαμβανόμενης ποιότητας, σε μαθηματική έκφραση, ενώ το δεύτερο ζήτημα, αφορά τις απαιτήσεις του χρήστη, σε ποιότητα υπηρεσιών, συμβαλλομένων παραγόντων όπως η κατάσταση του δικτύου, το περιεχόμενο που μεταδίδεται ή ακόμη και το μέσο που μεταδίδουμε την πληροφορία. Συνάμα, στο τρίτο κεφάλαιο, γίνεται αναφορά στη ψηφιακή επεξεργασία αλλά και στην ανάλυση της εικόνας. Με λίγα λόγια, έννοιες όπως η κωδικοποίηση, η συμπίεση των δεδομένων μας αλλά και ο μετασχηματισμός του συνημιτόνου, υπογραμμίζονται και συνδέονται με το τέταρτο και τελευταίο θεωρητικό σκέλος της εργασίας, όπου αναλύονται τα στάδια εξέλιξης του προτύπου κωδικοποίησης Mpeg, αλλά και τα πλεονεκτήματά του έναντι των προηγούμενων του. Εν κατακλείδι, ολοκληρώνουμε με την παρουσίαση του πέμπτου κεφαλαίου, όπου παρουσιάζονται μία σειρά από video μετάδοσης κινούμενης εικόνας, σε περιβάλλον Matlab, στα οποία παρατηρείται ανίχνευση κινούμενης εικόνας και ανάλυσής της, ενώ ταυτόχρονα τονίζουμε και ορισμένες εφαρμογές στην καθημερινότητά μας, από τη χρήση της κινούμενης εικόνας Video, μέσω της ανίχνευσης.

**ABSTRACT**

Nowadays, it is commonly accepted that digital technology has replaced analogue, with an only unique motivation, better and faster signal transmission. However, the presence of errors in the image is still feasible, either by signal errors, by distortion or by compression. In this dissertation, we describe and develop ways that show us the subjective and objective evaluation of the quality of the digital image during its transmission. In the first chapter, there is an introduction to the process of transmission of the moving image, but also how the concept of quality is perceived. At the same time, we deepen our assessment of quality, either subjectively or objectively. Moreover, in the second chapter, two issues are combined. The first concerns the metrics of perceived quality in mathematical expression, while the second concerns the user's requirements, quality of service, contracting factors such as the state of the network, the content that is transmitted, or even the medium we transmit the information. At the same time, in the third chapter, reference is made to digital processing as well as to image analysis. Last, but not least, it can be argued that concepts such as coding, data compression, and cosine transformation are highlighted and linked to the fourth and final theoretical part of the paper, which analyzes the development stages of the Mpeg coding standard and its advantages over of its predecessors. In conclusion, we conclude with the presentation of the fifth chapter, which presents a series of motion picture video transmissions in a Matlab environment, which detects moving image and its analysis, while at the same time we emphasize some applications in our everyday life, from the use of the moving Video, through detection.

**Κεφάλαιο 1<sup>ο</sup>**  
**Εισαγωγή.**

**1.1. Εισαγωγή.**

Κατά τη διάρκεια των τελευταίων ετών, με της πρόοδο της τεχνολογίας και την αύξηση της ταχύτητας των δικτύων, έχει γίνει εφικτή η μετάδοση video σε υψηλές αναλύσεις, αλλά και η επικοινωνία με video μέσω φορητών συσκευών. Τόσο οι ανάγκες, όσο και οι απαιτήσεις των χρηστών, αυξάνονται με ραγδαία ταχύτητα, και αυτό γίνεται αντιληπτό όχι μόνο από τον αριθμό των χρηστών που πληθαίνει, αλλά και από την μέση χρήση δεδομένων που καταναλώνονται. Επιπλέον, η χρήση του διαδικτύου έχει επεκταθεί σε όλες τις πτυχές της ζωής του σύγχρονου ατόμου, από απλή ενημέρωση, διασκέδαση, αλλά και επικοινωνία. Επίσης, αποτελεί κύριο παράγοντα για ερευνητικούς και επαγγελματικούς σκοπούς, αφού η μετάδοση των πληροφοριών γίνεται άμεσα, γρήγορα και αποτελεσματικά. Η παροχή υπηρεσιών κινούμενης εικόνας σε πραγματικό χρόνο και η ταυτόχρονη πρόσβαση πολλαπλών χρηστών σε εξυπηρετητές πολυμέσων δημιουργεί πρόσθετες απαιτήσεις όσον αφορά στο χειρισμό της πληροφορίας στο δίκτυο. Η εξέλιξη της τεχνολογίας, μας δίνει τη δυνατότητα να έχουμε δίκτυα νέας γενιάς κάνοντας χρήση οπτικών ινών συνδυάζοντας έτσι τεράστιο εύρος ζώνης, καλύτερες ποιότητες σήματος, μεγαλύτερη ασφάλεια και ευκολότερη εγκατάσταση και συντήρηση. Με τις κατάλληλες μεθόδους κωδικοποίησης και τεχνικές πολυπλεξίας μας προσφέρονται τρόποι για ακόμα καλύτερη εκμετάλλευση του φυσικού υλικού και έτσι μπορούμε με μεταφέρουμε περισσότερη πληροφορία πάνω από το ίδιο κανάλι επικοινωνίας. Επιπρόσθετα, με νέες τεχνικές συμπίεσης δεδομένων εικόνας και ήχου, μειώνεται το μέγεθος του κάθε αρχείου και μας δίνεται τελικά η δυνατότητα να έχουμε ζωντανή μετάδοση βίντεο. Τέτοιες μεταδόσεις αξιοποιούνται τόσο για ενημέρωση, με χαρακτηριστικό παράδειγμα να αποτελούν τα on Line δελτία ειδήσεων, όσο και για αμφίδρομη επικοινωνία όπως video conference για επαγγελματικούς σκοπούς, ή ακόμη και για εφαρμογές τηλεϊατρικής. Όπως είναι αναμενόμενο, ο τελικός χρήστης θα θελήσει να έχει πρόσβαση στις παρεχόμενες υπηρεσίες ανεξάρτητα από το που βρίσκεται, ανεξάρτητα από την ώρα και ανεξάρτητα από τη συσκευή που χρησιμοποιεί. Επομένως, σημειώνεται ανάπτυξη τόσο στο τεχνικό κομμάτι των συσκευών ώστε να μπορέσουν να διαχειριστούν το περιεχόμενο, όσο και στην τεχνολογία των ασύρματων δικτύων, όπου με την εφαρμογή νέων τεχνικών ψηφιακής επεξεργασίας και διαμόρφωσης σήματος αυξάνεται ο ρυθμός μετάδοσης δεδομένων. Ανάλογα τον τρόπο μετάδοσης (ασύρματη – ενσύρματη) και το είδος της επικοινωνίας (μονόδρομη-αμφίδρομη), υπάρχουν διαφορετικοί τρόποι προσέγγισης και διαφορετικές δυσκολίες. Υπάρχουν όμως κάποιοι κοινόι παράγοντες που μπορούν να επηρεάσουν την τελική ποιότητα που αντιλαμβάνεται ο χρήστης. Εδώ θα πρέπει να υπογραμμίσουμε πως, εκτός από τις κλασσικές μετρήσεις που μπορεί να γίνουν για την κατάσταση του δικτύου, θα πρέπει να δοθεί μεγάλη προσοχή στην ανθρώπινη αντίληψη, αφού ο τελικός χρήστης είναι αυτός που θα κρίνει αν νιώθει ικανοποιημένος από τη χρήση της υπηρεσίας ή όχι. Η μεγάλη πρόκληση είναι στη διαχείριση των δυσκολιών που αναπόφευκτα θα προκύψουν κάποια στιγμή. Επί παραδείγματι, σε περιπτώσεις Live streaming μουσικής, ο χρήστης είναι πιθανόν να 'συγχωρέσει' μια πτώση στα kbps προκειμένου να έχει μια ομαλή μετάδοση. Από την άλλη, οι συνεχείς μικρές διακοπές θα καθιστούσαν την υπηρεσία μη ανεκτή. Όπως επίσης, σε περιπτώσεις κακής σύνδεσης με αμφίδρομη επικοινωνία μέσω βίντεο, είναι προτιμότερο να έχουμε απώλειες στην μετάδοση εικόνας παρά στη μετάδοση ήχου. Κατά καιρούς έχουν προταθεί διάφοροι τρόποι προσέγγισης, αλλά τον τελευταίο λόγο τον έχει πάντα ο τελικός χρήστης, αφού αυτός κρίνει πια τεχνολογία τον εξυπηρετεί περισσότερο, έχει χαμηλότερο κόστος και τελικά πια επικρατεί. Έτσι, δεν προκαλεί εντύπωση το γεγονός ότι έχει δοθεί μεγάλη βαρύτητα στην τελική εμπειρία του χρήστη (QoE: Quality of Experience) και στους τρόπους με τους οποίους μπορεί να εκτιμηθεί.

**1.2. Μετάδοση κινούμενης εικόνας - Video Streaming.**

Τα πολυμέσα χρησιμοποιούν εξαιρετικά υψηλούς ρυθμούς μεταφοράς δεδομένων και αυτό οφείλεται στη φύση των ηχητικών και οπτικών πληροφοριών. Το μάτι και το αυτί είναι σε θέση να επεξεργαστούν τεράστιες ποσότητες πληροφοριών ανά δευτερόλεπτο και για αυτό, πρέπει να



τροφοδοτούνται με αυτόν το ρυθμό ώστε η οπτικοακουστική εμπειρία να είναι αποδεκτή. Για να διατηρηθεί αυτός ο ρυθμός, τα δεδομένα πρέπει να παραδίδονται σε πραγματικό χρόνο. Η συνεχής αυτή ροή (streaming), μπορεί να χωριστεί σε δυο γενικές κατηγορίες:

- ❖ **On Demand Streaming:** Στην πρώτη περίπτωση ο δέκτης ζητά συγκεκριμένο περιεχόμενο από τον αποστολέα, π.χ. παρακολούθηση ενός video clip, ενώ η αποστολή θα γίνει αποκλειστικά για τη συγκεκριμένη συνεδρία.
- ❖ **Real Time Streaming:** Στην περίπτωση που έχουμε Real Time Streaming ο αποστολέας είναι αυτός που θα καθορίσει το περιεχόμενο και ο δέκτης απλά το αναπαράγει ενώ είναι δυνατή η ταυτόχρονη λήψη από πολλούς αποδέκτες, όπως για παράδειγμα στην περίπτωση που έχουμε Live iptv.

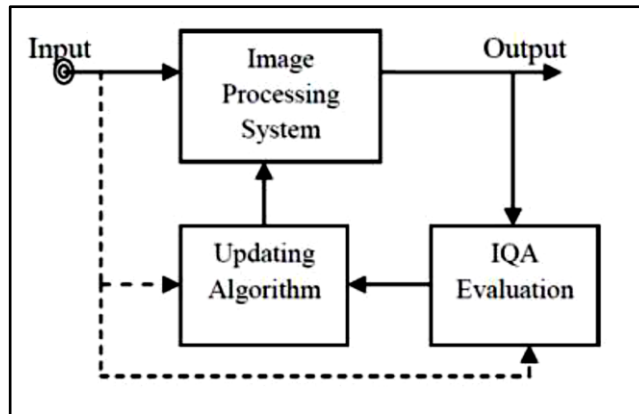
Παρόλα αυτά, και στις δύο περιπτώσεις, καθίσταται δυνατόν η μετάδοση να μπορεί να προσαρμοστεί στις υπάρχουσες συνθήκες, με αποτέλεσμα ο αποστολέας να ανταποκρίνεται στις δυνατότητες του δέκτη και να μπορεί να μειώσει τον ρυθμό μετάδοσης δεδομένων ανάλογα με την περίπτωση. Η διαφορά είναι ότι στην περίπτωση on demand, είναι πιθανόν να εφαρμοστεί αναγκαστική παύση της μετάδοσης για μικρό χρονικό διάστημα, φαινόμενο ενοχλητικό αλλά αποδεκτό από τον χρήστη αρκεί να μην συμβαίνει συχνά. Εν αντιθέσει με τη δεύτερη κατηγορία, όπου δεν υπάρχει αυτή η δυνατότητα, καθώς η μετάδοση δεν μπορεί να σταματήσει. Επομένως, είναι προτιμότερο να παραληφθούν κάποια πακέτα και ο χρήστης να το αντιληφθεί σαν 'σπάσιμο' της εικόνας. Για να αντιμετωπιστεί το φαινόμενο αυτό, γίνεται η χρήση buffer, μιας προσωρινής μνήμης από τον δέκτη όπου εκεί μπορεί και επεξεργάζεται συνθέτει και αποκωδικοποιεί τα λαμβανόμενα πακέτα. Έτσι ακόμα και στην περίπτωση Real Time Streaming, υπάρχει μια μικρή καθυστέρηση προκειμένου να έχουμε ομαλή ροή και φυσική κίνηση. Ωστόσο, πέρα από την κατάσταση του δικτύου και τις συνεχείς προσπάθειες για τη βελτίωσή του, με μοναδικό στόχο το μέγιστο δυνατό αποτέλεσμα, όσον αφορά το QoE, πρέπει να δοθεί έμφαση και στο αρχικό προς μετάδοση περιεχόμενο, καθώς και στο βέλτιστο τρόπο κωδικοποίησής του. Μερικοί από τους παράγοντες που χαρακτηρίζουν το αρχικό περιεχόμενο κινούμενης εικόνας είναι οι εξής:

- ❖ **Resolution:** Πρόκειται για την ανάλυση της κάθε εικόνας, στην περίπτωση βίντεο την ανάλυση του κάθε καρέ. Είναι το πλήθος των εικονοστοιχείων (pixels) που αποτελούν το εικονιζόμενο καρέ και εκφράζεται ως το γινόμενο του οριζόντιου (πλάτος) με τον κάθετο άξονα (ύψος). Μεγαλύτερη ανάλυση ισοδυναμεί με μεγαλύτερη διάστασης εικόνα. Εδώ θα πρέπει να γίνει διαχωρισμός ανάμεσα στην διάσταση της εικόνας και στην τελική αντιλαμβανόμενη ποιότητα. Με μεγαλύτερο μέγεθος καρέ είναι δυνατόν να αποτυπωθεί περισσότερη πληροφορία και να έχει άμεσο αντίκτυπο στην διαύγεια και λεπτομέρεια της εικόνας, αλλά αυτό εξαρτάται άμεσα και από το μέσο προβολής που έχουμε στη διάθεσή μας. Συνήθως οι αλγόριθμοι επεξεργασίας εικόνας διατρέχουν την επιφάνεια τμηματικά.
- ❖ **Frames Per Second (fps):** Είναι ο ρυθμός ανανέωσης της εικόνας σε κάθε δευτερόλεπτο, το πλήθος δηλαδή των καρέ που εναλλάσσονται. Μεγάλο fps έχει ως αποτέλεσμα την ομαλή ροή και σταθερή εικόνα βίντεο, ενώ χαμηλό fps είναι καθιστά την κίνηση μη φυσική. Ανάλογα με το περιεχόμενο έχουμε και διαφορετικές απαιτήσεις για το ελάχιστο fps. Σε περιπτώσεις έντονης κίνησης και γρήγορης εναλλαγής σκηνών, ο υψηλός ρυθμός ανανέωσης της εικόνας είναι απαραίτητος. Όπως είδαμε και προηγουμένως για την ανάλυση, και για το fps παίζει άμεσο ρόλο και η συσκευή προβολής αφού είναι ανούσιο να τροφοδοτείται με υψηλότερο fps από αυτό που τελικά μπορεί να αποδώσει.
- ❖ **Bit Rate:** Σε τελική ανάλυση, ο αριθμός των συνολικών pixels μιας εικόνας, σε συνδυασμό με τον ρυθμό ανανέωσης ανά δευτερόλεπτο μας δίνει το συνολικό πλήθος των δεδομένων του υλικού προς μετάδοση. Ανάλογα με τον τρόπο και τη μέθοδο κωδικοποίησης έχουμε και το τελικό bit rate, δηλαδή τον αριθμό των bit ανά δευτερόλεπτο.
- ❖ **Content:** Όπως έχουμε ήδη αναφέρει, το περιεχόμενο του βίντεο παίζει καθοριστικό ρόλο στις απαιτήσεις που έχουμε τόσο σε επίπεδο δικτύου, όσο και στο επίπεδο της συμπίεσης-κωδικοποίησης. Περιεχόμενο με γρήγορη κίνηση (πχ σκηνές με αθλήματα, δράση) απαιτεί υψηλότερο fps σε σχέση με βίντεο στατικού περιεχομένου (παρουσιάσεις slideshow, video conference).

Είναι κοινώς αποδεκτό, πως με τους κατάλληλους αλγόριθμους επεξεργασίας της κινούμενης εικόνας, μπορούμε να ελαχιστοποιήσουμε τον όγκο δεδομένων προς αποστολή χωρίς να έχουμε σημαντική μείωση στην αντιλαμβανόμενη ποιότητα. Επομένως, παρατηρείται τόσο μείωση των απαιτήσεων για το δίκτυο, όσο και κατ' επέκταση των αρνητικών παραγόντων του QoS που επηρεάζουν το QoE, με χαρακτηριστικό παράδειγμα το jitter ή ακόμη και το packet loss. Λαμβάνοντας υπόψη τα παραπάνω, έχουν παρουσιαστεί μία σειρά από αέναες μελέτες, παρουσιάζοντας ένα βέλτιστο τρόπο κωδικοποίησης της κινούμενης εικόνας, δίνοντας έμφαση τόσο στον υποκειμενικό παράγοντα (ανθρώπινη αντίληψη), όσο και στον αντικειμενικό (σηματοδοσία). Με κύριο γνώμονα ότι η ακολουθία του σήματος (bitstream) μπορεί να αναλυθεί περαιτέρω, και να χωριστεί ανάλογα με το περιεχόμενό της, όπως για παράδειγμα το μέγεθος της εικόνας (spatial resolution), το ρυθμό ανανέωσης (temporal resolution) και την ποιότητα σηματοθορυβικού λόγου (SNR). Ο κάθε αλγόριθμος, ανάλογα την περίπτωση, δίνει προτεραιότητα σε ξεχωριστό στοιχείο. Είναι δυνατόν, ανάλογα με την ανάλυση της εικόνας και εξετάζοντας τα pixels, να εντοπιστούν περιοχές με μεγάλη ομοιογένεια και να αναπαρασταθούν με μικρότερη λεπτομέρεια, χωρίς η αλλαγή αυτή να γίνει αντιληπτή από το χρήστη. Για να μειωθεί ο όγκος της πληροφορίας, μπορούμε να εκμεταλλευτούμε το γεγονός ότι υπάρχει μια αλληλουχία μεταξύ των στιγμιότυπων και από τη στιγμή που παρουσιάζουν κινούμενη εικόνα δεν μπορεί παρά να έχουν μεγάλες ομοιότητες. Έτσι έχουν προταθεί αλγόριθμοι συμπίεσης που αναλύουν τις διαφορές ανάμεσα σε διαδοχικά καρέ. Γενικότερα, γίνονται προσπάθειες για να μειωθεί το απαιτούμενο εύρος ζώνης, αλλά και για την αποτελεσματικότερη διαχείριση των πιθανών λαθών, είτε κατά τη μεταφορά του περιεχομένου, είτε κατά τη διαδικασία της συμπίεσης. Προκειμένου να υπολογιστεί η αποτελεσματικότητα αυτών των προσπαθειών, μπορούν να αξιοποιηθούν διάφορες μέθοδοι μέτρησης της αντιλαμβανόμενης ποιότητας ψηφιακού βίντεο οι οποίες χωρίζονται σε δύο μεγάλες κατηγορίες, τις υποκειμενικές που στην ουσία πρόκειται για έρευνες που γίνονται σε ένα δείγμα ατόμων, και τις αντικειμενικές όπου οι έρευνες γίνονται πάνω στο μεταδιδόμενο σήμα.

### **1.3. Εκτίμηση ποιότητας ψηφιακών μέσων.**

Η μέτρηση της οπτικής ποιότητας των εικόνων και video, είναι θεμελιώδους σημασίας για πολλές εφαρμογές επεξεργασίας τους, εξαιτίας της ραγδαίας εξάπλωσης της ψηφιακής εικόνας στην καθημερινότητά μας. Πράγματι, αξίζει να σημειωθεί πως, στο στάδιο της μετάδοσης, οι ψηφιακές εικόνες υπόκεινται σε μία ευρεία ποικιλία στρεβλώσεων κατά τη διάρκεια της απόκτησης, επεξεργασίας, συμπίεσης, αποθήκευσης μετάδοσης και της αναπαραγωγής τους, με αποτέλεσμα να καταλήγουμε σε υποβάθμιση της οπτικής ποιότητας. Όλα τα παραπάνω, καθιστούν αναγκαία τη μελέτη μέτρησης ποιότητας της εικόνας, για πολυάριθμες εφαρμογές επεξεργασίας της. Οφείλουμε να παραδεχτούμε, πως οι άνθρωποι είναι οπτικά πλάσματα. Όντως, η κύρια λειτουργία του ανθρώπινου ματιού είναι να εξάγει δομικές πληροφορίες από το οπτικό πεδίο και το HVS(ανθρώπινο οπτικό σύστημα) είναι ιδιαίτερα προσαρμοσμένο για αυτόν το σκοπό. Κάθε άνθρωπος έχει διαφορετικά κριτήρια εκτίμησης της ποιότητας, διαφορετικές αντιλήψεις και απαιτήσεις που μπορεί να επηρεάζουν την κρίση του. Επομένως, καταλήγουμε πως για τις εφαρμογές, στις οποίες οι εικόνες τελικά να είναι ορατές από τον άνθρωπο, η μοναδική σωστή μέθοδος ποσοτικού προσδιορισμού οπτικής ποιότητας της εικόνας, γίνεται μέσω της υποκειμενικής αξιολόγησης. Στην παρακάτω εικόνα, παρατηρούμε ένα διάγραμμα ενός βασικού συστήματος μέτρησης ποιότητας. Σύμφωνα με τη διαθεσιμότητα μίας εικόνας αναφοράς, η αντικειμενική μέτρηση IQA μπορεί να χαρακτηριστεί ως πλήρης αναφοράς, μη αναφοράς και μειωμένης αναφοράς μεθόδους. Όπως παρατηρείται και στην εικόνα, ανάλογα με την περίπτωση χρησιμοποιείται μία από τις παραπάνω 3 μεθόδους, ώστε να εκτιμηθεί η ποιότητα της εικόνας.



**Εικόνα 1: Βασική Ροή Μέτρησης Ποιότητας.**

Για παράδειγμα, σε περίπτωση ενίσχυσης της εικόνας, μία μέθοδος, μη αναφοράς, μπορεί να χρησιμοποιηθεί αν και μόνο αν η εικόνα που δημιουργήθηκε στο άκρο εξόδου είναι απαραίτητη για υπολογισμό. Συνάμα, ένα επιπλέον παράδειγμα αφορά την κωδικοποίηση όπου η πλήρους αναφοράς, μέτρηση, με βάση το παραπάνω διάγραμμα IQA, θα μπορούσε να χρησιμοποιηθεί όπου απαιτείται ταυτόχρονα αποκωδικοποιημένη εικόνα από το άκρο εξόδου και πρωτότυπη εικόνα αναφοράς από την είσοδο(συνδέεται μέσω της διακεκομμένης γραμμής εδώ).

#### **1.4. Ποιότητα Εικόνας.**

Μια εικόνα από τη στιγμή που αποτυπώνεται έως τη στιγμή που εμφανίζεται στον ανθρώπινο παρατηρητή υπόκειται σε πολλά είδη παραμορφώσεων, λόγω των διαδικασιών στις οποίες υποβάλλεται. Για να αξιολογήσουμε μία εικόνα επομένως, χρησιμοποιούμε δύο μεθόδους:

- ❖ **Υποκειμενική Μέθοδο:** Η εμπειρία του χρήστη μπορεί να επηρεαστεί από πολλούς παράγοντες που μπορεί να οφείλονται στην ποιότητα του δικτύου, στην συμπίεση του περιεχομένου αλλά και στο υλικό μετάδοσης. Από τη μία, έχει παρατηρηθεί ότι σημαντικό ρόλο παίζει η απόσταση και η γωνία θέασης, το μέγεθος της εικόνας, η φωτεινότητα, η αντίθεση αλλά και το ίδιο το περιεχόμενο. Ακόμα εξωτερικοί περιβαλλοντικοί παράγοντες όπως υγρασία, ομίχλη, και η κίνηση επιβαρύνουν την τελική εμπειρία. Από την άλλη, όταν η κινούμενη εικόνα συνοδεύεται από υψηλής ποιότητας ήχο, δείχνει να επηρεάζει θετικά την κρίση του θεατή. Είναι η μοναδική, σωστή μέθοδος ποσοτικού προσδιορισμού οπτικής ποιότητας της εικόνας. Ωστόσο, όσον αφορά την πρώτη μέθοδο, αξίζει να σημειωθεί πως, είναι μία αρκετά δαπανηρή, άβολη και χρονοβόρα διαδικασία, δεδομένου ότι απαιτείται επιλογή ενός συγκεκριμένου πλήθους παρατηρητών, οι οποίοι θα δουν μία σειρά από εικόνες και στη συνέχεια θα αξιολογήσουν την ποιότητά τους, σύμφωνα με την προσωπική τους κρίση.
- ❖ **Αντικειμενική Μέθοδο:** Στον αντίποδα, η αντικειμενική μέθοδος, αποτελείται από μαθηματικά μοντέλα αξιολόγησης ποιότητας(αυτοματοποιημένοι αλγόριθμοι), που προσεγγίζουν τα αποτελέσματα της υποκειμενικής αξιολόγησης της εικόνας. Εδώ, κυρίαρχος στόχος είναι η ανάπτυξη ποσοτικής μέτρησης που να μπορεί να προβλέψει την αντιληπτική ποιότητα εικόνας.

Σε αυτό το σημείο, και έπειτα από την παραπάνω διάκριση των δύο μεθόδων που παρουσιάζονται ως μετρικές ποιότητας εικόνων, κρίνεται αναγκαίο να παρουσιασθεί και η διάκριση τόσο της υποκειμενικής, όσο και της αντικειμενικής μεθόδου, σύμφωνα με το ITU-R(International Telecommunication Union, Radiocommunication Sector), το οποίο, έχει μελετήσει και προτείνει μια σειρά από τεστ για την μέτρηση της αντιλαμβανόμενης ποιότητας σε κινούμενη εικόνα:

- ❖ **Double Stimulus Continuous Quality Scale (DSCQS):** Εδώ, παρουσιάζονται δύο φορές στον παρατηρητή το αρχικό αλλά και το εξεταζόμενο υλικό σε τυχαία σειρά, χωρίς αυτός

να γνωρίζει ποιο είναι το αρχικό και καλείται στο τέλος της προβολής να βαθμολογήσει κάθε ένα ξεχωριστά σε συνεχή κλίμακα από 1 έως 100.

- ❖ **Double Stimulus Impairment Scale (DSIS):** Σε αυτή την περίπτωση, ο παρατηρητής γνωρίζει ποιο είναι το αρχικό υλικό και βαθμολογεί σε κλίμακα διάταξης από 1 έως 5 τον βαθμό της αλλοίωσης που παρατήρησε στο εξεταζόμενο υλικό (όπου 1 είναι πολύ ενοχλητικό και 5 ανεπαίσθητο). Εδώ δεν έχουμε επανάληψη της προβολής.
- ❖ **Single Stimulus Continuous Quality Evaluation (SSCQE):** Με αυτή την μέθοδο οι παρατηρητές καλούνται να παρακολουθήσουν ένα μεγαλύτερης διάρκειας βίντεο, το οποίο έχει υποστεί κάποια επεξεργασία. Κατά τη διάρκεια της προβολής μπορούν να βαθμολογούν συνεχώς με μια μπάρα κύλισης, καθώς η ποιότητα μπορεί να αλλάξει κατά τη διάρκεια της προβολής.
- ❖ **Absolute Category Rating (ACR):** Εδώ οι παρατηρητές παρακολουθούν μόνο μια φορά το εξεταζόμενο βίντεο και δίνουν μια βαθμολογία από 1 έως 5 για την συνολική εμπειρία τους.
- ❖ **Pair Comparison (PC):** Σε αυτή τη μέθοδο παρουσιάζονται όλοι οι συνδυασμοί ανά δύο, μεταξύ του αρχικού και των προς εξέταση βίντεο, και ο παρατηρητής καλείται να διαλέξει ποιο ζευγάρι πιστεύει πως έχει την καλύτερη ποιότητα.

Όπως για την εκτίμηση της ποιότητας της κινούμενης εικόνας, η εκτίμηση της ποιότητας του ήχου μπορεί να γίνει με παρόμοιο τρόπο, δηλαδή με έρευνες σε ειδικά διαμορφωμένα εργαστήρια. Το υλικό προς εξέταση αλλά και οι συμμετέχοντες επιλέγονται προσεκτικά και βαθμολογούν το υλικό είτε απευθείας είτε συγκριτικά με κάποιο άλλο, και σε κάποιες περιπτώσεις είναι γνωστό το ποιο είναι το αρχικό δείγμα ενώ σε άλλες όχι. Η βαθμολογία μπορεί να γίνει στο τέλος για τη συνολική εμπειρία ή κατά τη διάρκεια της δοκιμής. Η έρευνα μπορεί να γίνει και μέσω Internet αντί σε εργαστήρια, όπου το υλικό προς αξιολόγηση μεταδίδεται με Live streaming ή γίνεται πρώτα download και μετά αναπαραγωγή. Οι μετρήσεις αυτές για κινούμενη εικόνα και ήχο ανάγονται σε ξεχωριστά Mean Opinion Score (MOS) με εύρος τιμών από 1 έως 5 και μπορούν να χρησιμοποιηθούν ως σημείο αναφοράς για μετέπειτα αντικειμενικές μετρήσεις. Στην περίπτωση των DSCQS (Double Stimulus Continuous Quality Scale) γίνεται χρήση της διαφοράς του MOS του αρχικού περιεχομένου από το MOS του εξεταζόμενου. (Difference Mean Opinion Score (DMOS)). Βάσει αυτών των αποτελεσμάτων μπορεί να γίνει περαιτέρω ανάλυση και να εξεταστεί η συσχέτισή τους βάση του συντελεστή Pearson, Spearman, του Μέσου Τετραγωνικού Σφάλματος και του OR (Outlier Ratio). Εν συνεχεία, όσον αφορά τις αντικειμενικές μεθόδους εκτίμησης της ποιότητας, βάση του διαχωρισμού του ITU-R, διακρίνουμε πέντε βασικές κατηγορίες, ανάλογα με το τι δεδομένα χρησιμοποιούνται για την μέτρηση.

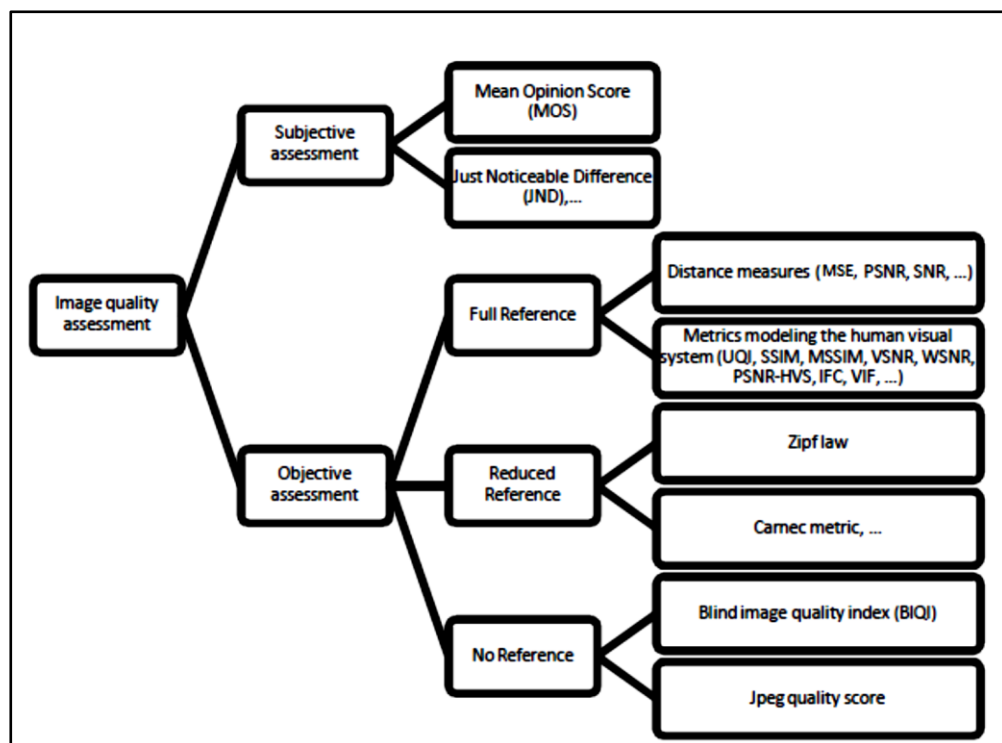
- ❖ **Media-layer Models:** Σε αυτά τα μοντέλα γίνεται μια εκτίμηση της ποιότητας της κινούμενης εικόνας ή του ήχου, έχοντας σαν παράμετρο το σήμα και το περιεχόμενο που μεταδίδεται. Ανάλογα με το αν έχουμε πρόσβαση στο αρχικό σήμα για σύγκριση, έχουμε διαφορετικές προσεγγίσεις. Έτσι έχουμε τα Full Reference (FR) μοντέλα, όπου εκεί έχουμε πλήρη πρόσβαση στο αρχικό σήμα και μπορεί να γίνουν άμεσες συγκρίσεις, τα Reduce Reference (RR) όπου έχουμε πρόσβαση στο τελικό σήμα και περιορισμένες πληροφορίες για το αρχικό, και τέλος τα μοντέλα No-Reference (NR), όπου δεν έχουμε καθόλου πρόσβαση στο αρχικό σήμα, παρά μόνο στο τελικό. Τα μοντέλα αυτά δεν λαμβάνουν υπόψη την κατάσταση του δικτύου και το hardware το οποίο χρησιμοποιείται.
- ❖ **Parametric Packet-Layer Models:** Σε αυτό το μοντέλο, η εκτίμηση της ποιότητας γίνεται realtime διαβάζοντας μόνο τις επικεφαλίδες των πακέτων, ώστε να μην επιβαρύνεται η διαδικασία της επεξεργασίας του σήματος.
- ❖ **Parametric Planning Models:** Σε αυτά τα μοντέλα η εκτίμηση της ποιότητας γίνεται βάσει των πληροφοριών που έχουμε από μόνο από το δίκτυο και το τερματικό στη φάση του σχεδιασμού. Είναι απαραίτητο δηλαδή να γνωρίζουμε από πριν το σύστημα στο οποίο θα γίνει η εξέταση και δεν γίνεται χρήση της μεταδιδόμενης πληροφορίας.

- ❖ **Bitstream Layer Models:** Εδώ λαμβάνονται υπόψη και οι επικεφαλίδες των πακέτων αλλά και το μεταδιδόμενο σήμα. Συνδυάζει, δηλαδή, στοιχεία των Media-layer και Parametric Packet-Layer.
- ❖ **Hybrid Models:** Μπορεί να δίνει συνδυασμός των παραπάνω μοντέλων για καλύτερα αποτελέσματα και εκμετάλλευση των πλεονεκτημάτων που έχει το καθένα.

Τέτοιου είδους μέθοδοι, θα μπορούσαν να εξαλείψουν την ανάγκη για δαπανηρές υποκειμενικές μελέτες. Η αντικειμενική αξιολόγηση θα μπορούσε να εφαρμοστεί ως εξής:

- ❖ Παρακολούθηση και έλεγχος της ποιότητας της εικόνας, σε συστήματα ποιοτικού ελέγχου.
- ❖ Συγκριτική αξιολόγηση των συστημάτων επεξεργασίας της εικόνας.
- ❖ Βελτιστοποίηση των αλγορίθμων και των παραμέτρων.
- ❖ Βοήθεια στους οικιακούς χρήστες στην καλύτερη διαχείριση των ψηφιακών τους φωτογραφιών και στην αξιολόγησή τους, από την εμπειρία τους στη φωτογράφιση.

Βασικό κριτήριο ταξινόμησης των αντικειμενικών μετρήσεων της ποιότητας της εικόνας, αποτελεί η διαθεσιμότητα ενός πρωτοτύπου εικόνας(μη παραμορφωμένη εικόνα), με την οποία η παραμορφωμένη εικόνα επρόκειτο να συγκριθεί. Οι περισσότερες υπάρχουσες προσεγγίσεις είναι γνωστές ως εξής:



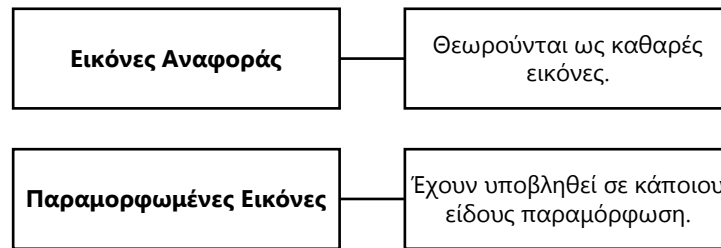
**Εικόνα 2: Μπλοκ Διάγραμμα Μέτρησης Ποιότητας Εικόνας.**

- ❖ **Πλήρους Αναφοράς(FR):** Στην περίπτωση αυτή, η εικόνα είναι πλήρως διαθέσιμη, ώστε να καθίσταται δυνατή η σύγκρισή της με την παραμορφωμένη εικόνα.
- ❖ **Μη Αναφοράς(NR):** Εδώ, η εικόνα αναφοράς δεν είναι διαθέσιμη, για αυτό και αποκαλείται και τυφλή αξιολόγηση της εικόνας.
- ❖ **Μειωμένης Αναφοράς(RR):** Στον τρίτο τύπο της αντικειμενικής μεθόδου, η αναφορά εικόνα είναι εν μέρει διαθέσιμη και υπό τη μορφή ενός συνόλου, εξαγονται χαρακτηριστικά, τα οποία διατίθενται ως επιπρόσθετη πληροφορία, προκειμένου να επιτευχθεί η αξιολόγηση της ποιότητας της παραμορφωμένης εικόνας.

## 1.5. Προσεγγίσεις μετρήσεων ποιότητας εικόνας.

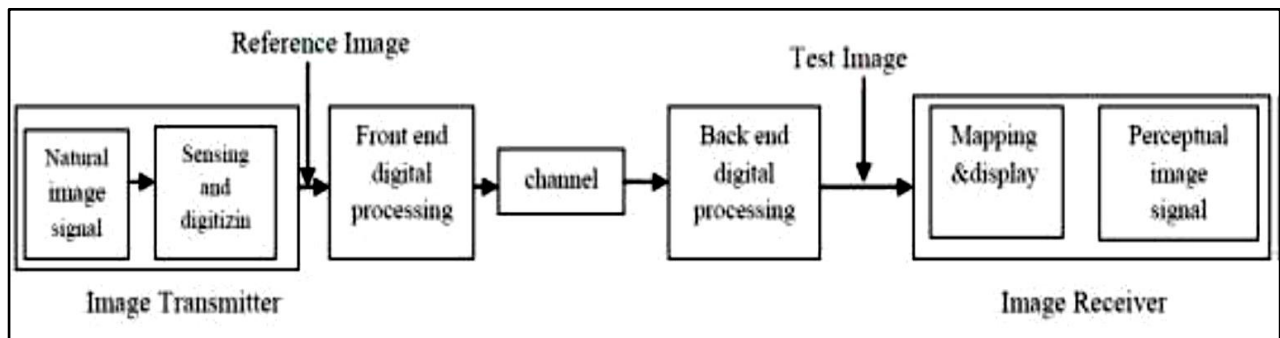
### 1.5.1 Αλγόριθμοι Πλήρους Αναφοράς.

Οι αλγόριθμοι πλήρους αναφοράς απαιτούν ως είσοδο όχι μόνο την παραμορφωμένη εικόνα, αλλά και μία «καθαρή» εικόνα αναφοράς, με την οποία αξιολογείται η παραμορφωμένη. Σε αυτήν την περίπτωση θα πραγματοποιηθεί χωρισμός των εικόνων σε δύο ομάδες:



**Εικόνα 3: Διάκριση εικόνας στην αντικειμενική μέθοδο.**

Ανάλογα με την πειραματική μέθοδο που ακολουθείται, οι παρατηρητές έχουν τη δυνατότητα να δουν είτε μόνο τις παραμορφωμένες εικόνες, είτε κάθε παραμορφωμένη εικόνα, ως ζεύγος με την αντίστοιχη εικόνα αναφοράς της. Κατόπιν, είναι σε θέση να αξιολογήσουν την αντιληπτή ποιότητα αναλόγως με την ποιότητα της παραμορφωμένης εικόνας, ή με τη διαφορά της από την εικόνα αναφοράς.



**Εικόνα 4: Επεξεργασία εικόνας.**

Παρόλα αυτά, αξίζει να τονισθεί πως, η αντιληπτή ποιότητα ποικίλει στον κάθε παρατηρητή, για αυτό και υπολογίζεται η μέση τιμή όλων των διαφορετικών τιμών της ποιότητας που λαμβάνονται από τους παρατηρητές και προκύπτει η τιμή MOS (Μέσος Όρος Γνωμοδότησης). Στις περιπτώσεις που αξιολογείται και η εικόνα αναφοράς, προκύπτει και η τιμή DMOS (Μέσος Όρος Διαφορικής Γνωμοδότησης) αντί της MOS, και η οποία απορρέει από την αξιολόγηση των εικόνων αναφοράς και εκπαίδευσης. Εν κατακλείδι, και στις δύο περιπτώσεις γνωμοδότησης, κρίνεται επιθυμητό, όσο το δυνατόν η καλύτερη ταύτιση της τιμής MOS/DMOS και των αποτελεσμάτων του αλγορίθμου.

### **1.5.2. Αλγόριθμοι Μη Αναφοράς.**

Το NR μοντέλο αξιολόγησης ποιότητας οπτικών μέσων, αφορά στην αυτόματη αξιολόγηση της ποιότητας μιας εικόνας χρησιμοποιώντας έναν αλγόριθμο. Η μοναδική πληροφορία που λαμβάνει ο συγκεκριμένος αλγόριθμος, πριν προβλέψει την ποιότητα, είναι η παραμορφωμένη εικόνα, την οποία και θέλουμε να αξιολογήσουμε. Με κριτήριο, την προηγούμενη γνώση των τύπων παραμόρφωσης, οι αλγόριθμοι μη αναφοράς, ταξινομούνται σε δύο κατηγορίες:

- ❖ Αλγόριθμους Ειδικών Παραμορφώσεων.
- ❖ Αλγόριθμους Μη Ειδικών Παραμορφώσεων.

Οι περισσότερες υπάρχουσες μέθοδοι μη αναφοράς, είναι ειδικών παραμορφώσεων, υποθέτοντας, ότι ο τύπος παραμόρφωσης, είναι γνωστός, όπως ο θόρυβος, ή η θόλωση. Με την

παραδοχή αυτή, περιορίζουμε το πεδίο εφαρμογής αυτών των αλγορίθμων. Οι αλγόριθμοι μη ειδικών παραμορφώσεων, δε λαμβάνουν υπόψη την προηγούμενη γνώση του τύπου παραμόρφωσης, αλλά αντίθετα δίνουν αποτελέσματα ποιότητας, με την υπόθεση ότι η εικόνα που αξιολογείται έχει τον ίδιο τύπο παραμόρφωσης με τις εικόνες της βάσης δεδομένων εκπαίδευσης. Αυτές οι μέθοδοι συνήθως περιλαμβάνουν τεχνικές μηχανικής εκμάθησης, όπου τα διακριτά χαρακτηριστικά της ποιότητας της εικόνας καταχωρούνται στα μοντέλα εκμάθησης και στη συνέχεια, χρησιμοποιούνται για αξιολόγηση της ποιότητας των εικόνων εκπαίδευσης. Εν κατακλείδι, η προσέγγιση αυτών των αλγορίθμων, βασίζεται στην αρχή ότι, οι φυσικές εικόνες έχουν ορισμένες συνήθειες στατιστικές ιδιότητες οι οποίες, λόγω ύπαρξης στρεβλώσεων τροποποιούνται.

### **1.5.3. Αλγόριθμοι Μειωμένης Αναφοράς.**

Εδώ, παρατηρούμε πως, διατίθενται κάποιες πληροφορίες σχετικά με την εικόνα αναφοράς αλλά όχι την πραγματική εικόνα, παρά μόνο την παραμορφωμένη. Οι μειωμένες αναφοράς αλγόριθμοι μπορούν να προβλέψουν την υποβάθμιση της εικόνας με μερικές μόνο πληροφορίες από την εικόνα αναφοράς, υπό μορφή ενός συνόλου χαρακτηριστικών. Αποτελούν ένα πρακτικό εργαλείο για εφαρμογές πραγματικού χρόνου, όπως οι μεταδόσεις οπτικών σημάτων μέσω ενσύρματου ή ασύρματου δικτύου, όπου μπορούν να χρησιμοποιηθούν για την παρακολούθηση των αλλαγών της ποιότητας της εικόνας, και να ελέγχουν τη ροή των πόρων του δικτύου. Σε γενικό επίπεδο, το σύστημα περιλαμβάνει μία εκχύλιση διαδικασίας χαρακτηριστικών στην πλευρά του αποστολέα και μία διαδικασία ανάλυσης της ποιότητας από την πλευρά του δέκτη. Τα εξαγόμενα χαρακτηριστικά ή οι πληροφορίες της κάθε πλευράς, συνήθως έχουν έναν πολύ χαμηλότερο ρυθμό δεδομένων από αυτόν των δεδομένων της εικόνας και συνήθως μεταδίδονται στο δέκτη, μέσα από ένα βοηθητικό κανάλι, το οποίο θεωρείται ότι δεν έχει λάθη, Συνάμα, αξίζει να σημειωθεί πως το βοηθητικό κανάλι και το κανάλι παραμόρφωσης μπορούν να συγχωνευθούν, όπου στο δεύτερο, τα χαρακτηριστικά μειωμένης αναφοράς της αρχικής εικόνας, θα πρέπει να λαμβάνουν μεγαλύτερη προστασία(έλεγχος σφαλμάτων κωδικοποίησης) από ότι τα δεδομένα εικόνας κατά τη διάρκεια της μετάδοσης. Εν τέλει, από τη σκοπιά του δέκτη, σημειώνουμε πως, η διαφορά των χαρακτηριστικών που εξάγονται από τις εικόνες παραμόρφωσης και αναφοράς, χρησιμοποιούνται για την αξιολόγηση της υποβάθμισης της ποιότητας της εικόνας. Η διεργασία εκχύλισης επομένως, προσαρμόζεται σύμφωνα με τις πληροφορίες που προέρχονται από τα RR χαρακτηριστικά μέσω του βοηθητικού καναλιού.

**Κεφάλαιο 2°**  
**Μέθοδοι Εκτίμησης**  
**Ποιότητας Εικόνων.**



**2.1. Μέσος Τετραγωνικού Σφάλματος(MSE).**

Ο μέσος τετραγωνικού σφάλματος, είναι ένας προφανής τρόπος μέτρησης της ομοιότητας και χρησιμοποιείται στον υπολογισμό ενός σήματος σφάλματος, αφαιρώντας το δοκιμαστικό σήμα από την αναφορά, και στη συνέχεια υπολογίζοντας το μέσο όρο ενέργειας του σήματος σφάλματος. Αυτός ο τρόπος μέτρησης, αποτελεί τον πιο απλό και είναι ευρέως χρησιμοποιούμενος, με πλήρη αναφορά μέτρησης ποιότητας της εικόνας. Επιπρόσθετα, χρησιμοποιείται συχνά στην επεξεργασία σήματος και ορίζεται ως εξής:

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2$$

Όπου το  $x(i,j)$  αναπαριστά την αρχική(αναφοράς) εικόνα και το  $y(i,j)$  αναπαριστά την παραμορφωμένη(τροποποιημένη) εικόνα. Οι μεταβλητές  $i,j$  εκφράζουν το εικονοστοιχείο στη θέση της  $M \times N$  εικόνας. Αξίζει να τονιστεί πως ο MSE παίρνει την τιμή μηδέν, αν και μόνο αν οι δύο εικόνες είναι πανομοιότυπες, και ισχύει ότι:

$$X(i,j) = Y(i,j)$$

**2.2. Λόγος Σήματος προς Θόρυβο(PSNR).**

Ο λόγος σήματος προς θόρυβο, χρησιμοποιείται ως μέτρο εκτίμησης της ποιότητας σε περιπτώσεις συμπίεσης της εικόνας. Αξιολογείται σε dB( ντεσιμπέλ) και είναι αντιστρόφως ανάλογος του μέσου τετραγωνικού σφάλματος MSE. Ορίζεται ως εξής:

$$PSNR = 10 \log_{10} (MAX_i^2 / MSE)$$

Το  $MAX_i$  ορίζεται ως η μέγιστη τιμή που μπορεί να λάβει το εικονοστοιχείο μίας εικόνας. Όταν τα εικονοστοιχεία αναπαρίστανται χρησιμοποιώντας 8 bits/δείγμα, τότε η τιμή του ορίζεται στο 255. Ακόμη, τυπικές τιμές που μπορεί να πάρει το PSNR κυμαίνονται μεταξύ του 30 και του 50 dB, για εικόνες οι οποίες παρουσιάζουν πολύ μικρές διαφορές λόγω κβάντισης.

**2.3. Καθολικός Δείκτης Ποιότητας Εικόνας(UIQI).**

Έστω  $x = \{x_i | i = 1, 2, 3, \dots, N\}$ ,  $y = \{y_i | i = 1, 2, 3, \dots, N\}$  για την αρχική και τη δοκιμαστική εικόνα αντίστοιχα:

$$Q = \frac{4 * \sigma_{xy} \bar{x} \bar{y}}{(\sigma_x^2 + \sigma_y^2) * ((\bar{x})^2 + (\bar{y})^2)}$$

Παρακάτω, εκφράζουμε αναλυτικά τους όρους της παραπάνω σχέσης. Επομένως, τα  $\bar{x}$ ,  $\bar{y}$ ,  $\sigma_x^2$ ,  $\sigma_y^2$  και  $\sigma_{xy}$  δίνονται ως εξής:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$$

$$\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

$$\sigma_y^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})^2$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

**Σχόλια – Παρατηρήσεις.**

- ❖ Το δυναμικό εύρος του Q είναι [0,1].
- ❖ Η καλύτερη τιμή του Q είναι όταν γίνεται ίση με 1 (Q=1) και επιτυγχάνεται όταν το  $y_i = x_i$  όπου  $i = 1, 2, \dots, n$ .

- ❖ Το παραπάνω μοντέλο δείκτη ποιότητας για οποιαδήποτε στρέβλωση, συνδυάζεται με τρεις διαφορετικούς παράγοντες:
  - ❖ Απώλεια Συσχετισμού
  - ❖ Στρέβλωση Φωτεινότητας
  - ❖ Παραμόρφωση Αντίθεσης

Προκειμένου να εμπεδώσουμε σε μεγαλύτερο βαθμό, την 3<sup>η</sup> παρατήρηση, ξαναγράφουμε τον ορισμό του Q ως προϊόν τριών συνιστωσών :

$$Q = Q_1 * Q_2 * Q_3 \quad (1)$$

$$Q_1 = \frac{\sigma_{xy}}{\sigma_x * \sigma_y} \quad (2)$$

$$Q_2 = \frac{2 * \bar{x} * \bar{y}}{((\bar{x})^2 + (\bar{y})^2)} \quad (3)$$

$$Q_3 = \frac{2 * \sigma_x * \sigma_y}{(\sigma_x^2 + \sigma_y^2)} \quad (4)$$

Επομένως, σύμφωνα με τα επιμέρους  $Q_1$ ,  $Q_2$  και  $Q_3$  προκύπτει και το Q ως συνολικό γινόμενο, το οποίο εκφράζεται παρακάτω:

$$Q = \frac{\sigma_{xy}}{\sigma_x + \sigma_y} \times \frac{2 * \bar{x} * \bar{y}}{((\bar{x})^2 + (\bar{y})^2)} \times \frac{2 * \sigma_x * \sigma_y}{(\sigma_x^2 + \sigma_y^2)}$$

#### Σχόλια – Παρατηρήσεις.

- ❖ **Συνιστώσα  $Q_1$ :** Είναι ο συντελεστής συσχέτισης μεταξύ του x και του y, που μετρά τον βαθμό γραμμικής συσχέτισης μεταξύ του x και του y. Η καλύτερη τιμή είναι το 1 και επιτυγχάνεται όταν  $y_i = ax_i + b$  για  $i = 1, 2, \dots, N$  όπου τα a και b είναι σταθερές. Ακόμα και αν τα x και y συνδέονται μεταξύ τους γραμμικά, μπορεί ακόμα η στρέβλωση μεταξύ τους να είναι σχετική, αυτά αξιολογούνται στην δεύτερη και στο τρίτη συνιστώσα.
- ❖ **Συνιστώσα  $Q_2$ :** Έχει εύρος τιμών [0, 1] και εκφράζει το πόσο κοντά στη φωτεινότητα βρίσκονται οι παράμετροι x και y. Αν  $\bar{x} = \bar{y}$ , τότε η τιμή της συνιστώσας θα είναι ίση με 1. Τέλος, οι παράμετροι  $\sigma_x$  και  $\sigma_y$  χρησιμοποιούνται για την εκτίμηση της αντίθεσης των x και y.
- ❖ **Συνιστώσα  $Q_3$ :** Μετρά τις ομοιότητες μεταξύ των αντιθέσεων των εικόνων. Το εύρος των τιμών είναι επίσης [0,1], όπου καλύτερη τιμή είναι το 1 και επιτυγχάνεται αν και μόνο αν ισχύει  $\sigma_x = \sigma_y$ .

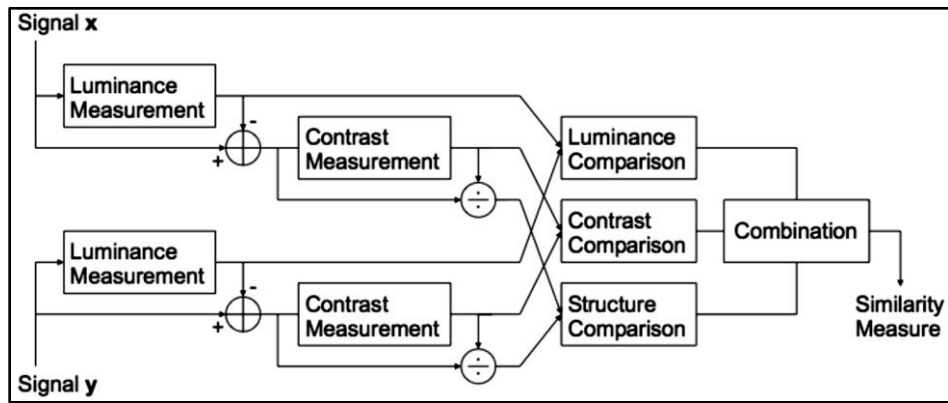
Εφαρμόζουμε την μέθοδο μέτρησης ποιότητας στην τοπική περιοχή, χρησιμοποιώντας μια προσέγγιση ολίσθησης παραθύρου. Ξεκινώντας από την πάνω αριστερή γωνία της εικόνας, με μία ολίσθηση παραθύρου μεγέθους B x B κινούμενη από pixel σε pixel οριζόντια και κατακόρυφα σε όλες τις γραμμές και στήλες της εικόνας μέχρι την κάτω δεξιά γωνία. Ο δείκτης υπολογίζεται για κάθε παράθυρο, και αυτό οδηγεί σε ένα χάρτη ποιότητας της εικόνας. Ο γενικός δείκτης ποιότητας εκφράζεται ως ο μέσος όρος όλων των τιμών Q στο χάρτη ποιότητας:

$$Q = \frac{1}{M} \sum_{j=1}^M Q_j$$

Όπου M ο συνολικός αριθμός παραθύρων.

#### 2.4. Διαρθρωτικός Δείκτης Μέτρησης Ομοιότητας (SSIM).

Η μέθοδος αυτή, βασίζεται στη σύγκριση των δομών της εικόνας αναφοράς και της παραμορφωμένης εικόνας.



**Εικόνα 5: Περιγραφή της λειτουργίας του SSIM.**

Αν  $x$  και  $y$  είναι δύο μη αρνητικά σήματα που αντιστοιχούν στην αρχική και στην παραμορφωμένη εικόνα, αν  $\bar{x}$ ,  $\bar{y}$ ,  $\sigma^2x$ ,  $\sigma^2y$  και  $\sigma_{xy}$  είναι ο μέσος όρος του  $x$ , ο μέσος όρος του  $y$ , η διακύμανση του  $x$  η διακύμανση του  $y$ , και η διακύμανση των  $x$  και  $y$ , αντίστοιχα. Εδώ ο μέσος όρος και η τυπική απόκλιση (τετραγωνική ρίζα της διακύμανσης) ενός σήματος είναι κατά προσέγγιση και θεωρούνται ως εκτιμήσεις της φωτεινότητας και της αντίθεσης του σήματος. Ο SSIM περιγράφεται με μαθηματικό τρόπο στην παρακάτω εξίσωση:

$$SSIM = \frac{(2 \times \bar{x} \times \bar{y} + C1) \times (\sigma^2_x + \sigma^2_y + C2)}{(\sigma^2_x + \sigma^2_y + C2) \times ((\bar{x})^2) + ((\bar{y})^2) + C1}$$

#### Σχόλια – Παρατηρήσεις.

- ❖ Σε αυτόν τον τύπο, εμπλέκονται τρία διαφορετικά μέτρα:
  - ❖ Η φωτεινότητα.
  - ❖ Η αντίθεση.
  - ❖ Η δομή μέτρων σύγκρισης.
- ❖ Τα  $C1$  και  $C2$  είναι σταθερές, και προστίθενται για την πρόληψη της ασταθούς μέτρησης όταν  $((\bar{x})^2 + (\bar{y})^2)$  ή  $(\sigma^2_x + \sigma^2_y)$  είναι κοντά στο μηδέν.
- ❖ Τα  $\bar{x}$ ,  $\bar{y}$ ,  $\sigma^2x$ ,  $\sigma^2y$  και  $\sigma_{xy}$  δίνονται ως:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$$

$$\sigma^2_x = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

$$\sigma^2_y = \frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})^2$$

$$\sigma^2_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

Παρόμοια με την περίπτωση του UIQI, γίνεται και στον SSIM, εφαρμόζοντας την μέθοδο μέτρησης ποιότητας στην τοπική περιοχή, χρησιμοποιώντας ένα συρόμενο παράθυρο προσέγγισης. Ξεκινώντας από την πάνω αριστερή γωνία της εικόνας, ένα συρόμενο παράθυρο μεγέθους  $B \times B$  κινείται από pixel σε pixel οριζόντια και κατακόρυφα μέσα από όλες τις σειρές και στήλες της εικόνας έως ότου φτάσει στην κάτω δεξιά γωνία. Η συνολική ποιότητα εικόνας MSSIM λαμβάνεται με τον υπολογισμό του μέσου όρου τιμών των SSIM επί όλων των παραθύρων:

$$MSSIM = \frac{1}{M} \sum_{j=1}^M SSIM_j$$

Όπου M ο συνολικός αριθμός παραθύρων.

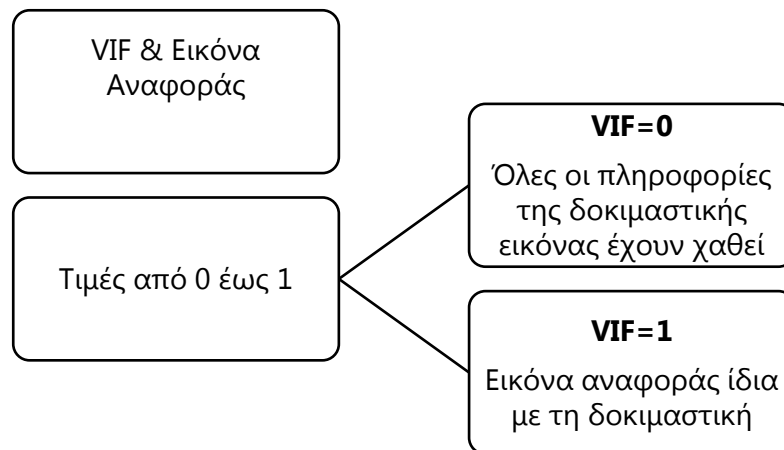
### 2.5. Πληροφορία Οπτικής Πιστότητας (VIF).

Μεταξύ των πιο πρόσφατων μετρήσεων αντικειμένων που ενσωματώνουν το ανθρώπινο οπτικό σύστημα είναι η πληροφορία οπτικής πιστότητας (VIF). Ο VIF είναι το αποτέλεσμα μιας βελτιωμένης πιστότητας στο κριτήριο της πληροφορίας, ο οποίος είναι για να ενσωματώσει ένα βήμα εξομάλυνσης. Υπολογίζει την ποιότητα της εικόνας με βάση τη φυσική-σκηνή στατιστικές. Λειτουργεί υπό την προϋπόθεση ότι ο HVS έχει εξελιχθεί με βάση τις στατιστικές ιδιότητες του φυσικού περιβάλλοντος. Κατά συνέπεια, η ποιότητα της παραμορφωμένης εικόνας μπορεί να ποσοτικοποιηθεί με βάση την ποσότητα των πληροφοριών. Αυτή η μέτρηση λαμβάνεται από τη σχέση:

$$VIF = \frac{\sum_{\text{jsubbands}} I(\overline{C^{N,j}}; \overline{F^{N,j}} | \overline{S^{N,j}})}{\sum_{\text{jsubbands}} I(\overline{C^{N,j}}; \overline{E^{N,j}} | \overline{S^{N,j}})}$$

Όπου  $\overline{C^{N,j}}$  αντιπροσωπεύει τα N στοιχεία της πλήρους αναφοράς  $C_j$  για j αριθμούς υπό περιοχών.

Στο γραφικό SmartArt που ακολουθεί παρακάτω, υπογραμμίζουμε τις τιμές που προσεγγίζει η πληροφορία οπτικής πιστότητας, σε σχέση με την εικόνα αναφοράς.



**Εικόνα 6: Σύνδεση VIF και εικόνας αναφοράς.**

### 2.6. Κλίση Απόκλιση Ομοιότητας Μεγέθους (GMSD).

$$GMSD = \sqrt{1/N \sum_{i=1}^N (GMS(i) - GMSM)^2}$$

Οι εικόνες διαβάθμισης μεγέθους των R και D υπολογίζονται ως ακολούθως:

$$m_r(i) = \sqrt{(r \otimes h_x)^2 + (r \otimes h_y)^2} \quad (i)$$

$$m_d(i) = \sqrt{(d \otimes h_x)^2 + (d \otimes h_y)^2} \quad (ii)$$

Όπου  $h_x$  οι τιμές τις εικόνας αναφοράς και  $h_y$  τις διαστρεβλωμένης αποδίδουν τις οριζόντιες και κάθετες εικόνες r και d. Όπου το σύμβολο " $\otimes$ " σημαίνει πράξη συνέλιξης. Έχοντας τις εικόνες κλίσης μεγέθους  $m_r$  και  $m_d$ , η ομοιότητα μεγέθους κλίσης (GMS) υπολογίζεται ως εξής:

$$GMS(i) = \frac{2m_r(i)m_d(i)+c}{m^2_r(i)+m^2_d(i)+c}$$

Στη συνέχεια εφαρμόζουμε την μέση συγκέντρωση του GMS και υπολογίζουμε τον μέσο ομοιότητας κλίσης Μεγέθους (GMSM) ως εξής:

$$GMSM = \frac{1}{N} \sum_{i=1}^N GMS(i)$$

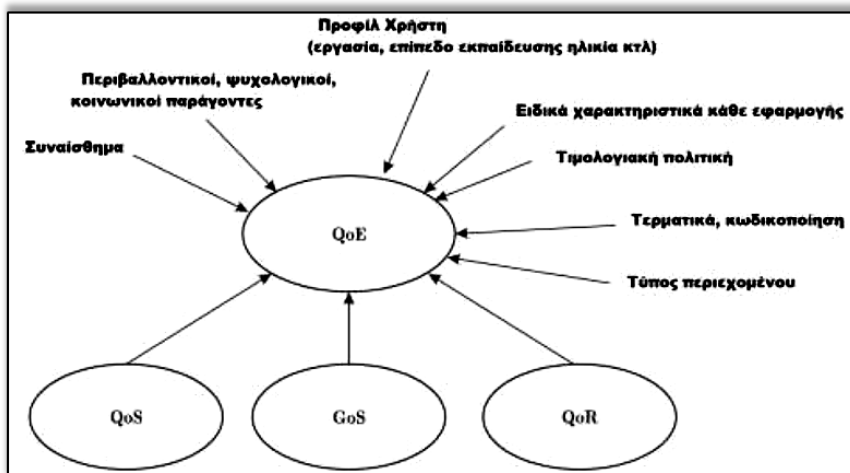
Όπου N ο συνολικός αριθμός pixel της εικόνα. Όσο υψηλότερη τιμή έχει η παράμετρος GMSM τόσο υψηλότερη συνολική ποιότητα εικόνας έχουμε. Έτσι έχοντας υπολογίσει τους GMS(i) και GMSM μπορούμε να υπολογίσουμε την Κλίση Απόκλιση ομοιότητας Μεγέθους (GMSD) ως εξής:

$$GMSD = \sqrt{\frac{1}{N} \sum_{i=1}^N (GMS(i) - GMSM)^2}$$

Η τιμή του GMSD αντικατοπτρίζει το εύρος της στρέβλωσης βαρύτητας σε μια εικόνα. Όσο υψηλότερη είναι η τιμή του GMSD, τόσο μεγαλύτερο είναι το εύρος παραμόρφωσης, και, συνεπώς, τόσο χαμηλότερη είναι η αντιληπτική ποιότητας της εικόνας.

**2.7. Ποιότητα εμπειρίας χρήστη QoE (Quality of Experience).**

Γενικότερα, ο όρος QoE (Quality of Experience) χρησιμοποιείται για να εκφράσει το συνολικό βαθμό ικανοποίησης του τελικού χρήστη από μια υπηρεσία. Όπως έχει αναφερθεί σε προηγούμενη ενότητα, ο βαθμός αυτός, επηρεάζεται όχι μόνο από την κατάσταση του δικτύου, αλλά και από άλλους παράγοντες, όπως τις συσκευές που χρησιμοποιούνται ή το περιεχόμενο που προσφέρεται. Για παράδειγμα σε ιδανικές συνθήκες της κατάστασης του δικτύου, ο χρήστης μπορεί να βαθμολογήσει εντελώς διαφορετικά την εμπειρία του στην προβολή του ίδιου περιεχομένου, ανάλογα με τη συσκευή που έγινε η προβολή, το μέγεθος της οθόνης, ή ακόμη και την παρουσίαση των χρωμάτων. Σε αντίθεση, η τελική αίσθηση που έχει ο χρήστης μπορεί να συνδέεται άμεσα με το θέμα του περιεχομένου που προβλήθηκε. Πιο συγκεκριμένα, αν του είναι ευχάριστο και γνώριμο είναι πολύ πιθανό να αποκομίσει θετικότερες εντυπώσεις. Συνάμα, μπορεί να επηρεαστεί και από πρόσθετους παράγοντες, λόγω χάρη από την ευκολία χρήσης, τις απαιτήσεις που μπορεί να έχει ο ίδιος ο χρήστης, ή την αποτελεσματικότητα. Τέλος, για την εκτίμηση του QoE πρέπει να λαμβάνονται υπόψη και κριτήρια όπως φύλλο, ηλικία, ή ακόμη και η προηγούμενη εμπειρία χρήσης. Για την καλύτερη κατανόηση της τελικής εμπειρίας του χρήστη, χρειάζεται να δοθεί ιδιαίτερη προσοχή στις προϋποθέσεις που θέτει ο καθένας, δηλαδή να μπορεί να έχει πρόσβαση σε οποιαδήποτε υπηρεσία, οποιαδήποτε ώρα, σε οποιοδήποτε μέρος και με οποιαδήποτε συσκευή. Από τη μεριά του δικτύου έχουμε περιορισμούς ανάλογα το δίκτυο, τον εξοπλισμό, την κωδικοποίηση, τα τερματικά, τα πρωτόκολλα τα οποία περιλαμβάνονται σε αυτό. Οι παρεχόμενες υπηρεσίες δεν θα πρέπει να έχουν ως απαιτούμενο την τεχνολογία αιχμής, ώστε να είναι δυνατή η χρήση τους από το σύνολο των χρηστών και με χαμηλό κόστος. Όπως είναι αναμενόμενο, ένα πλήθος υπηρεσιών, από τις πλέον απλές σε απαιτήσεις μέχρι και τις πιο σύνθετες, θα πρέπει να είναι στη διάθεση του χρήστη ανεξάρτητα από τη θέση του ή τη σύνδεσή του.



**Εικόνα 7: Παράγοντες που επηρεάζουν το QoE.**

## **2.8. Ποιότητα υπηρεσίας - QoS (Quality of Service).**

Ο όρος QoS (Quality of Service) χρησιμοποιείται καθαρά για την αποτύπωση της ποιότητας της υπηρεσίας σε επίπεδο δικτύου και βάσει αυτού γίνεται καλύτερη διαχείριση και αξιοποίηση των πόρων του δικτύου. Ένα ρεύμα πακέτων από μια προέλευση προς έναν προορισμό, ονομάζεται ροή (flow). Τα πακέτα που ανήκουν σε μια ροή μπορεί να ακολουθούν το ίδιο ή διαφορετικό δρομολόγιο. Οι ανάγκες της κάθε ροής μπορούν να χαρακτηριστούν από τέσσερις βασικές παραμέτρους οι οποίες σχετίζονται άμεσα με το QoS και θα πρέπει να εξεταστούν:

- ❖ **Bandwidth:** Ένα μέτρο του ρυθμού με τον οποίο μπορούν να μεταδοθούν δεδομένα μέσω ενός δικτύου σε δεδομένο χρόνο. Μετριέται συνήθως σε bits/sec.
- ❖ **Latency:** Είναι η καθυστέρηση από τη στιγμή της αποστολής ενός πακέτου δεδομένων μέχρι να φτάσει στον προορισμό του και μετριέται σε ms. Η καθυστέρηση αυτή επηρεάζεται από την κατάσταση του δικτύου ή τη διαδρομή που θα ακολουθήσουν τα πακέτα.
- ❖ **Jitter:** Από τη στιγμή που το κάθε πακέτο μπορεί να δρομολογείται ξεχωριστά, η καθυστέρηση μπορεί να έχει μεγάλη διακύμανση ανάμεσα στα πακέτα. Σε επικοινωνίες πραγματικού χρόνου, ο αποδέκτης περιμένει από τον αποστολέα να τηρεί ένα συγκεκριμένο ρυθμό κατά την αποστολή. Όμως, η χρονική καθυστέρηση μεταξύ των πακέτων μπορεί λόγω των συνθηκών του δικτύου να διαφέρει. Αυτή η διαφορά του χρόνου ανάμεσα στην αναμενόμενη λήψη και την πραγματική, η διακύμανση δηλαδή της καθυστέρησης ονομάζεται jitter.
- ❖ **Packet-loss:** Κατά τη μεταφορά των πακέτων μπορεί να υπάρξουν και απώλειες (packet loss) που μπορεί να οφείλονται είτε στο jitter, αν έχουμε υπηρεσία πραγματικού χρόνου (αφού σε αυτή την περίπτωση αν κάποιο πακέτο αργήσει πάνω από ένα επιτρεπτό όριο θεωρείται χαμένο και απορρίπτεται), είτε σε υπερφόρτωση του δικτύου κτλ. Φυσικά, μεγάλο ρόλο παίζει η κωδικοποίηση που γίνεται αλλά και το hardware που χρησιμοποιείται.

Οι παραπάνω παράγοντες μπορεί να έχουν αντίκτυπο στην τελική ποιότητα του περιεχομένου και μπορούν να επιφέρουν μείωση της τιμής του QoE. Σε περιπτώσεις που έχουμε μετάδοση κινούμενης εικόνας, μπορεί να παρατηρήσουμε περιοχές της εικόνας που να έχουν μικρά block με το ίδιο χρώμα και αυτό μπορεί να συμβεί λόγω κάποιου λάθους στην κωδικοποίηση. Επίσης μπορεί να παρατηρηθεί κάποιο θόλωμα ή πάγωμα στην εικόνα, για το οποίο πιθανότατα να προέρχεται από το packet loss και να ευθύνεται η σύνδεση. Ακόμα, υπάρχει περίπτωση να μην έχουμε ομαλή ροή του βίντεο αν έχουμε υψηλό jitter. Σε περίπτωση που έχουμε μετάδοση φωνής (VOIP), μπορεί να έχουμε μικρά κενά κατά την αναπαραγωγή του ήχου. Ο χρήστης δεν θα αντιληφθεί αν υπάρχει υψηλό packet loss ή jitter, αλλά θα εκφράσει τη δυσαρέσκειά του για τα παράσιτα που μπορεί να παρουσιαστούν ή την καθυστέρηση ήχου. Σε περίπτωση συνομιλίας, είναι σημαντικό να μην υπάρχει καθυστέρηση και επιστροφή ήχου αφού κάτι τέτοιο προκαλεί σύγχυση και δεν μπορεί να γίνει αποτελεσματική χρήση της υπηρεσίας. Σε ταυτόχρονη μεταφορά εικόνας – ήχου (πχ video-conference, iptv) για να υπάρχει ικανοποίηση από τον χρήστη πρέπει, πέρα από την ομαλή ροή, να υπάρχει και σωστός συγχρονισμός, καθώς τα πακέτα ήχου και εικόνας μεταφέρονται ξεχωριστά. Γενικότερα, κάθε εφαρμογή έχει διαφορετικές απαιτήσεις σε εύρος ζώνης (bandwidth), και διαφορετική ανοχή σε καθυστέρηση, σε packet loss, στο jitter. Βάσει αυτών των απαιτήσεων μπορούν να κατηγοριοποιηθούν ως εξής:

- ❖ **Ελαστικές ή μη διαδραστικές:** Πρόκειται για εφαρμογές email και κατέβασμα αρχείων.
- ❖ **Ελαστικές – διαδραστικές:** Εφαρμογές περιήγησης στο διαδίκτυο.
- ❖ **Ανελαστικές – μη διαδραστικές:** Εδώ συναντάμε μονόδρομη επικοινωνία, πχ μετάδοση βίντεο, ήχου, IPTV.
- ❖ **Ανελαστικές – διαδραστικές:** Εδώ έχουμε αμφίδρομη επικοινωνία, για παράδειγμα Video Conference, Voip, online παιχνίδια όπου τυχόν καθυστέρηση θα καταστήσει ανέφικτη τη χρήση της υπηρεσίας.

Ωστόσο, η κατηγοριοποίηση μπορεί να γίνει και με άλλους τρόπους. Το ITU-T κάνει διαχωρισμό μεταξύ εφαρμογών μετάδοσης εικόνας, ήχου και δεδομένων. Για τις κατηγορίες εικόνας και ήχου γίνεται επιπλέον διαχωρισμός σε αυτές που πραγματοποιούνται σε ζωντανό χρόνο (real-time), αμφίδρομες ή μη, και σε αυτές που δεν γίνονται real-time. Οι εφαρμογές της ίδιας κατηγορίας έχουν κοινά χαρακτηριστικά, παρόμοια ευαισθησία στην διακύμανση της ποιότητας του δικτύου.

Οι απαιτήσεις αυτές των εφαρμογών, μπορούν να το QoS θα μπορούσε να θεωρηθεί υποσύνολο του QoE αφού το επηρεάζει άμεσα. Όπως έχουμε αναφέρει, πέρα από τις δυνατότητες και την ποιότητα του δικτύου, η αντίληψη της ποιότητας από τον χρήστη, θα γίνει μέσω της συσκευής από την οποία θα γίνει η χρήση της υπηρεσίας. Η συσκευή αυτή μπορεί να είναι οποιαδήποτε, από την τηλεόραση, τον σταθερό υπολογιστή μέχρι και κάποιο tablet ή κινητό τηλέφωνο. Όλες αυτές οι συσκευές διαφέρουν πολύ σε μέγεθος οθόνης, ανάλυσης, βάθος χρώματος, φωτεινότητα, αντίθεση, και χρόνο απόκρισης. Επιπλέον, είναι διαφορετικές και οι δυνατότητες κάθε συσκευής σε επεξεργαστική ισχύ, μέγεθος μνήμης κατανάλωση ενέργειας κτλ. Όπως είναι λογικό, παίζουν καθοριστικό ρόλο στο τελικό QoE του χρήστη καθώς διαφορετικά θα αξιολογήσει το περιεχόμενο σε μία τηλεόραση 50 ιντσών τελευταίας τεχνολογίας με υψηλή ανάλυση σε μεταφραστούν σε παραμέτρους του δικτύου και του QoS γενικότερα, όπου μπορούμε να έχουμε διάφορες κλάσεις CoS (Class of Service) ανάλογα τον τύπο του QoS. Όπως γίνεται αντιληπτό υπάρχει άμεση σχέση μεταξύ του QoS και του QoE και από μια γενικότερη άποψη, σχέση με το ίδιο περιεχόμενο αν προβληθεί σε κινητό τηλέφωνο. Αυτό που μπορούμε να εκμεταλλευτούμε είναι ότι η φορητή συσκευή πιθανόν να έχει χαμηλότερη ανάλυση, διαφορετικό βάθος χρώματος αλλά και εκτός αυτού να μην μπορεί να διαχειριστεί τέτοιο όγκο πληροφοριών. Για παράδειγμα, μια συσκευή που δεν υποστηρίζει υψηλής ανάλυσης βίντεο, θα πρέπει πρώτα να λάβει το περιεχόμενο, μετά να το επεξεργαστεί, να γίνει μείωση κλίμακας στην ποιότητα και μετά να το προβάλει. Με αυτόν τον τρόπο έχουμε σπατάλη και στον όγκο δεδομένων αφού τελικά δεν χρησιμοποιείται όλο το υλικό αλλά και σε επεξεργαστική ισχύ της συσκευής, αφού προκειμένου να κάνει την μετατροπή αυτή, με ότι αυτό συνεπάγεται (αυξημένη κατανάλωση ή και καθυστέρηση στην εκτέλεση παράλληλων εργασιών). Έτσι, με τον κατάλληλο προγραμματισμό και τους κατάλληλους μηχανισμούς μπορεί να γίνει οικονομία στη διαχείριση των πόρων αφού δεν είναι ανάγκη να μεταδίδεται το σύνολο της πληροφορίας.

### **2.9. Βαθμός υπηρεσίας GoS (Grade of Service) – Ποιότητα προσαρμοστικότητας QoR (Quality of Resilience)**

Πέρα από το QoS, μπορεί να γίνει περαιτέρω διαχωρισμός σε επίπεδο δικτύου, ανάλογα με την χρονική στιγμή της μετάδοσης. Με τον όρο Grade of Service (GoS), περιγράφεται το οτιδήποτε συμβαίνει από την αρχή της σύνδεσης, τη διατήρηση αλλά και το κλείσιμο της. Αρχικά εφαρμοζόταν σε απλές αναλογικές τηλεφωνικές κλήσεις, αλλά επεκτάθηκε και στα δίκτυα της επόμενης γενιάς. Οι παράμετροι για το GoS περιλαμβάνουν τον χρόνο καθυστέρησης για τη δημιουργία της σύνδεσης, την πιθανότητα μη πραγματοποίησης σύνδεσης, την πιθανότητα να χαθεί μια σύνδεση κ.α. Όλα αυτά, μπορούν να παίξουν άμεσα καθοριστικό ρόλο στο τελικό QoE που λαμβάνει ο χρήστης, για παράδειγμα σε περίπτωση μετάδοσης βίντεο, έχουμε τον χρόνο καθυστέρησης για την έναρξη της αναπαραγωγής. Δυσανεμία από τον χρήστη θα έχουμε και στην περίπτωση που γίνεται χρήση υπηρεσιών IPTV και υπάρχει αναμονή κατά την εναλλαγή του τηλεοπτικού προγράμματος (zapping). Το Quality of Resilience (QoR) περιγράφει την ικανότητα του δικτύου να παραμείνει ζωντανό, δηλαδή περιλαμβάνει τον χρόνο επανασύνδεσης μετά από τυχόν διακοπή, ή τη διαθεσιμότητα. Σε περίπτωση όπου έχουμε διακοπή και αποκατάσταση από διαφορετική διαδρομή, θα πρέπει να εξεταστεί αν η νέα διαδρομή προσφέρει το ίδιο επίπεδο QoS και QoE, καθώς επίσης και αν έχει επηρεαστεί μόνο η συγκεκριμένη μετάδοση ή έχει επέλθει κάποιο πρόβλημα από την επιπλέον κίνηση που έχει προστεθεί στους υπόλοιπους κόμβους του δικτύου. Τέλος, θα πρέπει να υπολογιστεί και πόση πληροφορία έχει χαθεί κατά την όλη διαδικασία αποκατάστασης. Η κατηγοριοποίηση του QoR με βάση τους μηχανισμούς επανασύνδεσης μπορεί να γίνει ως εξής:

- ❖ Η μέθοδος επιλογής νέας διαδρομής μπορεί να έχει συμπεριληφθεί στο αρχικό σενάριο ή να υπολογιστεί εκείνη τη στιγμή.
- ❖ Οι πόροι για την επανασύνδεση μπορεί να είναι δεσμευμένοι από την αρχή, να μοιράζονται ή μπορεί να μην υπάρχουν και καθόλου.
- ❖ Τα επίπεδα που μπορεί να έχει η διαδικασία της επανασύνδεσης.
- ❖ Η διαδικασία επανασύνδεσης μπορεί να λάβει χώρα με βάση ολόκληρο το δίκτυο, ή κάνοντας χρήση ενός μέρους του ή ακόμα και σε τοπικό επίπεδο.

- ❖ Τομείς όπου θα γίνει η επιχείρηση της επανασύνδεσης.

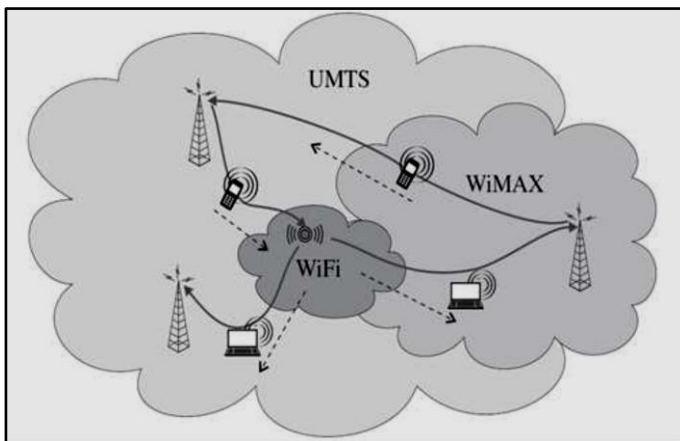
Γενικά, η διασφάλιση υψηλού QoR επηρεάζει το QoS και το GoS και κατ' επέκταση την τελική εμπειρία του χρήστη, το QoE. Αν δεν έχει μελετηθεί σωστά, έχει άμεσο αντίκτυπο, καθώς οι χρήστες θα παρατηρήσουν διακοπές στη σύνδεσή τους, οι ιστοσελίδες και τα e-mail δεν θα είναι προσβάσιμα και θα τερματίζονται οι μεταδόσεις πολυμέσων. Αν έχουμε αργή επανασύνδεση, το πρόβλημα θα περιοριστεί στις real-time εφαρμογές. Τέλος, ακόμα και στην περίπτωση ταχύτατης επανασύνδεσης, είναι πιθανό να παρουσιαστούν προβλήματα καθώς με το νέο μονοπάτι που θα ακολουθούν τα πακέτα, έχουμε αλλαγή και στις παραμέτρους των QoS και GoS.



**Εικόνα 8 : Παράγοντες που σχετίζονται με το QoE.**

## 2.10. Βασικές απαιτήσεις του χρήστη

Οι απαιτήσεις του χρήστη επεκτείνονται σε όλες τις υπηρεσίες, από απλό browsing και χρήση



**Εικόνα 9: Αναπαράσταση της συνδεσιμότητας στα σύγχρονα δίκτυα κινητών επικοινωνιών.**

email, μέχρι και μετάδοση IPTV, video conference ακόμα σε τηλεϊατρική. Οι υπηρεσίες πολυμέσων είναι αυτές που έχουν και τις περισσότερες απαιτήσεις σε πόρους δικτύου. Σίγουρα θα υπάρχει η απαίτηση για πρόσβαση σε υπηρεσίες από οποιοδήποτε μέρος, είτε αυτό είναι σε χώρο εργασίας ή στην οικία του, ανεξάρτητα τον τρόπο σύνδεσης. Για τις περιπτώσεις ασύρματης σύνδεσης όμως, θα πρέπει να παρέχονται οι υπηρεσίες αδιάκοπτα ανεξάρτητα από το δίκτυο που χρησιμοποιείται, γεγονός που αποτελεί πρόκληση, καθώς διάφορες τεχνολογίες δικτύου έχουν και διαφορετικούς μηχανισμούς QoS, και ανάλογα τον πάροχο υπάρχουν και διαφορετικά standards. Επίσης, για χρήστες που βρίσκονται σε κίνηση, η υποστήριξη του handover παίζει σημαντικό ρόλο. Για παράδειγμα, οι παλαιότερες τεχνολογίες WLAN (IEEE 802.11 a/b/g) δεν υποστηρίζουν το handover, καθώς για να γίνει η μετάβαση, θα πρέπει να έχουμε πρώτα διακοπή και μετά νέα σύνδεση, κάτι που θα έχει ως αποτέλεσμα την



μερική απώλεια πακέτων δεδομένων. Αντίθετα, σε δίκτυα κινητής τηλεφωνίας, GSM, 3G και LTE, το handover είναι κοινό χαρακτηριστικό, τόσο το οριζόντιο δηλαδή στην ίδια τεχνολογία δικτύου, όσο και στο κάθετο όπως για παράδειγμα μεταξύ 3G και GSM. Επίσης, πέρα από το σημείο και τον τρόπο που θα συνδεθεί ο χρήστης, υπάρχει απαίτηση για πρόσβαση στις υπηρεσίες οποιαδήποτε ώρα. Θα πρέπει να έχουμε υπολογίσει την δυνατότητα σύνδεσης σε ώρες αιχμής, την πιθανότητα να χαθεί η σύνδεση αλλά και να γίνει μια νέα και γενικότερα να έχουν εξεταστεί όλοι οι τομείς που αφορά το QoR (Quality of Resilience). Απαιτείται η ύπαρξη εφεδρικών πόρων για το δίκτυο οι οποίοι θα μπορούν να αξιοποιηθούν κάθε στιγμή, αλλά και η σχεδίαση πιθανών εναλλακτικών διαδρομών για τα μεταφερόμενα πακέτα. Επιπρόσθετα, θα πρέπει να έχει ληφθεί υπόψη ότι σε πιθανό πρόβλημα σε κόμβους που δεν χρησιμοποιούνται από την συγκεκριμένη συνεδρία, μπορεί να επηρεάσει αρνητικά, καθώς μπορεί να απαιτηθούν πόροι για την κάλυψη της έκτακτης αυτής ανάγκης. Όμως, πέρα από τις δυνατότητες και την ποιότητα του δικτύου, η αντίληψη της ποιότητας από τον χρήστη, θα γίνει μέσω της συσκευής από την οποία θα γίνει η χρήση της υπηρεσίας. Η συσκευή αυτή μπορεί να είναι οποιαδήποτε, από την τηλεόραση, τον σταθερό υπολογιστή μέχρι και κάποιο tablet ή κινητό τηλέφωνο. Όλες αυτές οι συσκευές διαφέρουν πολύ σε μέγεθος οθόνης, ανάλυσης, βάθος χρώματος, φωτεινότητα, αντίθεση, χρόνο απόκρισης. Ακόμα είναι διαφορετικές και οι δυνατότητες κάθε συσκευής σε επεξεργαστική ισχύ, μέγεθος μνήμης ή κατανάλωση ενέργειας. Για παράδειγμα, μια συσκευή που δεν υποστηρίζει υψηλής ανάλυσης βίντεο, θα πρέπει πρώτα να λάβει το περιεχόμενο, μετά να το επεξεργαστεί, να γίνει μείωση στην κλίμακα της ποιότητας και μετά να το προβάλει. Με αυτόν τον τρόπο έχουμε σπατάλη και στον όγκο δεδομένων αφού τελικά δεν χρησιμοποιείται όλο το υλικό αλλά και σε επεξεργαστική ισχύ της συσκευής, αφού προκειμένου να κάνει την μετατροπή αυτή, με ότι αυτό συνεπάγεται (αυξημένη κατανάλωση, ή καθυστέρηση στην εκτέλεση παράλληλων εργασιών). Έτσι, με τον κατάλληλο προγραμματισμό και τους κατάλληλους μηχανισμούς μπορεί να γίνει οικονομία στη διαχείριση των πόρων αφού δεν είναι ανάγκη να μεταδίδεται το σύνολο της πληροφορίας.

**Κεφάλαιο 3<sup>ο</sup>**  
**Ψηφιακή Επεξεργασία**  
**και Ανάλυση Εικόνας.**

**3.1. Η ανάγκη για ψηφιακή συμπίεση δεδομένων.**

Με δεδομένη την ολοένα και αυξανόμενη χρήση της ψηφιακής τεχνολογίας, η οποία αντικαθιστά την αναλογική που χρησιμοποιείται εδώ και δεκαετίες στις τηλεπικοινωνίες, την τηλεόραση, την ηχογράφηση και αναπαραγωγή της μουσικής αλλά και σε άλλους τομείς, έχει αρχίσει παράλληλα να γίνεται έρευνα και προς την κατεύθυνση της συμπίεσης της ψηφιακής πληροφορίας με στόχο την οικονομία εύρους φάσματος (bandwidth). Είναι φανερό ότι η ποιοτική ψηφιοποίηση κάποιου μέσου που απαιτεί υψηλή συχνότητα δειγματοληψίας και μεγάλο μέγεθος δείγματος θα δημιουργούσε ένα εξίσου μεγάλο αρχείο. Επομένως, για αρχεία με μεγάλο πληροφοριακό περιεχόμενο, εντοπίζονται τα ακόλουθα προβλήματα:

- ❖ Έλλειψη κατάλληλου αποθηκευτικού χώρου.
- ❖ Χαμηλή σε ρυθμό μεταφορά της πληροφορίας για αναπαραγωγή στα αντίστοιχα συστήματα.
- ❖ Μη ικανοποιητική μεταφορά τους στο διαδίκτυο.

**Σχόλια-Παρατηρήσεις.**

- ❖ Έτσι για μια εικόνα 1024 \* 768 pixel με βάθος χρώματος 24 bit δημιουργεί αρχείο  $1024*768*24=18874368 \text{ bit} = 18 \text{ Mb} = 2,25 \text{ MB}$
- ❖ Ένα ασυμπίεστο video με 24 frame το δευτερόλεπτο χρειάζεται αποθηκευτικό χώρο  $2,25 \text{ MB} * 24 \text{ (frame)} = 54 \text{ MB}$
- ❖ Και για μια ταινία μιας ώρας απαιτείται χώρος  $54\text{MB} * 3600 \text{ (sec)} = 190 \text{ GB}$  περίπου.

Τέτοιου μεγέθους αρχεία δημιουργούν πρόβλημα και πρέπει με κάποιο τρόπο να συμπειστούν ώστε να μην αποθηκεύονται πιο εύκολα, αλλά να μεταφέρονται και να αναπαράγονται εύκολα και γρήγορα.

**3.2. Λόγος συμπίεσης – Ρυθμός συμπίεσης.**

Στόχος της συμπίεσης μπορεί να είναι είτε ο περιορισμός του μεγέθους ενός αρχείου είτε ο περιορισμός του ποσού πληροφορίας που πρέπει να μεταδοθεί. Αν ενδιαφερόμαστε για τον περιορισμό του αποθηκευτικού χώρου αναφερόμαστε στον λόγο συμπίεσης και κάνουμε λόγο για ένα μέγεθος που εκφράζει την σχέση του χώρου που καταλαμβάνουν τα συμπιεσμένα δεδομένα. Ο λόγος συμπίεσης είναι καθαρός αριθμός και δίνεται από την παρακάτω σχέση:

$$\text{Λόγος Συμπίεσης} = (\text{Αρχικό Μέγεθος Δεδομένων}) / (\text{Μέγεθος Συμπιεσμένων Δεδομένων})$$

Λόγος συμπίεσης 3 : 1 σημαίνει ότι μετά την συμπίεση το μέγεθος του αρχείου είναι το 1/3 του αρχικού. Αν μας ενδιαφέρει η οικονομία που κάνουμε στην χρήση καναλιού τότε αναφερόμαστε στον ρυθμό συμπίεσης που εκφράζει τον ρυθμό μετάδοσης των συμπιεσμένων δεδομένων και μετριέται συνήθως σε bits per second , bits per sample , bits per pixel .

**3.3. Συμπίεση Δεδομένων.**

Ένα σημαντικό ερώτημα που τίθεται είναι αν μπορεί η πληροφορία να κωδικοποιηθεί πιο αποδοτικά, δηλαδή να βρεθεί ένας τρόπος να μεταδοθεί η πληροφορία με λιγότερα bit αλλά παράλληλα με την ίδια ποιότητα. Συμπερασματικά, καταλήγουμε πως η συμπίεση είναι μια διαδικασία που κωδικοποιεί ξανά την εκπεμπόμενη πληροφορία με στόχο να μεταδώσει το ίδιο περιεχόμενο αλλά με λιγότερα bits. Αυτή η διαδικασία, εκμεταλλεύεται κανόνες και περιορισμούς, προσπαθεί να εκμεταλλευτεί κάθε πλεονασμό που υπάρχει στην αρχική πληροφορία. Παρόλα αυτά, δεν είναι ορθό να συγχέουμε έναν αλγόριθμο συμπίεσης (compression algorithm) και έναν συμπιεστή (codec or compressor). Στην πρώτη κατηγορία κυρίαρχος στόχος αποτελεί η επίτευξη συμπίεσης των δεδομένων εισόδου, πράγμα που επιτυγχάνεται εφαρμόζοντας μια απλή και

συγκεκριμένη μεθοδολογία επεξεργασίας των δεδομένων έτσι ώστε να προκύψει μια περισσότερο συμπίεσμένη μορφή τους. Ενώ, η δεύτερη κατηγορία, αυτή του συμπίεστη, παρατηρείται μια διαδικασία συμπίεσης και αποσυμπίεσης, περιλαμβάνοντας περισσότερους από έναν αλγόριθμους στην σειρά. Ειδικότερα, τα δεδομένα εξόδου του ενός αποτελούν είσοδο για τον επόμενο. Γνωστοί τέτοιοι συμπίεστες που εφαρμόζονται σε πληροφορία εικόνας και video είναι οι jpeg και mpeg, βασιζόμενοι στον "Διακριτό μετασχηματισμό συνημίτονου" (Discrete Cosine Transform). Η συμπίεση μπορεί να μειώσει τον όγκο των δεδομένων που στέλνονται ή αποθηκεύονται, με την ελαχιστοποίηση του ενυπάρχοντος πλεονασμού. Ο πλεονασμός παρουσιάζεται κατά τη δημιουργία των δεδομένων. Με τη διαδικασία της συμπίεσης η μεταφορά και η αποθήκευση γίνονται με πιο αποδοτικό τρόπο, ενώ παράλληλα διατηρείται η ακεραιότητα των δεδομένων.

### **3.3.1. Παράγοντες που σχετίζονται με την συμπίεση.**

Στόχος της συμπίεσης video είναι να μειώσει ένα ποσοστό της απαιτούμενης πληροφορίας, ώστε να μπορεί να αποθηκευτεί το ψηφιακό σήμα, διατηρώντας βέβαια όσο είναι δυνατόν την ποιότητα του αρχικού video. Σύμφωνα λοιπόν με τα παραπάνω, υπάρχουν σημαντικοί παράγοντες που πρέπει να ληφθούν υπόψη κατά την αναφορά μας στην ψηφιακή συμπίεση video.

- ❖ **Πραγματικός - Μη Πραγματικός χρόνος**: Μερικά συστήματα συμπίεσης συλλαμβάνουν, συμπιέζουν, αποσυμπιέζουν, και αναπαράγουν το video στα 30 fps, και όλα αυτά σε πραγματικό χρόνο, χωρίς καμία καθυστέρηση. Άλλα πάλι συστήματα είναι μόνο σε θέση να κάνουν σύλληψη μερικά από τα 30 fps και ή είναι σε θέση να αναπαράγουν μόνο μερικά από αυτά. Η ανεπαρκής συχνότητα ανανέωσης είναι ένα από τα πιο αξιοσημείωτα ελαττώματα του video. Κάτω των 24 πλαισίων το δευτερόλεπτο (fps), το video θα είναι αισθητά απότομο. Επιπλέον τα πλαίσια που παραλείπονται περιέχουν υπερβολικά σημαντικές πληροφορίες συγχρονισμού των δεδομένων. Με άλλα λόγια, εάν η κίνηση των χειλιών ενός ατόμου έχει χαθεί εξαιτίας των παραλειπόμενων πλαισίων κατά την διάρκεια της σύλληψης ή της αναπαραγωγής θα είναι αδύνατον να συγχρονιστεί ο ήχος σωστά με το video. Άρα καταλαβαίνουμε ότι η έννοια του πραγματικού χρόνου έχει να κάνει με την συχνότητα ανανέωσης των πλαισίων.
- ❖ **Συμμετρική – Ασύμμετρη Συμπίεση**: Συμμετρική συμπίεση παρουσιάζεται όταν μπορεί κανείς να αναπαράγει ένα video μεγέθους 640 x 480 και με frame rate 30 fps, να κάνει σύλληψη και να το αποθηκεύσει με τις ίδιες τιμές. Εν αντιθέσει, στην περίπτωση της ασύμμετρης συμπίεσης δεν συμβαίνει αυτό. Εδώ, ο βαθμός ασυμμετρίας συνήθως εκφράζεται ως μια αναλογία. Η ασύμμετρη συμπίεση μπορεί να είναι πιο περίπλοκη και πιο αποδοτική όσον αφορά την ποιότητα και την ταχύτητα κατά την αναπαραγωγή για τον λόγο ότι χρησιμοποιείται πολύ περισσότερος χρόνος όταν συμπιέζεται το video. Τέλος, η συμπίεση αυτή χαρακτηρίζεται και ως χρονοβόρα.
- ❖ **Αναλογίες Συμπίεσης**: Η αναλογία της συμπίεσης αντιπροσωπεύει αριθμητικά τα αρχικό video σε σύγκριση με το συμπίεσμένο. Όσο πιο πολύ συμπιέζεται το video, τόσο πιο υψηλή είναι η αναλογία συμπίεσης ή αλλιώς η αριθμητική διαφορά ανάμεσα στις δύο τιμές. Γενικά, όσο πιο υψηλή είναι η αναλογία συμπίεσης τόσο πιο χαμηλή θα είναι η ποιότητα της εικόνας. Οι αναλογίες συμπίεσης διαφέρουν όχι μόνο από μια μέθοδο συμπίεσης σε μια άλλη, αλλά και από το υλικό και λογισμικό που χρησιμοποιείται από έναν υπολογιστή για να γίνει η συμπίεση.
- ❖ **Απωλεστική - Μη απωλεστική Συμπίεση**: Οι τεχνικές συμπίεσης κατηγοριοποιούνται σε απωλεστικής (lossy) και μη απωλεστικής (lossless) συμπίεσης. Η διαφορά είναι απλή, στην συμπίεση χωρίς απώλειες η πληροφορία μετά την αποσυμπίεση είναι ίδια με την

αρχική ενώ στην απωλεστική συμπίεση η πληροφορία μετά την αποσυμπίεση είναι λιγότερη σε σχέση με την αρχική κάποια πληροφορία έχει χαθεί. Στην μη απωλεστική συμπίεση (lossless compression) διατηρείται η ακεραιότητα των δεδομένων. Τα αρχικά δεδομένα και τα δεδομένα μετά τη συμπίεση και την αποσυμπίεση είναι ακριβώς τα ίδια, επειδή σε αυτές τις μεθόδους ο αλγόριθμος συμπίεσης και ο αλγόριθμος αποσυμπίεσης είναι ακριβώς αντίστροφοι. Κατά τη διαδικασία δε χάνεται κανένα μέρος των δεδομένων. Τα πλεονάζοντα δεδομένα κωδικοποιούνται κατά τη συμπίεση και αποκωδικοποιούνται κατά την αποσυμπίεση. Αυτοί οι μέθοδοι χρησιμοποιούνται όταν δεν πρέπει να χαθεί ούτε ένα bit δεδομένων όπως στην περίπτωση ενός αρχείου κειμένου ή ενός προγράμματος. Τα γνωστά προγράμματα 7z , bz , zip και rar χρησιμοποιούν μη απωλεστική συμπίεση. Ευρέως διαδεδομένοι μη απωλεστική αλγόριθμοι είναι ο Run Length Encoding (RLE) , Huffman , Delta , LZW . Στην απωλεστική συμπίεση έχουμε απώλεια δεδομένων όπου μπορεί να μην είναι αποδεκτή σε αρχεία κειμένου ή σε προγράμματα είναι όμως αποδεκτή σε εικόνες και ταινίες. Ο λόγος είναι ότι τα μάτια μας και τα αφτιά μας δεν μπορούν να διακρίνουν πολύ μικρές αλλαγές. Για τέτοιες περιπτώσεις είναι κατάλληλες οι απωλέστηκες μέθοδοι συμπίεσης (lossy data compression). Οι μέθοδοι αυτές είναι οικονομικότερες και απαιτούν λιγότερο χρόνο και χώρο όταν πρέπει να σταλούν εκατομμύρια bit εικόνων και video το δευτερόλεπτο. Χαρακτηριστικό παράδειγμα απωλεστικής συμπίεσης εικόνας είναι η μέθοδος JPEG για video η μέθοδος και για ήχο το πρότυπο MP3.

### **3.4. Γενικές αρχές συμπίεσης video.**

#### **3.4.1. Ποιοτική ανοχή.**

Σε αντίθεση με την αντίληψη ότι η ψηφιακή μετάδοση θα πρέπει να έχει το εύρος ζώνης (bandwidth) που χρειάζεται για να μεταδώσει και το πιο απαιτητικό πλαίσιο (frame) μιας σειράς κινούμενων εικόνων, ακόμα και αν αυτό παρουσιάζεται με συχνότητα 1%, τώρα πια κάτω από την εμπορική πίεση που υπάρχει (λόγο κόστους) θεωρείται λογικό να δεχόμαστε κάποιο ποσοστό παραμόρφωσης σε τέτοιες σπάνιες σκηνές, με αντάλλαγμα να μπορούμε να χρησιμοποιήσουμε τα bit που εξοικονομούμε για την αναβάθμιση του μέσου όρου ανάλυσης του συνόλου των σκηνών. Οπότε, σε αντίθεση με την ηχογράφηση ήχου σε CD, που γίνεται χωρίς καμία συμπίεση, στην περίπτωση συμπίεσης video αναζητείται μία χρυσή τομή (sweet spot) ανάμεσα στην ποιότητα και το bandwidth που πολλές φορές βασίζεται σε υποκειμενικές μετρήσεις που γίνονται σε άτομα που θεωρούνται ικανά "δείγματα" πάνω στην εκτίμηση της εικόνας και του ήχου λόγω επαγγελματικής εμπειρίας όπως θα αναφέρουμε μερικά παρακάτω. Αν η εμπειρία τους δεν τους επιτρέπει να αντιληφθούν σημαντικές διαφορές στην ποιότητα του κωδικοποιημένου ήχου από τον αρχικό και συγχρόνως θεωρούν ανεκτές τις ατέλειες (artifacts) της εικόνας μετά από κάποιο ποσοστό συμπίεσης, η μέθοδος και το ποσοστό αυτό θεωρούνται αποδεκτά και τυποποιούνται. Να σημειωθεί εδώ ότι ενώ ο ήχος προσφέρεται για μεγάλα ποσοστά συμπίεσης χωρίς να μπορεί να γίνει αισθητή υποβάθμιση στην ποιότητά του, στον τομέα της εικόνας τα πράγματα είναι λίγο πιο δύσκολα και οι όποιες ατέλειες είναι ορατές, απλά γίνεται προσπάθεια να μην γίνονται ενοχλητικές για το θεατή.

#### **3.4.2. Πλεονασμός (Redundancy).**

Αν σκεφτούμε ότι για κάθε ένα δευτερόλεπτο μιας ταινίας video χρειάζονται περίπου 30 εικόνες, καθεμία από τις οποίες απαιτεί ίσως και 1 MB μνήμης, είναι προφανές ότι ένα αρχείο video χρειάζεται τεράστιο όγκο δεδομένων. Η ποσότητα της πληροφορίας ενός αρχείου video εξαρτάται από την ποιότητα της εικόνας και τον αριθμό των καρτέ ανά δευτερόλεπτο που απαιτούνται για την σωστή αναπαραγωγή κίνησης. Η ανάγκη που υπάρχει για την συμπίεση των δεδομένων ενός αρχείου είναι επιτακτική όχι μόνο για τον όγκο του αρχείου, αλλά και για τους λόγους διακίνησης του σε τακτά χρονικά διαστήματα. Η βασική αρχή πάνω στην οποία στηρίζονται όλες οι μέθοδοι ψηφιακής συμπίεσης είναι το γεγονός ότι το σήμα περιέχει ένα

ποσοστό πλεονασμού (Redundancy), δηλαδή πληροφορία που είτε μπορεί να παραληφθεί, είτε να κωδικοποιηθεί με λιγότερη ακρίβεια, χωρίς αυτό να έχει αξιοσημείωτη επίδραση στο τελικό αποτέλεσμα. Υπάρχουν δύο είδη πλεονασμού:

- ❖ **Στατικός πλεονασμός (Spatial Redundancy):** Το επίπεδο του σήματος μπορεί σε κάθε δεδομένη χρονική στιγμή να προϋπολογιστεί από την προηγούμενη τιμή του, διότι οι τιμές δειγμάτων της εικόνας σε μεγάλο ποσοστό σχετίζονται μεταξύ τους. Αυτό επαληθεύεται και εποπτικά για παράδειγμα, σε μία εικόνα του δελτίου ειδήσεων ένα μεγάλο κομμάτι της εικόνας (φόντο) παραμένει αμετάβλητο και μόνο το κομμάτι της εικόνας που καταλαμβάνει ο παρουσιαστής μεταβάλλεται ελαφρά (κινήσεις των χειλιών, των βλεφάρων). Έτσι μπορούμε να υπολογίσουμε ένα τμήμα της εικόνας από μία προηγούμενη και να προσθέσουμε απλά τις διαφορές που έχουν προκύψει χωρίς να χρειάζεται να κωδικοποιούμε σε κάθε frame την πλεονάζουσα πληροφορία.
- ❖ **Υποκειμενικός Πλεονασμός (Temporal Redundancy):** Ανάλογα με το περιεχόμενο της εικόνας το ανθρώπινο μάτι μπορεί να ανεχτεί ένα ποσοστό παραμόρφωσης ή αλλοίωσης ορισμένων παραμέτρων της εικόνας χωρίς αυτό να γίνει αντιληπτό. Για παράδειγμα, είναι διαδεδομένο ότι η ανθρώπινη όραση είναι πολύ πιο ευαίσθητη στη φωτεινότητα της εικόνας παρά στα χρώματα. Αντίστοιχες ιδιότητες έχει και η ακοή. Άρα, μπορούμε να αφιερώσουμε λιγότερο από το διαθέσιμο εύρος ζώνης στην περιγραφή των χρωμάτων και γενικά της πλεονάζουσας πληροφορίας και αυτό να περάσει απαρατήρητο από το θεατή.

### **3.5. Δειγματοληψία.**

Για να ψηφιοποιηθεί ένα σήμα από αναλογικό σε ψηφιακό, απαιτούνται τρία βήματα. Σε πρώτο στάδιο, πραγματοποιείται η δειγματοληψία του σήματος, δηλαδή από το πλήθος τιμών του συνεχούς σήματος κρατάμε μόνο ένα σύνολο διακριτών τιμών. Εν συνέχεια έχουμε κβαντοποίηση και κωδικοποίηση του σήματος. Βασικό κανόνα για την δειγματοληψία οποιουδήποτε αναλογικού σήματος αποτελεί το θεώρημα Nyquist-Shannon, όπου σύμφωνα με αυτό το θεώρημα της δειγματοληψίας, για να πάρουμε κάποια δείγματα από το αναλογικό video χρησιμοποιούμε μια προκαθορισμένη συχνότητα όπου θα πρέπει να ικανοποιεί το θεώρημα Nyquist. Το θεώρημα αυτό βασίζεται στον κανόνα ότι για την μετατροπή ενός αναλογικού σήματος σε ψηφιακό του οποίου η υψηλότερη συχνότητα είναι  $x$  ποσότητα Hertz, τότε για την συχνότητα δειγματοληψίας απαιτεί τουλάχιστον  $2 \times$  ποσότητα Hertz. Σε περίπτωση που δεν συμβεί αυτός ο διπλασιασμός της συχνότητας, τότε θα παρατηρηθεί το φαινόμενο aliasing. Υπάρχουν τρία βασικά μεγέθη που μπορούν και επηρεάζουν τον όγκο των ψηφιακών δεδομένων που θα παραχθούν. Αυτά είναι η ανάλυση της εικόνας, το βάθος χρώματος και ο αριθμός των πλαίσιων. Στόχος της δειγματοληψίας είναι να μειώσει τη διάσταση του video στην οριζόντια ή κάθετη διεύθυνση και ουσιαστικά τον αριθμό των pixels που θα κωδικοποιηθούν πριν την διαδικασία της κωδικοποίησης.

### **3.6. Κβαντοποίηση.**

Η μέθοδος που συμβάλλει στο να απαλλαγούμε από σημαντικό μέρος της πληροφορίας είναι η κβαντοποίηση. Με τον όρο αυτό, εννοούμε τη μετατροπή ενός σήματος άπειρων (η πάρα πολλών) τιμών σε ένα σήμα ορισμένων διακριτών τιμών. Είναι προφανές ότι η κβαντοποίηση εισάγει σφάλμα ανάλογο με τον αριθμό των bits που απορρίπτονται και κατά συνέπεια ευθύνεται στο μεγαλύτερο βαθμό για την απώλεια πληροφορίας κατά τη συμπίεση (lossy compression) σε αντίθεση με το μετασχηματισμό DCT που είναι μία, σε μεγάλο βαθμό αντιστρεπτή διαδικασία. Για να περιγράψουμε όσο περισσότερες από τις τιμές του σήματος γίνεται με δοσμένο αριθμό bits, διαιρούμε τις τιμές των δειγμάτων είτε με σταθερές τιμές (uniform Quantization) είτε με πίνακες κβαντοποίησης (Quantization tables). Στη συγκεκριμένη περίπτωση, δηλαδή στο πρότυπο MPEG χρησιμοποιείται η δεύτερη μέθοδος και μάλιστα υπάρχει ένας πίνακας για τα πλαίσια που έχουν κωδικοποιηθεί με ενδοπλασιακή (intra-frame coding) και ένας για αυτά με διαπλασιακή (inter-frame coding) όπως θα δούμε παρακάτω.

Εν ολίγοις, η κβαντοποίηση είναι η μετατροπή ενός μεγέθους από συνεχές σε ψηφιακό, δηλαδή σε κάτι που μετρείται σε διακριτά βήματα και όχι με συνεχή τρόπο. Άρα είναι η διαδικασία που μετατρέπει ένα σήμα από το χώρο των συνεχών τιμών στο χώρο των διακριτών. Αυτό πραγματοποιείται με το να υποδιαιρεθεί το εύρος τιμών του συνεχούς σήματος σε συγκεκριμένα επίπεδα και σε κάθε επίπεδο να εκχωρείται ένας ψηφιακός κωδικός.

### **3.7. Κωδικοποίηση.**

Υλοποιώντας τις διαδικασίες δειγματοληψίας και κβαντοποίησης, ένα συνεχές σήμα μετατρέπεται σε ένα διακριτό σύνολο τιμών. Στην συνέχεια χρησιμοποιείται η διαδικασία της κωδικοποίησης για τη μετατροπή του διακριτού συνόλου των τιμών των δειγμάτων σε μια πιο κατάλληλη μορφή. Κάθε σχέδιο για την αναπαράσταση καθενός από αυτά τα διακριτά σύνολα τιμών σαν μια ιδιαίτερη διάταξη διακριτών γεγονότων ονομάζεται κώδικας (code). Ένα από τα διακριτά γεγονότα σε ένα κώδικα ονομάζεται στοιχείο του κώδικα (code element) ή σύμβολο (symbol). Δύο είδη κωδικοποίησης που θα αναφερθούν παρακάτω είναι η ενδοπλαισιακή (intraframe) και διαπλαισιακή (interframe). Έτσι σε πολλές περιπτώσεις (πιο πολύπλοκες υλοποιήσεις) ο κωδικοποιητής μπορεί να επιλέξει το είδος κωδικοποίησης ανάλογα με τις ανάγκες της εικόνας. Δηλαδή οι δύο τρόποι κωδικοποίησης χρησιμοποιούνται περιοδικά για να φιλτράρουν την εικόνα από αυξημένες παραμορφώσεις, ή στην αντίθετη περίπτωση να ελαττώσουν το ρυθμό μετάδοσης. Αυτός είναι και ο λόγος που στο MPEG-2 εμφανίζεται μεταβαλλόμενος ρυθμός μετάδοσης.

### **3.8. Γνωριμία με την ψηφιακή εικόνα.**

Με τον όρο ψηφιακή εικόνα, εννοούμε την μετάβαση από τον αναλογικό κόσμο στον ψηφιακό, δηλαδή τη μετατροπή των αναλογικών σημάτων σε ψηφιακά. Έτσι, μια πραγματική εικόνα μεταφέρεται στον ψηφιακό κόσμο με τη μορφή διακεκριμένου σήματος που έχει τη μορφή ψηφιακών πινάκων. Μια ψηφιακή εικόνα μπορεί να είναι δυαδική (binary images), μονοχρωματική αποχρώσεων του γκρι (gray-level ή grayscale images) ή έγχρωμη (color image). Η απλούστερη μορφή μιας εικόνας, είναι η δυαδική μορφή. Μια δυαδική εικόνα, έχει μόνο δυο στάθμες φωτεινότητας που συνήθως είναι το μαύρο και το άσπρο. Το μαύρο αντιστοιχεί στην τιμή 0 και το άσπρο στην τιμή 1 (ή σε αντίστοιχη με grayscale εικόνες στην τιμή 255). Μια δυαδική εικόνα, επίσης, καταλαμβάνει μικρότερη μνήμη και η επεξεργασία της απαιτεί μικρότερο υπολογιστικό κόστος. Σε δυαδική μορφή μπορούν να απεικονισθούν σημαντικές πληροφορίες, όπως είναι το εμβάδον και η θέση αντικειμένων, ή η μορφή αντικειμένων. Οι έγχρωμες εικόνες, αποτελούν το μέσο για την απεικόνιση του πραγματικού κόσμου. Μια έγχρωμη ψηφιακή εικόνα, αποτελείται από τρεις gray-level εικόνες. Δηλαδή, το χρώμα κάθε εικονοστοιχείου, έχει τρεις συνιστώσες, οι οποίες αντιστοιχούν στις γκρι αποχρώσεις των αντίστοιχων εικονοστοιχείων των τριών γκρι εικόνων.

#### **3.8.1. Είδη ψηφιακών εικόνων.**

Υπάρχουν τρία είδη ψηφιακών εικόνων που χαρακτηρίζονται από το πλήθος των χρωμάτων που περιέχουν, και χωρίζονται ως εξής:

- ❖ **Στις δυαδικές εικόνες (binary images):** Σε αυτήν την περίπτωση, έχω μόνο δύο αποχρώσεις: μαύρα (0) και λευκά (1) pixels. Κάθε εικονοστοιχείο δηλαδή, μπορεί να χρωματιστεί με ένα από τα δυο χρώματα (μαύρο ή άσπρο), όπως επίσης απαιτείται ένα bit πληροφορίας, για παράδειγμα με τιμή 0 για το μαύρο και 1 για το άσπρο.
- ❖ **Στις εικόνες αποχρώσεων του γκρι (gray-level ή gray-scale images):** Εδώ, έχω  $2^8=256$  αποχρώσεις του γκρι. Από αυτές τις αποχρώσεις, συνήθως λαμβάνονται 256 αντιπροσωπευτικές, οι οποίες κωδικοποιούνται με τιμές από 0,1,2,...,255. Η απόχρωση κάθε εικονοστοιχείου, απαιτεί πληροφορία 8 bit.
- ❖ **Στις έγχρωμες εικόνες (color images):** Έχω 28 διαφορετικές αποχρώσεις του κόκκινου, 28 διαφορετικές αποχρώσεις του πράσινου και 28 διαφορετικές αποχρώσεις του μπλε

(Red, Green, Blue: RGB). Δηλαδή, για κάθε ένα από αυτά τα τρία χρώματα, λαμβάνονται 256 αποχρώσεις, δηλαδή πληροφορία 8 bit. Συνεπώς, κάθε εικονοστοιχείο της έγχρωμης εικόνας, απαιτεί 24 bit.

### **3.9. Ανάλυση και επεξεργασία ψηφιακής εικόνας.**

Η ανάλυση εικόνας, χρησιμοποιείται για την ποσοτικοποίηση των χαρακτηριστικών, τα οποία βρίσκονται σε διασπορά (σκόρπια), ώστε να προσδιοριστεί εάν και σε τι βαθμό διασποράς είναι ομογενείς ή ανομοιογενείς. Στην ανάλυση εικόνας, η τμηματοποίηση είναι μέρος μιας ψηφιακής εικόνας μέσα σε πολλαπλά πεδία (τμήματα pixels). Σκοπός της τμηματοποίησης είναι να εντοπίσει αντικείμενα τα οποία μας ενδιαφέρουν. Δυστυχώς, πολλοί σημαντικοί αλγόριθμοι είναι αρκετά απλοί ώστε να λύσουν αυτό το πρόβλημα επακριβώς: αντισταθμίζουν για τον συγκεκριμένο περιορισμό με την προβλεψιμότητα, την γενικότητα και την αποδοτικότητά τους.

#### **3.9.1. Βήματα για την Ανάλυση εικόνας.**

Η Επεξεργασία Εικόνας, εφαρμόζεται σύμφωνα με τα παρακάτω βήματα:

❖ **Βήμα 1<sup>ο</sup> Thresholding:** Είναι μια λειτουργία στη οποία μια grayscale εικόνα μετατρέπεται σε binary. Αυτό γίνεται για να απλοποιηθεί η εικόνα και στη συνέχεια να επεξεργαστεί. Η μετατροπή μιας εικόνας σε binary, γίνεται καθορίζοντας αρχικά ένα threshold level και αλλάζοντας έπειτα όλα τα pixels με grey level κάτω από το threshold level σε 0 (μαύρο) και τα υπόλοιπα με grey level ίσο ή μεγαλύτερο σε 255 (άσπρο). Το threshold level μπορεί να βρεθεί είτε με μαθηματικές μεθόδους (Automatic thresholding), είτε μελετώντας την εικόνα και το ιστογράμμά της (Manual thresholding). Δύο τύποι λαθών μπορούν να γίνουν όταν επιλέγεται το threshold level:

- ❖ **Τύπος 1:** Να μη συμπεριλαμβάνονται στη νέα εικόνα όλα τα επιθυμητά pixels.
- ❖ **Τύπος 2:** Κάποια από τα pixels που περιλαμβάνονται στη νέα εικόνα δεν έπρεπε να υπήρχαν.

Πριν επιλεγεί το threshold level θα πρέπει να αποφασιστεί ποιο είδος λάθους είναι το πιο αποδεκτό. Στα διάφορα project προτιμώνται λάθη τύπου 2 και χρησιμοποιείται automatic thresholding, σύμφωνα με το οποίο το threshold level ως εξής: μια εικόνα αποτελείται συχνά από δύο μέρη, το αντικείμενο και το background. Το grey level του αντικειμένου κατανέμεται από μια στατιστική κατανομή ενώ το grey level του background από μία άλλη. Αυτό μπορούμε να το δούμε ως δύο peaks του ιστογράμματος, το μέσο των οποίων ορίζεται ως threshold level. Αυτό υλοποιείται με τις παρακάτω εξισώσεις:

- ❖ **Για το grey level:**

$$T = \max \left\{ \frac{t(g)}{P - t(g)} \cdot (m(g) - m(G - 1))^2 \right\} - 1$$

<i>T</i>	<i>The threshold level</i>
<i>g</i>	<i>The grey level</i>
<i>P</i>	<i>Number of pixels in the image</i>
<i>G</i>	<i>The maximum number of grey levels</i>
<i>t(g)</i>	<i>The number of pixels at grey level g and below</i>
<i>m(g)</i>	<i>Mean grey level for the grey levels between zero and g</i>

**Εικόνα 9: Εξίσωση για το grey level.**

- ❖ **Για τον αριθμό των pixels με grey level g λιγότερο:**



$$t(g) = \sum_{i=0}^g f(i)$$

$f(i)$       The number of pixels at the grey level  $g$ .

Εικόνα 10: Αριθμός pixel για το grey level.

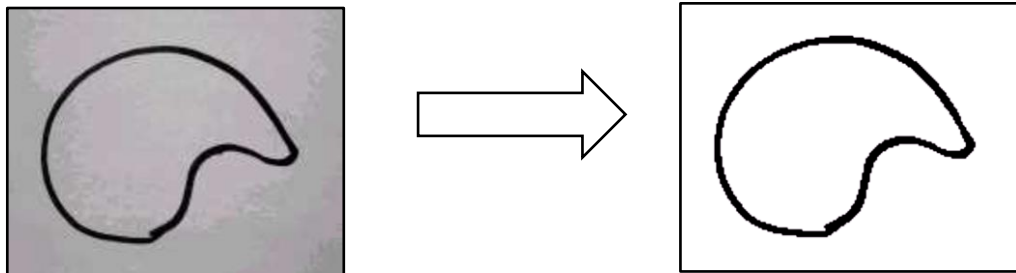
- ❖ Από τις εξισώσεις αυτές, προκύπτει το γράφημα που εικονίζεται παρακάτω:



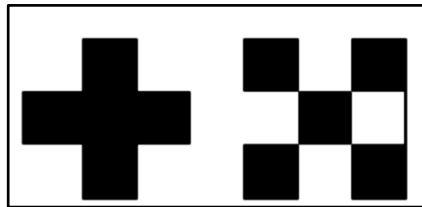
Εικόνα 11: Grey level γράφημα.

- ❖ **Σχόλια-Παρατηρήσεις.**

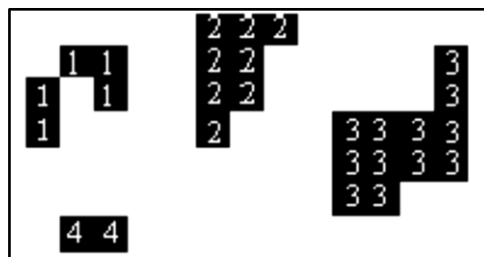
Συνεπώς threshold level = 86, επειδή είναι το μέγιστο. Το αποτέλεσμα του thresholding εικονίζεται παρακάτω, όπου από την ακατέργαστη PGM εικόνα προκύπτει η binary εικόνα στα δεξιά.



- ❖ **Βήμα 2° – Αφαίρεση περιγράμματος:** Στη συνέχεια όλα τα pixel που βρίσκονται στο περίγραμμα της εικόνας αλλάζουν το χρώμα τους στο χρώμα του background, γίνονται δηλαδή άσπρα. Με τον τρόπο αυτό, εξασφαλίζουμε την αποφυγή προβλημάτων όταν αργότερα εφαρμοστούν μάσκες, λαμβάνοντας υπόψη ότι με τη διαδικασία αυτή δε χάνεται ιδιαίτερα σημαντική πληροφορία.
- ❖ **Βήμα 3° – Component labeling:** Διαιρεί την binary εικόνα στα αντικείμενα που την αποτελούν. Έτσι, είναι δυνατόν να επιλέξουμε μεταξύ αντικειμένων διαφορετικού μεγέθους. Κάτι τέτοιο υλοποιείται σε πολλά project, όπου μετά το component labeling βρίσκουμε το μεγαλύτερο τμήμα στην εικόνα, υποθέτουμε ότι αυτό είναι το μονοπάτι και μετατρέπουμε τα pixel με διαφορετική ετικέτα από αυτή του μονοπατιού, στο χρώμα του background. Τα αντικείμενα στα οποία δίνεται η ίδια ετικέτα, είναι συνδεδεμένα. Υπάρχουν δύο διαφορετικοί ορισμοί για τη σύνδεση:
  - ❖ **Four connectedness:** Δύο pixel συνδέονται μόνο εάν έχουν κοινό όριο, όπως φαίνεται στην αριστερή εικόνα.
  - ❖ **Eight connectedness:** Τα pixels συνδέονται εάν ακουμπούν οι γωνίες της, όπως φαίνεται στη δεξιά εικόνα.

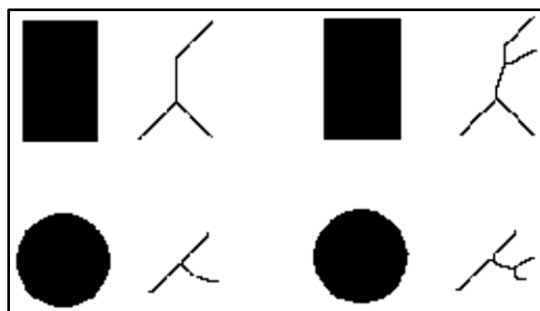


Όταν αποφασίζεται ποιος τύπος σύνδεσης θα χρησιμοποιηθεί, μπορεί να αρχίσει η διαδικασία της ετικετοποίησης. Η διαδικασία αυτή είναι μια επαναληπτική διαδικασία στην οποία ερευνάται κάθε pixel. Εάν το pixel έχει διαφορετικό χρώμα από το χρώμα του background, του δίνεται μια ετικέτα με έναν αριθμό. Πριν να γίνει αυτό, ερευνώνται τα γειτονικά pixel και αν ένα από αυτά έχει ήδη ετικέτα και συνδέεται με το pixel σύμφωνα με τον τύπο που καθορίστηκε, η ίδια ετικέτα δίνεται και στο pixel που ερευνάται. Το αποτέλεσμα είναι μια ετικέτα για κάθε αντικείμενο της εικόνας, όπως φαίνεται στο σχήμα.



**Εικόνα 12: Διαδικασία Ετικετοποίησης.**

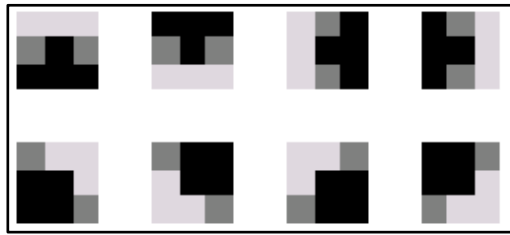
- ❖ **Βήμα 4<sup>ο</sup> – Skeletonization:** Είναι η λειτουργία κατά την οποία βρίσκεται ο μεσαίος άξονας των γραμμών ή των αντικειμένων στην εικόνα. Αυτό σημαίνει ότι μια γραμμή που είναι αρκετά pixel παχιά, ελαττώνεται σε γραμμή πάχους ενός μόνο pixel. Ο τρόπος με τον οποίο συνδέονται τα pixel στο σκελετό εξαρτάται από τον τύπο σύνδεσης που επιλέγεται για τη σκελετοποίηση. Παραδείγματα σκελετοποίησης φαίνονται στην εικόνα. Ένα πρόβλημα είναι ότι η διαδικασία της σκελετοποίησης μπορεί να επηρεαστεί από θόρυβο, όπως φαίνεται στα σχήματα που βρίσκονται στα δεξιά της εικόνας.



**Εικόνα 13: Παράδειγμα σκελετοποίησης.**

Η σκελετοποίηση ενός αντικειμένου, είναι μια διαδικασία στην οποία το εξωτερικό layer του απομακρύνεται με μια επαναλαμβανόμενη διαδικασία, η οποία ονομάζεται thinning.

- ❖ **Βήμα 5<sup>ο</sup> – Thinning:** Είναι μια διαδικασία στην οποία αφαιρείται το εξωτερικό layer ενός αντικειμένου, εφαρμόζοντας μάσκες σε όλα τα pixel στην επιφάνειά του. Αν η μάσκα ταιριάζει, το pixel στο κέντρο της διαγράφεται και η διαδικασία συνεχίζεται μέχρις ότου δοκιμαστούν όλα τα pixel.

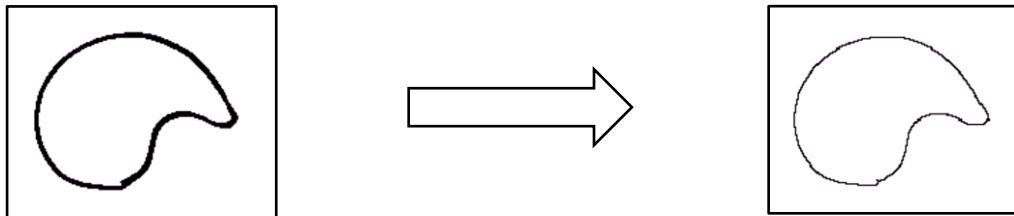


**Εικόνα 14: Μάσκες για thinning σε four connectedness.**



**Εικόνα 15: Χρήση μασκών για four connectedness.**

Τα μαύρα pixel είναι οι περιοχές του αντικειμένου, τα ανοιχτά γκρι έχουν το χρώμα του background και τα σκούρα γκρι είναι pixel, το χρώμα των οποίων δεν ενδιαφέρει. Για eight connectedness εφαρμόζονται ανάλογες μάσκες. Το αποτέλεσμα του thinning φαίνεται παρακάτω:



Μια εικόνα μετά το thinning έχει συχνά κάποια κλαδιά από pixel, λόγω θορύβου. Αυτά αφαιρούνται με τη διαδικασία pruning.

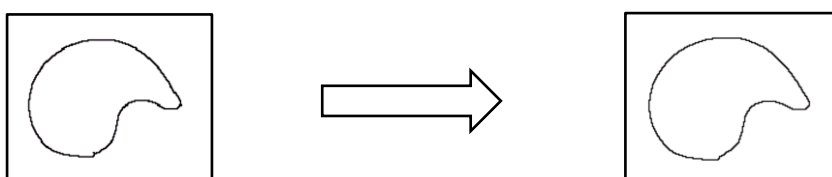
- ❖ **Βήμα 6<sup>ο</sup> – Pruning:** Είναι μια επαναληπτική διαδικασία για την αφαίρεση των κλάδων όπως σκελετού σε μια εικόνα, κατά την οποία αφαιρείται το ακριανό pixel του κάθε κλάδου σε κάθε επανάληψη. Η διαδικασία αυτή σταματάει όταν δε μπορούν πια να γίνουν άλλες αλλαγές στην εικόνα. Και γίνεται χρησιμοποιώντας μάσκες που εφαρμόζονται στην εικόνα.

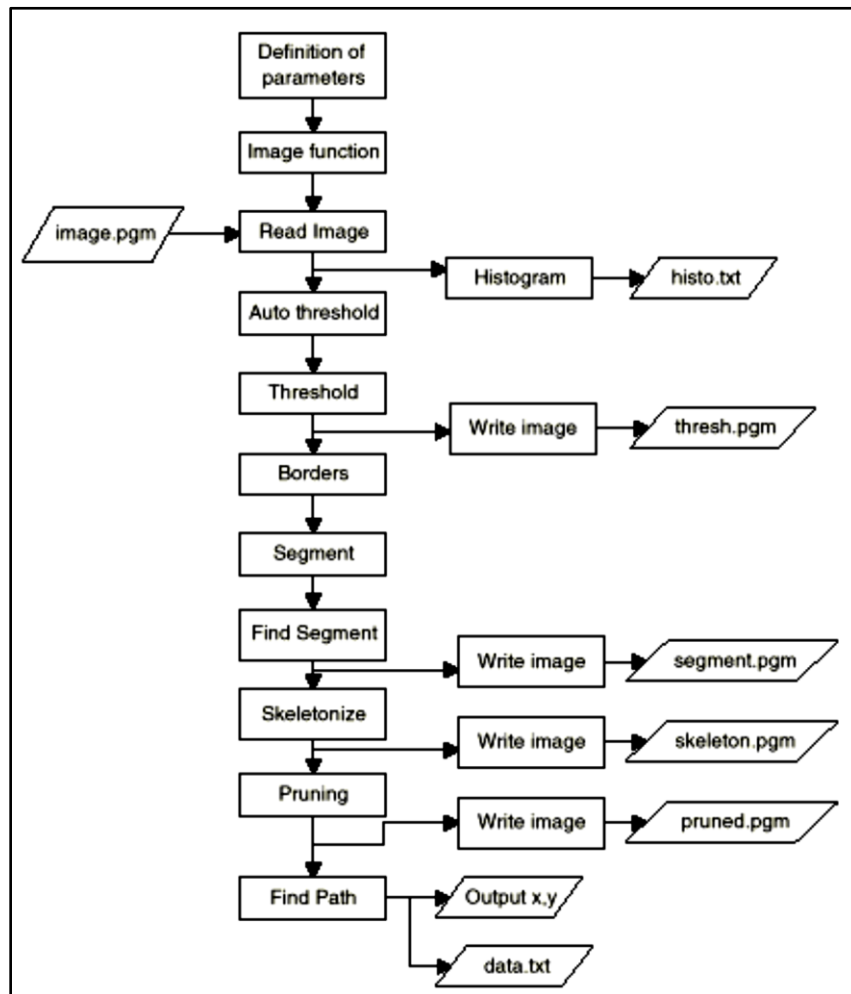
Υπάρχουν πάλι 2 σετ από μάσκες, ένα για four connectedness και ένα για eight connectedness.



**Εικόνα 16: Παράδειγμα αφαίρεσης κλάδων.**

Αποτέλεσμα του pruning όπως της εικόνας που προέκυψε παρακάτω με thinning.





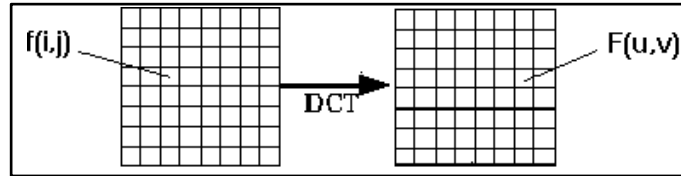
**Εικόνα 17: Επεξεργασία εικόνας προγραμματιστικά.**

### **3.10. Περί μετασχηματισμών σημάτων διακριτού χρόνου.**

Υπάρχει ένα πλήθος μετασχηματισμών που χρησιμοποιούνται στην ανάλυση των σημάτων και των συστημάτων. Οι πιο βασικοί είναι, ο μετασχηματισμός Fourier και ο μετασχηματισμός Z. Άλλοι μετασχηματισμοί είναι αυτοί του συνημιτόνου, του ημιτόνου, ο Sort Time Fourier Transform και τα κυματίδια (wavelets). Οι μετασχηματισμοί Fourier, ημιτόνου και συνημιτόνου, είναι μετασχηματισμοί των διακριτών σημάτων σε ορθοκανονικές και ορθομοναδιαίες βάσεις. Υπάρχει πλούσια βιβλιογραφία για τον ορισμό των μετασχηματισμών αυτών και την αυστηρή μαθηματική τους διερεύνηση. Εδώ θα εστιάσουμε στην ερμηνεία τους ώστε να βοηθηθεί ο αναγνώστης να αντιληφθεί γεωμετρικά και παραστατικά την βασική ιδέα από τις συνήθεις μαθηματικές εκφράσεις τους. Αυτό θα βοηθήσει σημαντικά στην δυνατότητά του να κατανοήσει την χρήση των μετασχηματισμών αυτών σε διάφορα πεδία (σχεδιασμός φίλτρων, συμπίεση) καθώς και να προχωρήσει στην κατανόηση της φύσης πιο σύνθετων μετασχηματισμών όπως τα κυματίδια.

### **3.11. Ο διακριτός μετασχηματισμός συνημιτόνου (DCT: Discrete Cosine Transform).**

Ο διακριτός μετασχηματισμός συνημιτόνου (DCT) βοηθά στη διάκριση της εικόνας σε μέρη (ή φασματικές υποζώνες) διαφορετικής σημασίας (σε σχέση με την οπτική ποιότητα της εικόνας). Το DCT είναι παρόμοιο με το διακριτό μετασχηματισμό Fourier: μετασχηματίζει ένα σήμα ή εικόνα από τη χωρική περιοχή στο πεδίο συχνότητας.



### 3.11.1. Κωδικοποίηση DCT.

Η γενική εξίσωση για ένα 1D ( N στοιχεία δεδομένων) DCT ορίζεται από την ακόλουθη εξίσωση:

$$F(u) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \Lambda(i) \cdot \cos \left[ \frac{\pi \cdot u}{2 \cdot N} (2i + 1) \right] f(i)$$

Ενώ, ο αντίστοιχος αντίστροφος 1D μετασχηματισμός DCT είναι απλός, της μορφής  $F^{-1}(u)$ , όπου ισχύει το εξής:

$$\Lambda(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

### 3.11.2. Δισδιάστατος (2Δ) διακριτός μετασχηματισμός συνημιτόνου (2D-DCT: Discrete Cosine Transform).

Η γενική εξίσωση για ένα 2D ( N by M image) DCT ορίζεται από την ακόλουθη εξίσωση:

$$F(u, v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Lambda(i) \cdot \Lambda(j) \cdot \cos \left[ \frac{\pi \cdot u}{2 \cdot N} (2i + 1) \right] \cos \left[ \frac{\pi \cdot v}{2 \cdot M} (2j + 1) \right] \cdot f(i, j)$$

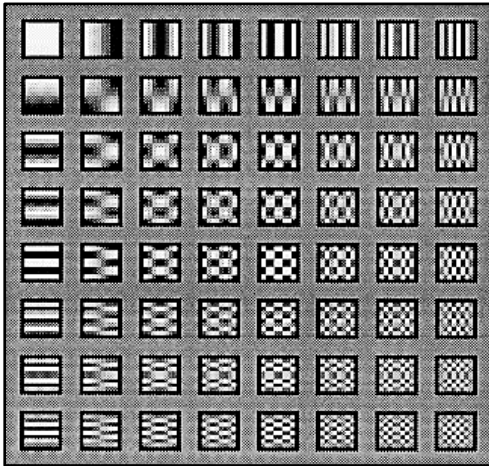
Και ο αντίστοιχος αντίστροφος μετασχηματισμός 2D DCT είναι απλός  $F^{-1}(u, v)$ , ομοίως περιγράφεται από την παρακάτω σχέση:

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

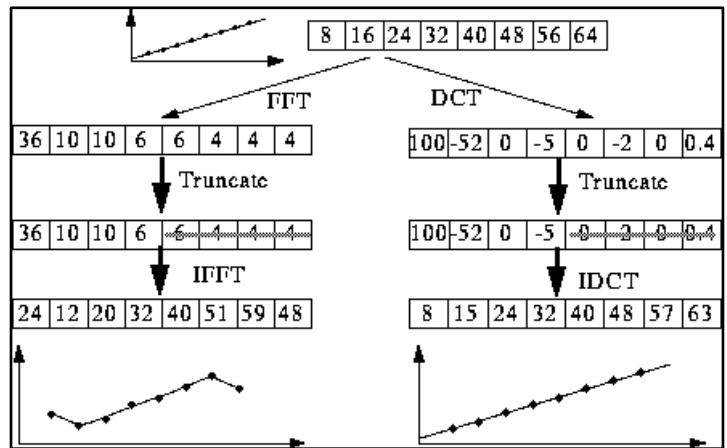
### Σχόλια – Παρατηρήσεις.

- ❖ Η εικόνα εισόδου είναι N με M.
- ❖  $f(i, j)$  είναι η ένταση του εικονοστοιχείου στη σειρά i και στη στήλη j.
- ❖  $F(u, v)$  είναι ο συντελεστής DCT στη σειρά k1 και η στήλη k2 της μήτρας DCT.
- ❖ Για τις περισσότερες εικόνες, μεγάλο μέρος της ενέργειας του σήματος βρίσκεται σε χαμηλές συχνότητες. αυτά εμφανίζονται στην επάνω αριστερή γωνία του DCT.
- ❖ Η συμπίεση επιτυγχάνεται καθώς οι κάτω δεξιά τιμές αντιπροσωπεύουν υψηλότερες συχνότητες και είναι συχνά μικρές - αρκετά μικρές ώστε να παραμεληθούν με μικρή ορατή παραμόρφωση.
- ❖ Η είσοδος DCT είναι ένας αριθμός 8 με 8 ακέραιων αριθμών. Αυτός ο πίνακας περιέχει το επίπεδο κλίμακας γκρι χρώματος κάθε εικονοστοιχείου.
- ❖ Τα εικονοστοιχεία 8 bit έχουν επίπεδα από 0 έως 255.
- ❖ Η σειρά εξόδων των συντελεστών DCT περιέχει ακέραιους αριθμούς, αυτά μπορεί να κυμαίνονται από -1024 έως 1023.

- ❖ Είναι υπολογιστικά ευκολότερο να εφαρμοστεί και πιο αποδοτικό να θεωρηθεί το DCT ως ένα σύνολο λειτουργιών βάσης το οποίο με δεδομένο ένα γνωστό μέγεθος συστοιχίας μεγέθους (8 x 8) μπορεί να προ-υπολογιστεί και να αποθηκευτεί. Αυτό περιλαμβάνει απλώς υπολογισμό τιμών για μια μάσκα συνέλιξης (παράθυρο 8 x8) που εφαρμόζεται (τιμές  $\sin m \times \pi \times \text{rixel}$  το παράθυρο αλληλεπικαλύπτεται με εικόνα εφαρμόστε παράθυρο accros όλες τις γραμμές / στήλες της εικόνας). Οι τιμές υπολογίζονται απλώς από τον τύπο DCT.
- ❖ Το DCT είναι παρόμοιο με το γρήγορο μετασχηματισμό Fourier (FFT), αλλά μπορεί να προσεγγίσει καλές γραμμές με λιγότερους συντελεστές.



Εικόνα 18 : Βασικές λειτουργίες DCT 64(8x8).



Εικόνα 19 : Σύγκριση DCT και FFT.

**Κεφάλαιο 4<sup>ο</sup>**  
**Το Πρότυπο**  
**Συμπίεσης MPEG.**

**4.1. Εισαγωγή.**

Τα αρχικά MPEG προέρχονται από τις λέξεις «moving picture expert group». Πρόκειται για μια επιτροπή ειδικών που δραστηριοποιούνται στα πλαίσια του οργανισμού ISO με στόχο να αναπτύξει απωλεστικούς συμπιεστές (codes) για συμπίεση πληροφορίας video. Μέχρι σήμερα η επιτροπή MPEG έχει διατύπωση πολλούς συμπιεστές που φέρουν την ονομασία MPEG και συνδέονται από ένα χαρακτηριστικό αριθμό. Έτσι με την ονομασία MPEG αναφερόμαστε ουσιαστικά σε μια οικογένεια ισχυρών συμπιεστών οπτικοακουστικής πληροφορίας (video). Οι πιο γνωστοί από αυτούς είναι ο MPEG-1, MPEG-2, MPEG-4. Η γενική ιδέα πίσω από την κωδικοποίηση του video χρησιμοποιώντας MPEG είναι η καταγραφή των μεταβολών/διαφορών μεταξύ των frame του video, διορθώνοντας σε τακτά διαστήματα την μετάδοση με πλήρεις, αναλυτικές εικόνες. Ο στόχος είναι η ανάπτυξη μιας μεθόδου για αποθήκευση και μετάδοση μέσω δικτύου ψηφιακού video και audio. Το MPEG standard σήμερα, αποτελείται από έξι φάσεις, MPEG-1, MPEG-2, MPEG-3, MPEG-4, MPEG-7 και MPEG-21 οι οποίες ορίζονται από τις ανάλογες συστάσεις της ISO/IEC. Ένα video αποτελείται από μια ακολουθία κινούμενων εικόνων, όπου οι διαδοχικές εικόνες που μεταδίδονται με σταθερό ρυθμό (συνήθως 25 εικόνες ανά sec) παρουσιάζουν χωρική και χρονική συσχέτιση μεταξύ τους, γεγονός που εκμεταλλεύεται το MPEG για να συμπίεσει την πληροφορία. Τα frames του video ομαδοποιούνται ανά ομάδες GOP (Group of Pictures), τα οποία κωδικοποιούνται και αποκωδικοποιούνται ανεξάρτητα. Υπάρχουν τρία είδη κωδικοποιημένων πλαισίων που θα περιγράψουν παρακάτω.

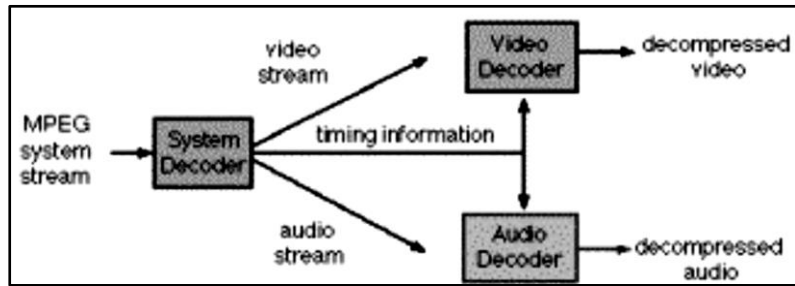
**4.2. Ανατομία σήματος MPEG.**

Ένα σήμα κωδικοποιημένο κατά MPEG αποτελείται από τρία επίπεδα:

- ❖ **Το επίπεδο συστήματος (system layer):** Εδώ, περιλαμβάνονται πληροφορίες σχετικά με το συγχρονισμό, πραγματοποιείται έλεγχος σχετικά με τη ροή του σήματος, ώστε να μην παρατηρείται έλλειψη ή πλεονασμός δεδομένων, παρέχονται πληροφορίες για σημεία αναφοράς που διευκολύνουν την τυχαία προσπέλαση και τέλος περιέχονται πληροφορίες για το διαχωρισμό του video από το audio και για την συγχρονισμένη απεικόνισή τους.
- ❖ **Το τηλεοπτικό επίπεδο (video layer).**
- ❖ **Το ακουστικό επίπεδο (audio layer).**

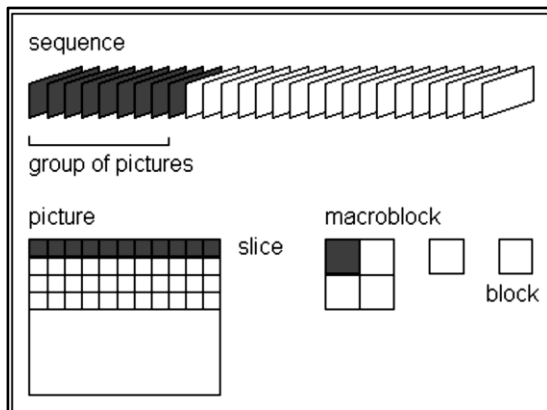
Τα επίπεδα τηλεοπτικό και ακουστικό, περιέχουν κωδικοποιημένη την εικόνα και τον ήχο αντίστοιχα. Η κωδικοποίηση αυτών των επιπέδων μπορεί να έχει γίνει ταυτόχρονα ή ξεχωριστά. Σε κάθε περίπτωση, τα δεδομένα των τριών επιπέδων συνενώνονται σε ένα ενιαίο σήμα (bit-stream) μέσω μιας διαδικασίας που ονομάζεται πολυπλεξία (multiplexing ή muxing). Η αντίστροφη διαδικασία ονομάζεται demultiplexing (ή demuxing). Μερικά συστήματα κάνουν την πολυπλεξία σε πραγματικό χρόνο (real-time) και άλλα όχι. Η διαδικασία της κωδικοποίησης και αποκωδικοποίησης του MPEG γίνεται με hardware ή με software. Η λύση του software είναι πιο φθηνή αλλά έχει μειονέκτημα στην ποιότητα της εικόνας και απαιτεί αρκετά ισχυρούς υπολογιστές για να λειτουργήσει. Η λύση του hardware είναι ακριβότερη, βασίζεται σε υλοποιήσεις των διαφόρων μεθόδων με ολοκληρωμένα VLSI και παράγει αυτόνομα συστήματα (κωδικοποιητές ή αποκωδικοποιητές) που δεν χρειάζονται υπολογιστή για να λειτουργήσουν, ενώ ταυτόχρονα έχουν πολύ καλή ποιότητα εικόνας. Είναι σημαντικό να σημειώσουμε ότι οι περισσότερες μέθοδοι κωδικοποίησης video όπως Cinepak, Indeo, Motion-JPEG, στηρίζονται στις μεθόδους συμπίεσης ακίνητης εικόνας JPEG αντιμετωπίζοντας την κινούμενη εικόνα σαν μια σειρά από διαδοχικές ακίνητες εικόνες. Η μέθοδος MPEG αν και δανείζεται σε μεγάλο βαθμό τις βασικές αρχές του JPEG επεκτείνει την προσέγγιση αυτή και περιγράφει την παράμετρο της κίνησης με ένα πιο λεπτομερή και αποτελεσματικό τρόπο από τις άλλες μεθόδους, με ευεργετικά αποτελέσματα από πλευράς ποσοστού συμπίεσης.



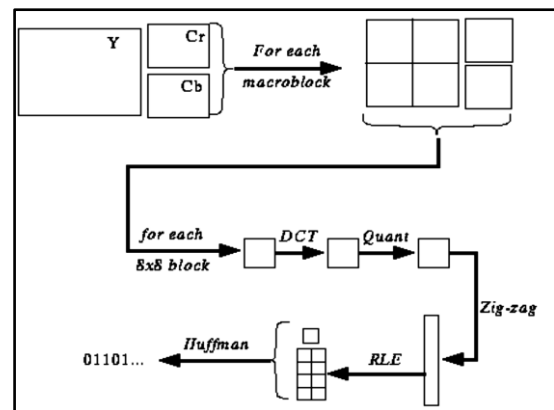


**Εικόνα 20: Αναπαράσταση λειτουργίας του MPEG.**

Προκειμένου να γίνει η επεξεργασία της εικόνας και η διαδικασία πρόβλεψης της κίνησης, το κάθε frame της εικόνας χωρίζεται σε τμήματα που ονομάζονται slices. Τα slices περιέχουν μία σειρά macroblocks, τα οποία όπως φανερώνει και το όνομά τους περιέχουν μία σειρά από blocks. Τα blocks είναι τμήματα της εικόνας διαστάσεων 8x8. Ο ρόλος των slices είναι να περιορίσουν τη διάδοση των λαθών. Αν ένα slice ληφθεί με λάθος παραλείπεται χωρίς να χάνουμε όλη την εικόνα. Γενικά τα slices βοηθούν στην απόκρυψη των σφαλμάτων (error concealment). Το κάθε macroblock περιέχει ένα πίνακα με πληροφορίες φωτεινότητας (16x16) και δύο πίνακες του μισού μεγέθους (8x8) για το χρώμα. Η ακολουθία αυτή είναι δομημένη ιεραρχικά:



**Εικόνα 21: Ανατομία του MPEG.**



**Εικόνα 22: Ενδο-πλαισιακός Τρόπος Συμπίεσης MPEG.**

#### **4.3. Βασικοί Τρόποι Συμπίεσης MPEG.**

Στην κωδικοποίηση της κινούμενης εικόνας απαντώνται δύο κυρίαρχοι τρόποι συμπίεσης. Η πρώτη είναι βασισμένη στη μέθοδο συμπίεσης ακίνητης εικόνας JPEG, ενώ η δεύτερη στην τμηματική πρόβλεψη κίνησης (block motion compensation). Πιο συγκεκριμένα:

- ❖ **Ενδοπλαισιακά (intra-frame):** Σύμφωνα με αυτή την προσέγγιση η κάθε εικόνα (frame) αντιμετωπίζεται σαν αυτόνομη μονάδα και κωδικοποιείται ανεξάρτητα από τις υπόλοιπες, οπότε το τελικό σήμα είναι μία σειρά από διακριτές ακίνητες εικόνες δηλαδή με τεχνικές συμπίεσης που εφαρμόζονται στην πληροφορία μόνο ενός frame. Το MPEG εφαρμόζει τον μετασχηματισμό DCT στην πληροφορία ενός frame και μέσω του κβαντισμού των συντελεστών DCT συμπιέζει την πληροφορία αυτήν. Τα frame που εφαρμόζονται κατά αυτόν τον τρόπο χαρακτηρίζονται ως I-frames και αποτελούν key-frames στην αλληλουχία των συμπιεστών πλαισίων που δημιουργεί το MPEG.
- ❖ **Διαπλαισιακά (inter-frame):** Λαμβάνονται κατά την κωδικοποίηση υπ'όψη οι πιθανές ομοιότητες μεταξύ των πλαισίων (frame) και κωδικοποιείται η διαφορά τους με χρήση της τμηματικής πρόβλεψης κίνησης (block motion compensation).

Έτσι στο τελικό σήμα υπάρχει μία εξάρτηση μεταξύ των πλαισίων, αφού για να αποκωδικοποιηθεί κάποιο πιθανόν θα πρέπει να ληφθούν πληροφορίες και από κάποιο άλλο είτε προηγούμενο είτε επόμενο. Γίνεται επομένως, πολύ μεγαλύτερη συμπίεση, αφού μόνο οι διαφορές μεταξύ των πλαισίων κωδικοποιούνται, δηλαδή με τεχνικές συμπίεσης που εκμεταλλεύονται τον τρόπο που μεταβάλλεται η πληροφορία μεταξύ διαδοχικών πλαισίων. Μια τέτοια τεχνική είναι η εκτίμηση κίνησης (motion estimation), στην οποία δύο διαδοχικά πλαίσια συνήθως δεν διαφέρουν πολύ. Έτσι η κωδικοποίηση κάθε επόμενου πλαισίου μπορεί να βασιστεί στην πληροφορία του προηγούμενου και δεν χρειάζεται να αποθηκευτεί ολόκληρη η πληροφορία του επόμενου πλαισίου αλλά μόνο το τμήμα που διαφέρει από το προηγούμενο πλαίσιο, κάτι που σίγουρα εμφανίζεται διαφοροποιημένο λόγο κάποιας κίνησης. Τέλος, αξίζει να αναφερθεί πως για να εφαρμόσει το MPEG τη διαπλασιακή συμπίεση δημιουργεί δύο νέα είδη πλαισίων (frame), το πλαίσιο P (P-Frames, predicative frames) και το πλαίσιο B (B-Frames, bi-directional frames).

#### **4.4. Είδη πλαισίων στο MPEG.**

Είναι κοινώς αποδεκτό πως, ο όρος frame rate αναφέρεται στο ρυθμό ανανέωσης των αλληπάλγηλων εικόνων μίας οθόνης ανά δευτερόλεπτο. Αυτές οι εικόνες ονομάζονται frames (καρέ) και μετριοούνται ανά δευτερόλεπτο (frames per second ή fps). Όσο πιο μεγάλος είναι αυτός ο αριθμός, τόσο πιο ομαλή είναι η κίνηση. Το ανθρώπινο μάτι αντιλαμβάνεται "σπάσιμο" στην κίνηση ενός μοντέλου, όταν ο ρυθμός ανανέωσης των καρέ της οθόνης ανά δευτερόλεπτο βρίσκεται κάτω των 30. Ένα Frame περιέχει όλη την απαραίτητη πληροφορία για το χρώμα και την φωτεινότητα, έτσι ώστε να εμφανιστεί η εικόνα. Αυτή η πληροφορία οργανώνεται σε τρεις πίνακες που περιέχουν τις τιμές για την φωτεινότητα (luminance) και χρώμα (chrominance). Το μέγεθος αυτών των πινάκων διαφέρει ανάλογα με τον ρυθμό δειγματοληψίας χρώματος που εφαρμόζεται. Ειδικότερα, οφείλουμε να τονίσουμε πως για την κωδικοποίηση της εικόνας δεν χρησιμοποιούνται τρία ανεξάρτητα σήματα για τις συνιστώσες R,G,B παρόλο που κάθε χρώμα μπορεί να περιγραφεί από αυτές και άλλο ένα για τη φωτεινότητα. Ωστόσο, στην πράξη χρησιμοποιείται μόνο αυτό της φωτεινότητας (Y) και δύο άλλα (Cr) και (Cb) που περιέχουν τις πληροφορίες για τα R,G,B και αν συνδυαστούν με τη φωτεινότητα, προκύπτουν οι παρακάτω σχέσεις:

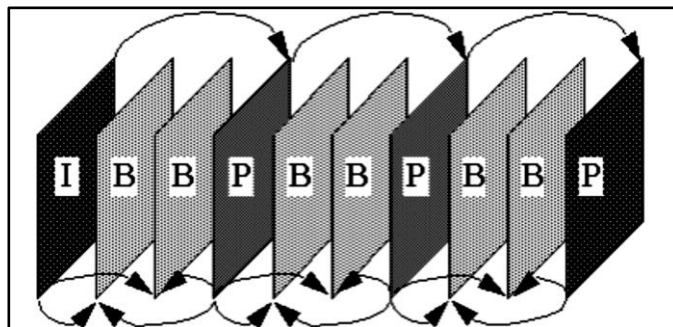
- ❖  $Y = 0,30R + 0,59G + 0,14B$   
 $Cr = 0,74(R-Y) - 0,27(B-Y) = 0,60R + 0,28G + 0,32B$   
 $Cb = 0,48(R-Y) + 0,41(B-Y) = 0,21R + 0,52G + 0,31B$  (για το NTSC)
- ❖  $Y = 0,30R + 0,59G + 0,11B$   
 $Cr = 0,493(B-Y) = -0,15R - 0,29G + 0,44B$   
 $Cb = 0,877(R-Y) = 0,62R - 0,52G - 0,10B$  (για το PAL)

Πράγματι, η ανθρώπινη όραση είναι λιγότερο ευαίσθητη στις χρωματικές μεταβολές παρά σε εκείνες της φωτεινότητας, οπότε συμπεραίνουμε πως χρειάζονται λιγότερα δεδομένα χρωματικής πληροφορίας για να δημιουργηθεί η αίσθηση της ποιοτική εικόνας στον θεατή. Έτσι, κατά τη διαδικασία της δειγματοληψίας μπορούν να ψηφιοποιηθούν λιγότερα δείγματα από τις συνιστώσες χρώματος (Cb και Cr) σε σχέση με το σήμα φωτεινότητας (luma σήμα Y), γεγονός που αποτελεί σημαντικό παράγοντα για την μείωση όγκου. Η αναλογία δειγμάτων που παράγονται μεταξύ των σημάτων εκφράζεται ως ένας διπλός λόγος ακέραιων αριθμών όπου ο πρώτος είναι πάντοτε το 4. Αν δεν υπάρξει δειγματοληψίας ο λόγος αυτός είναι 4:4:4. Οι πιο συνηθισμένες αναλογίες είναι 4:2:2 και 4:2:0. Κατά την αποκωδικοποίηση η πληροφορία χρώματος που λείπει αναπαράγεται με τον μέσο όρο των γειτονικών pixel. Συνοψίζοντας, σε ένα σήμα MPEG υπάρχουν τριών ειδών πλαίσια (frame):

- ❖ **I-frames (Intra frames):** Τα I-frames είναι εικόνες που δεν χρησιμοποιούν εκτίμηση κίνησης, αλλά μεταδίδουν συνολικά την πληροφορία της εικόνας σε μορφή κωδικοποιημένη σύμφωνα με το πρότυπο JPEG, και ακολουθείται η ακόλουθη διαδικασία.

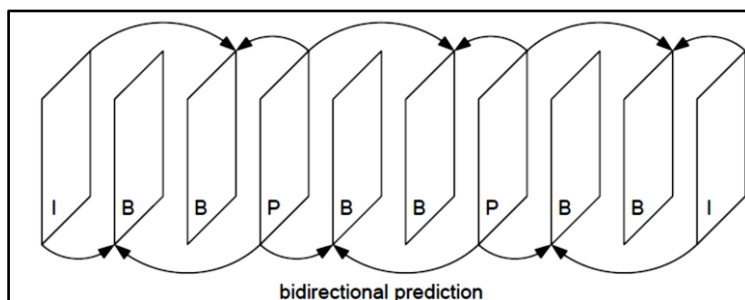
Η εικόνα χωρίζεται σε υπο-εικόνες (blocks) των 8x8 ή 16x16 pixels και εφαρμόζεται ο διακριτός μετασχηματισμός συνημιτόνου (Discrete Cosine Transformation - DCT) σε καθεμιά από αυτές. Στην μετασχηματισμένη πλέον εικόνα, στο DCT πεδίο, εφαρμόζεται η διαδικασία του κβαντισμού και της συμπίεσης-κωδικοποίησης (Huffman), οπότε τελικά κατασκευάζεται η εικόνα σε συμπιεσμένη μορφή, αλλά με όλη την πληροφορία και γι' αυτό χρησιμοποιείται και ως πλαίσιο αναφοράς για τα υπόλοιπα είδη πλαισίων. Άρα τα πλαίσια τύπου I είναι τα μόνα που είναι κωδικοποιημένα στο σύνολό τους και η αποκωδικοποίηση μπορεί να γίνει χωρίς αναφορά σε κάποιο άλλο. Εν ολίγοις η εικόνα χωρίζεται σε macroblocks και για κάθε block ξεχωριστά εφαρμόζεται DCT, Κβαντοποίηση, Zig-Zag Scanning, Run-Length-Encoding και Huffman Encoding .

- ❖ **P-frames (Predicted frames):** Στα P-frames χρησιμοποιείται η διαδικασία της εκτίμησης και αντιστάθμισης κίνησης (motion estimation and compensation), δηλαδή μεταδίδεται η πληροφορία της κίνησης των blocks από το προηγούμενο πλαίσιο αναφοράς (το προηγούμενο I ή P frame) προς το τρέχον πλαίσιο P, καθώς και το σφάλμα που προκύπτει από αυτή τη διαδικασία. Με τη σειρά τους μπορούν να αποτελέσουν και αυτά σημείο αναφοράς για επόμενα πλαίσια και αυτός είναι και ο λόγος που συμβάλλουν στην εισαγωγή και διάδοση σφαλμάτων, καθώς η διαδικασία της πρόβλεψης κίνησης δεν μπορεί να είναι 100% ακριβής. Συνάμα, η πληροφορία της κίνησης αναπαρίσταται μέσω των διανυσμάτων κίνησης, ενώ το σφάλμα μεταδίδεται με τη μορφή κωδικοποιημένης με τον DCT μετασχηματισμό εικόνας. Υπάρχει σημαντική ελευθερία στον τρόπο εξαγωγής της πληροφορίας κίνησης, γεγονός που επιτρέπει συνεχώς τη βελτίωση της ποιότητας της κωδικοποίησης και της εκμετάλλευσης των ιδιοτήτων του προτύπου. Δεν έχουν το μέγεθος των I πλαισίων γιατί δεν έχουν περιγραφεί με την ίδια ακρίβεια, δηλαδή παρουσιάζουν μεγαλύτερο ποσοστό συμπίεσης.



**Εικόνα 23: Προς τα εμπρός πρόβλεψη στα P-frames του MPEG.**

Η διαδικασία λοιπόν της κωδικοποίησης τους, είναι: Σύγκριση macroblocks και δημιουργία ενός γραμμικού συνδυασμού αυτών που παρουσιάζουν σημαντική ομοιότητα, δημιουργία motion vector, μετασχηματισμός DCT σε κάθε block του νέου macroblock, Κβαντοποίηση, Run-Length-Encoding και το αποτέλεσμα κωδικοποιείται με κωδικοποίηση Huffman, όπως και στα I πλαίσια.



**Εικόνα 24: Πρόβλεψη Διπλής Κατεύθυνσης στα B-frames του MPEG.**

- ❖ **B-frames (Bidirectional Interpolated frames):** Στα B-frames υλοποιείται και πάλι η διαδικασία εκτίμησης και αντιστάθμισης κίνησης, με τη διαφορά πως εδώ, χρησιμοποιείται το προηγούμενο πλαίσιο αναφοράς (το προηγούμενο I ή P frame) ή το επόμενο πλαίσιο αναφοράς (το επόμενο I ή P frame) ή ακόμη και τα δύο ταυτόχρονα, ενώ το frame συντίθεται μέσω διαδικασίας χώρο-χρονικά ισοσταθμισμένης γραμμικής παρεμβολής. Δε συντελούν τόσο πολύ στη διάδοση των σφαλμάτων, γιατί δεν χρησιμοποιούνται ως σημεία αναφοράς και επιπλέον μειώνουν σημαντικά το σφάλμα παίρνοντας το μέσο όρο από δύο πλαίσια. Μπορούμε να πούμε ότι η χρονική διάρκεια ζωής τους περιορίζεται μόνο σε αυτά και δεν επεκτείνεται με το να κληροδοτούν πληροφορίες σε άλλα πλαίσια, κάτι που πολλές φορές σε συνδυασμό και με την υπολογιστική πολυπλοκότητα που απαιτούν για την κωδικοποίηση και αποκωδικοποίηση τα κάνει μη επιθυμητά από τους κατασκευαστές. Η πληροφορία της κίνησης αναπαρίσταται μέσω διανυσμάτων κίνησης, που εδώ είναι δύο ειδών: διανύσματα προς τα εμπρός κίνησης (forward motion vectors) με αναφορά στο προηγούμενο πλαίσιο αναφοράς (forward prediction), και διανύσματα προς τα πίσω κίνησης (backward motion vectors) με αναφορά στο επόμενο πλαίσιο αναφοράς (backward prediction). Ο κύριος λόγος ύπαρξης των B-πλαίσια είναι η κάλυψη της περίπτωσης πως ορισμένες πληροφορίες της εικόνας να υπάρχουν σε επόμενα πλαίσια και να παραλείπονται από τα προηγούμενα. Συνεπώς, η πρόβλεψή τους με τα P πλαίσια θα ήταν αδύνατη, διότι κατασκευάζονται με βάση τα I και τα B με βάση τα I και P. Άρα, είναι προφανές ότι τα I πρέπει να έχουν σταλεί πριν τα αντίστοιχα P. Επίσης και τα P αλλά και τα I πρέπει να έχουν σταλεί πριν από τα αντίστοιχα B, παρόλο που στη μετάδοση αυτά παρεμβάλλονται ανάμεσά τους. Η μικρότερη μονάδα που μπορεί να αποκωδικοποιηθεί ανεξάρτητα ονομάζεται GOP (Group of Pictures) και περιέχει όλα τα I,P,B πλαίσια που χρειάζονται για την αποκωδικοποίηση, χωρίς να γίνονται αναφορές σε άλλο GOP.

#### **4.5.Ομάδα πλαισίων (Group of Pictures).**

Ως ομάδα πλαισίων, ορίζεται το σύνολο των πλαισίων (frames) που βρίσκονται ανάμεσα στο key-frame (τα καρτέ από τα οποία εξαρτώνται τα B frames) του τρέχοντος GOP, και του key-frame του προηγούμενου GOP. Ένα πλαίσιο I μαζί με τα πλαίσια P και B που το ακολουθούν αποτελεί μια ομάδα πλαισίων ( Group of picture ). Το μέγεθος μιας τέτοιας ομάδας μπορεί να ρυθμιστεί σε 8, 12 ή 16 πλαίσια ανάλογα με το πιο μέγεθος οδηγεί σε καλύτερη κωδικοποίηση. Κάθε αρχείο video mpeg αποτελείται από μια τέτοια ακολουθία (GOPs) που με την σειρά αποτελούνται από πλαίσια. Κάθε πλαίσιο αναλύεται σε macroblocks και κάθε macroblock μπορεί να αποτελείται από 6 απλά block (4 για το σήμα luma και 3 για το chroma υπό-δειγματοληψία 4:2:0) ή και λιγότερα. Η οργάνωση με βάση τις ομάδες (GOPs) επιτρέπει την τυχαία προσπέλαση στην πληροφορία του αρχείου καθώς μια ομάδα αποτελεί την ανεξάρτητα αποκωδικοποιημένη μονάδα πληροφορίας μέσα στο αρχείο.

#### ❖ **Χαρακτηριστικά Ομάδας Πλαισίων:**

- ❖ Η μικρότερη μονάδα που αποκωδικοποιείται ανεξάρτητα.
- ❖ Περιέχει όλα τα I,P,B πλαίσια που χρειάζονται για αποκωδικοποίηση (χωρίς αναφορά σε άλλα GOP)

#### **4.5.1 Κομμάτια (Slices).**

Ένα MPEG Frame χωρίζεται σε κομμάτια (slices). Ένα slice αποτελείται από μια σειρά από macroblocks ακολουθώντας μια σάρωση του frame με κατεύθυνση από αριστερά προς τα δεξιά και από πάνω προς τα κάτω. Στην κωδικοποιημένη ροή των bits κάθε slice ξεκινάει με μια επικεφαλίδα η οποία δεν έχει κωδικοποιηθεί όπως γίνεται σε όλα τα δεδομένα που υπάρχουν στην ροή μετάδοσης bits και συνεπώς, είναι το χαμηλότερο επίπεδο ενοτήτων στη

κωδικοποιημένη ροή που μπορεί να έχει κάποιος πρόσβαση, χωρίς να χρειαστεί αποκωδικοποίηση. Η παρουσία τμημάτων δεν συμβάλλει στη συμπίεση της εικόνας, παρά μόνο στην ύπαρξη πληροφορίας συγχρονισμού μέσα σε μία εικόνα σε ότι αφορά την απεικόνισή της. Σε περίπτωση ενός λάθους κατά τη μετάδοση και την απώλεια πληροφορίας σχετικά με την εικόνα, η διεργασία δεν απορρίπτει ολόκληρο το frame, παρά χρησιμοποιεί την πληροφορία στο τμήμα και συνεχίζει με την αποκωδικοποίηση του επόμενου block. Συμπεραίνουμε, ότι έχοντας πολλά blocks σε μία εικόνα, έχουμε μεγάλη ανεκτικότητα σε λάθη. Παράλληλα όμως, αυτό συνεπάγεται ένα μεγάλο overhead για κάθε εικόνα, οπότε πρέπει να βρούμε την χρυσή τομή ανάμεσα σε ποιότητα, παρουσία λαθών και απαιτούμενο εύρος ζώνης (bandwidth).

#### 4.5.2. Τμήματα-Μακροτμήματα Blocks – Macroblocks.

Τα blocks και τα macroblocks συνθέτουν μία ιεραρχία στη δομή του MPEG: κάθε frame διαιρείται σε έναν αριθμό blocks, τα οποία ομαδοποιούνται σε macroblocks και αυτά με την σειρά τους σε slices. Κάθε block αποτελείται από οχτώ γραμμές και κάθε γραμμή περιέχει οχτώ δείγματα τιμών luminance και chrominance από ένα εικονο-πλαίσιο (pixel). Έτσι συνολικά, έχουμε 64 τιμές chrominance ή luminance σε ένα block, κάθε block συνεπώς αντιπροσωπεύει 64 εικονοστοιχεία (pixels). Κάθε macroblock περιέχει σταθερά τέσσερα blocks με τιμές luminance και έναν αριθμό από blocks με τιμές chrominance. Ο αριθμός των blocks chrominance σε ένα macroblock εξαρτάται από το ρυθμό δειγματοληψίας χρώματος. Έτσι, σύμφωνα με τον παρακάτω πίνακα, συναντώνται:

<b>Blocks - Macroblocks</b>	
<b>Ρυθμός</b>	<b>Περιεχόμενο</b>
Macroblocks ρυθμού 4:2:0	Δύο blocks με πληροφορία chrominance.
Macroblocks ρυθμού 4:2:2	Τέσσερα blocks με πληροφορία chrominance.
Macroblocks ρυθμού 4:4:4	Οχτώ blocks με πληροφορία chrominance

**Πίνακας 1 : Τμήματα-Μακροτμήματα.**

#### 4.6. Διαδικασία Συμπίεσης κατά το MPEG.

- ❖ Αναλύεται το αρχείο video για να καθορίσει το ποια πλαίσια θα κωδικοποιηθούν ως πλαίσια I, P και B. Επίσης καθορίζεται και το μέγεθος των ομάδων πλαισίων (GOPs) και ο μέγιστος και ελάχιστος ρυθμός δεδομένων (bit rate).
- ❖ Τα πλαίσια διαιρούνται σε macro-blocks (16X16pixels).
- ❖ Το σήμα μετατρέπεται από RGB σε YUV ενώ ταυτόχρονα γίνεται και υπο-δειγματοληψία χρώματος. Το MPEG εφαρμόζει το σχήμα 4:2:0. Έτσι από κάθε macro-block 16X16 προκύπτουν τελικά τέσσερα 8X8 block με πληροφορία φωτεινότητας (Y) και δύο με πληροφορία χρώματος (ένα U και ένα V).
- ❖ Εφαρμόζει την τεχνική αντιστάθμιση κίνησης σε κάθε πλαίσιο P και B. Δηλαδή, υπολογίζει τα διανύσματα κίνησης και τούς συντελεστές πιθανού σφάλματος για κάθε macro-block.
- ❖ Σε κάθε πλαίσιο I, P και B τα macro-blocks συμπιέζονται κατά JPEG δηλαδή:
  - ❖ Εφαρμογή μετασχηματισμού DCT.
  - ❖ Κβαντισμός ασυντέλεστων DCT.
  - ❖ Συμπίεση RLE των συντελεστών DCT κάθε macro-block.
- ❖ Στην μορφή πληροφορίας που προκύπτει εφαρμόζεται και ο αλγόριθμος συμπίεσης Huffman.

Από το παραπάνω παρατηρούμε πως, οι κυρίαρχες φάσεις συμπίεσης στην κωδικοποίηση κατά MPEG είναι:

---

Υποδειγματοληψία χρώματος (chroma sub sampling).

---

Κβάντωση των συντελεστών DCT.

---

Αντιστάθμιση κίνησης.

Επίσης ο συμπιεστής MPEG είναι ασύμμετρος συμπιεστής, δηλαδή η διαδικασία κωδικοποίησης απαιτεί περισσότερο χρόνο από αυτήν της αποκωδικοποίησης.

#### **4.6.1. Το πρότυπο κωδικοποίησης MPEG – 1.**

Το αποτέλεσμα των εργασιών των πρώτων συνεδριάσεων της επιτροπής MPEG (1988), ήταν η δημιουργία του πρώτου προτύπου MPEG-1 (ISO/IEC 11172 '92). Με το MPEG-1 παράγεται μια ανάλυση εικόνας 352x240 pixels με τα καρέ να έχουν μια ταχύτητα της τάξεως των 30 καρέ ανά δευτερόλεπτο (30fps). Το πρότυπο αυτό αφορά εγγραφή και αναπαραγωγή κινητού video, που χρησιμοποιείται σε εφαρμογές ατομικών υπολογιστών. Η ποιότητα της εικόνας video που παράγει, είναι λίγο κατώτερη των συμβατικών VCR (video). Οι εφαρμογές αυτού του προτύπου περιορίζονται σε μορφές οπτικού σήματος, που δεν έχει αλληλοδιαδοχή των πεδίων και υποστηρίζει κωδικοποιήσεις οπτικών σημάτων μέχρι 1,5 Mbit/s. Η πιο δημοφιλής χρήση του MPEG-1 είναι στα δισκάκια VCD και όπου αλλού χρειάζεται μικρό σε σχέση με το MPEG-2 bandwidth.

#### **4.6.2. Το πρότυπο κωδικοποίησης MPEG-2.**

Το πρωτόκολλο MPEG – 2 παρουσιάστηκε ως επίσημο πρωτόκολλο αρχικά το Νοέμβριο του 1994 στη Σιγκαπούρη. Η πρώτη αυτή έκδοση περιλάμβανε μόνο τα τρία πρώτα κεφάλαια του πρωτοκόλλου, ενώ τα υπόλοιπα οκτώ παρουσιάστηκαν ένα χρόνο αργότερα. Η ανάπτυξη αυτού του νέου πρωτοκόλλου είχε ως πρωταρχικό στόχο την έλλειψη κάποιων δυνατοτήτων από το MPEG-1, οι οποίες άρχισαν να ζητούνται από τη βιομηχανία ψηφιακής εικόνας. Το βασικότερο πρόβλημα που ανέκυπτε στο MPEG – 1 ήταν ότι η ποιότητα του προτύπου VHS που απέδιδε ήταν χαμηλή σε σχέση με την ποιότητα των προτύπων της τηλεόρασης PAL και του HDTV που χρησιμοποιούνταν εκείνη την εποχή από τη τηλεόραση. Επίσης, οι μελέτες μέχρι τότε έδειχναν ότι ο αλγόριθμος που χρησιμοποιούσε το MPEG-1 ήταν αρκετά αναποτελεσματικός σε σχέση με τους αλγορίθμους που είχαν αναπτυχθεί εκείνη την εποχή για τη συμπίεση video, ενώ παράλληλα ο νέος αλγόριθμος συμπίεσης του MPEG-2 θα έπρεπε να είναι συμβατός με αυτόν του MPEG-1 ώστε να υπάρχει μια συνέχεια μεταξύ των δύο πρωτοκόλλων. Παρόμοιο πρόβλημα αντιμετώπιζε η ομάδα ανάπτυξης και με τον ήχο. Για να σιγουρευτεί μια αρμονική λύση, που θα κάλυπτε μια ευρεία γκάμα εφαρμογών, η ομάδα εργασίας ανέπτυξε το MPEG-2 σε στενή συνεργασία με τις ομάδες μελέτης ITU-TS Study Group και Expert Group for ATM Video Coding. Εκτός των δύο αυτών ομάδων, άλλοι αντιπρόσωποι των σχεδιαστών των ITU-TS, EBU, ITU-RS, SMPTE και North America HDTV εργάστηκαν επίσης στο MPEG-2. Με λίγα λόγια η επιτυχία του προτύπου MPEG-1 οδήγησε στη δημιουργία του επόμενου προτύπου MPEG-2 το οποίο προσπάθησε να φέρει μια πιο γενική περιγραφή των μεθόδων κωδικοποίησης εικόνας και ήχου, με περισσότερες επιλογές διαβάθμισης της ποιότητας, το οποίο θα στόχευε κυρίως σε εφαρμογές μεσαίας και υψηλής ποιότητας. Επομένως αναφερόμαστε, σε ένα πρότυπο κωδικοποίησης video και audio για καλύτερη ποιότητα μετάδοσης στις τηλεοράσεις, που παράλληλα αρχίζει και χρησιμοποιείται και στις περισσότερες ταινίες DVD.

#### **4.6.2.1 Οργάνωση προτύπου.**

Για να κατανοήσουμε την οργάνωση του αρχικά θα πρέπει να δεχτούμε ότι το βασικό στοιχείο της κωδικοποίησης του είναι η εικόνα, η οποία υφίσταται την ενδοπλαισιακή συμπίεση. Για να εφαρμοσθεί λοιπόν η συμπίεση, οι εικόνες οργανώνονται όπως ήδη αναφέραμε σε ομάδες εικόνων (GOP), συνήθως 12 στον αριθμό, όμως μόνο μία στις δώδεκα, που ονομάζεται και πλαίσιο I, (από το intra-frame), προέρχεται από όλα τα στοιχεία της εικόνας δεν χάνει δηλαδή καμία πληροφορία. Οι υπόλοιπες σχηματίζονται από την εκπομπή στοιχείων που περιέχουν τις διαφορές με τις προηγούμενες και ονομάζονται προβλεπόμενα πλαίσια (Predicted frames). Τα intra-frame τα οποία περιέχουν πολλές πληροφορίες εκπέμπονται με πολύ μικρή συχνότητα, ενώ τα υπόλοιπα πλαίσια που περιέχουν πιο λίγες εκπέμπονται με μεγαλύτερη συχνότητα. Με τον τρόπο αυτό επιτυγχάνεται μεγάλη συμπίεση (διαπλαισιακή). Η συμπίεση στο MPEG-2 επιτυγχάνεται με την ενδοπλαισιακή και την διαπλαισιακή συμπίεση. Ειδικότερα, στην ενδοπλαισιακή συμπίεση MPEG-2 η εικόνα διαιρείται και επεξεργάζεται σε δύο διαστάσεις με ομάδες των 8 pixel οριζοντίως και 8 pixel καθέτως. Στο σύστημα MPEG χρησιμοποιείται ο διακριτός μετασχηματισμός του συνημιτόνου (DCT), όπου μετασχηματίζονται και απεικονίζονται τα στοιχεία της ομάδας pixel από το διάστημα του χώρου σ' ένα διάστημα συχνότητας στο χώρο. Έτσι, δεν μειώνουμε τον αριθμό των δυφίων, που απαιτούνται για να παραστήσουμε την ομάδα. Η μείωση στον αριθμό των δυφίων προκύπτει γιατί οι ομάδες φυσικών και συνήθων εικόνων η κατανομή των ομοίων pixel δεν είναι ομοιόμορφη. Με το μετασχηματισμό η ενέργεια συγκεντρώνεται στους συντελεστές των χαμηλών συχνοτήτων, ενώ οι άλλοι συντελεστές έχουν σχεδόν μηδενική τιμή. Η μείωση στη ροή των δυφίων εξασφαλίζεται επίσης και με τη κβαντοποίηση και με τη μη εκπομπή των συντελεστών, που έχουν πολύ μικρή τιμή, όπως επίσης και με την κωδικοποίηση των υπολοίπων συντελεστών. Θα πρέπει να τονίσουμε ότι στο μετασχηματισμό του συνημιτόνου η έννοια της συχνότητας δεν αναφέρεται στο χρόνο αλλά στο μήκος. Η οριζόντια συχνότητα αυξάνει από αριστερά προς τα δεξιά και η κατακόρυφη συχνότητα από πάνω προς τα κάτω. Όταν έχουμε μεγάλο πλεονασμό στο χώρο, τότε οι περισσότεροι συντελεστές θα είναι μηδενικοί. Αν όλα τα στοιχεία της ομάδας είναι ομοιόμορφα, θα υπάρχει ένας μόνο συντελεστής στην επάνω αριστερή γωνία και θα παριστά τη συνεχή συνιστώσα του οπτικού σήματος. Μετά το μετασχηματισμό οι συντελεστές κβαντοποιούνται, ώστε να μειώσουμε τον αριθμό δυφίων που απαιτούνται για την εκπομπή. Για κάθε συντελεστή εφαρμόζεται διαφορετική κβαντοποίηση. Η κβαντοποίηση εισάγει σφάλμα και παραμορφώσεις στην εικόνα. Εν γένει είναι ανεκτό μεγαλύτερο σφάλμα κβαντοποίησης για τις υψηλές συχνότητες, όπως επίσης στα σήματα χρωματισμού. Μετά την κβαντοποίηση οι συντελεστές σαρώνονται σε σχήμα zig-zang, και αυτοί που παράγονται από την σάρωση, κωδικοποιούνται με μέτρηση των συντελεστών που έχουν μηδενική τιμή και προηγούνται κάποιου που δεν έχει μηδενική τιμή. Η κωδικοποίηση αυτή καλείται κωδικοποίηση μέτρησης μήκους (run-length code). Επειδή πολλές από τις τιμές των συντελεστών είναι μηδενικές, η χρήση του κώδικα αυτού μειώνει το μέγεθος του απαιτούμενου κώδικα για την περιγραφή των μηδενικών. Συνεπώς ο κώδικας μέτρησης μήκους αποτελείται από δύο αριθμούς. Ο πρώτος δίνει τον αριθμό των μηδενικών που και ο δεύτερος το μέγεθος του αμέσως επόμενου συντελεστή που δεν έχει μηδενική τιμή. Η κωδικοποίηση μέτρησης μήκους υφίσταται περαιτέρω κωδικοποίηση με έναν κώδικα που είναι γνωστός με το όνομα κώδικας μεταβλητού μήκους (variable length code). Ο συγκεκριμένος κώδικας καθορίζει άλλους που έχουν μεταβλητό μήκος, το οποίο εξαρτάται από τη συχνότητα εμφάνισης του κάθε συνδυασμού (αριθμού προηγούμενων μηδενικών και μεγέθους επομένου συντελεστή). Οι συχνά εμφανιζόμενοι συνδυασμοί έχουν μικρό μήκος ενώ οι σπανιότερα εμφανιζόμενοι μεγαλύτερο μήκος.

#### **4.6.2.2 Επίπεδα και προφίλ του προτύπου MPEG-2.**

Το πρότυπο MPEG-2 υποστηρίζει έναν αριθμό από προφίλ που στοχεύουν σε μια συγκεκριμένη περιοχή εφαρμογών και καθορίζουν το σύνολο των εργαλείων συμπίεσης που

χρησιμοποιούνται. Υποστηρίζονται τέσσερα προφίλ, εκ των οποίων το κύριο είναι για χρήση γενικού σκοπού, το απλό προφίλ είναι παρόμοιο με το κύριο, ενώ τα άλλα διαπραγματεύονται την επεκτασιμότητα (scalability) και την τηλεόραση υψηλής ευκρίνειας (HDTV). Τα προφίλ διαφέρουν όσον αφορά την παρουσία ή απουσία εικόνων Β, την ανάλυση της χρωματικότητας καθώς και την επεκτασιμότητα ενός κωδικοποιημένου συρμού προς τα άλλα format εικόνων. Επιπροσθέτως, το πρότυπο MPEG-2 υποστηρίζει τέσσερα επίπεδα (levels) που ορίζουν την ανάλυση της εικόνας. Το χαμηλό επίπεδο (352 x 288) αντιστοιχεί στην προς τα πίσω συμβατότητα με το πρότυπο συμπίεσης MPEG-1 και στους video εγγραφείς (VCR), το κύριο επίπεδο (720 x 576) αντιστοιχεί στην τυποποιημένη ανάλυση για τη μετάδοση NTSC, το υψηλό-1440 επίπεδο (1440 x 1152) αποτελεί την HDTV και το υψηλό επίπεδο (1920 x 1152) βελτιστοποιεί την HDTV. Τα προφίλ διαφέρουν όσον αφορά την παρουσία ή απουσία εικόνων Β, την ανάλυση της χρωματικότητας καθώς και την επεκτασιμότητα ενός κωδικοποιημένου συρμού προς τα άλλα format εικόνων. Επιπροσθέτως, το πρότυπο MPEG-2 υποστηρίζει τέσσερα επίπεδα (levels) που ορίζουν την ανάλυση της εικόνας. Το χαμηλό επίπεδο (352 x 288) αντιστοιχεί στην προς τα πίσω συμβατότητα με το πρότυπο συμπίεσης MPEG-1 και στους video εγγραφείς (VCR), το κύριο επίπεδο (720 x 576) αντιστοιχεί στην τυποποιημένη ανάλυση για τη μετάδοση NTSC, το υψηλό-1440 επίπεδο (1440 x 1152) αποτελεί την HDTV και το υψηλό επίπεδο (1920 x 1152) βελτιστοποιεί την HDTV.

#### **4.6.2.3 Κλιμακωσιμότητα στο MPEG-2.**

Το πρότυπο συμπίεσης MPEG-2 αποτελεί τη βάση των προτύπων που έχουν αναπτυχθεί για την μετάδοση ψηφιακής τηλεόρασης, σε παγκόσμιο επίπεδο. Στο παρακάτω γραφικό, παρατηρούμε τα είδη της κλιμακωσιμότητας, όπως παρουσιάζεται στο πρότυπο MPEG, αλλά και μία σύντομη περιγραφή της λειτουργίας τους.

##### **Χωρική Κλιμακωσιμότητα (Spatial Scalability)**

- Δύο επίπεδα video με διαφορετικές χωρικές αναλύσεις.

##### **Χρονική Κλιμακωσιμότητα (Temporal Scalability)**

- Δύο επίπεδα video με το ένα να κάνει χρονική πρόβλεψη ως προς το άλλο.

##### **Τεμαχισμός Δεδομένων (Data Partitioning)**

- Χωρισμός σε κρίσιμα δεδομένα, όπως επικεφαλίδες, διανύσματα κίνησης και λιγότερο κρίσιμα όπως DCT συντελεστές.

##### **Κλιμακωσιμότητα Εύρους (SNR Scalability)**

- Δύο επίπεδα video με ίδια χωρική ανάλυση, αλλά διαφορετική ποιότητα (προοδευτική ή κατά επίπεδα κωδικοποίηση)

#### **4.6.2.4. Ατέλειες του MPEG-2.**

Παρόλο που κάθε τύπος μιας σειράς προγράμματος αποτελείται από ένα μοναδικό σύνολο από παραμέτρους εικόνας, συγκεκριμένες γενικεύσεις μπορούν να γίνουν, όσον αφορά τις ατέλειες, που ενδεχομένως να παρουσιαστούν ως αποτέλεσμα του προτύπου συμπίεσης MPEG-2. Οι ατέλειες συμπεριλαμβάνουν τα ακόλουθα φαινόμενα:

- ❖ **Μπλοκ εφέ:** Αυτά μπορούν να γίνουν ορατά, όταν το μάτι εντοπίσει ένα αντικείμενο που κινείται γρήγορα κατά μήκος της οθόνης. Τα μπλοκ κομμάτια παραμένουν σταθερά, ακόμα και όταν το αντικείμενο περνάει κινούμενο από μέσα τους. Αυτό το φαινόμενο προκαλείται από κακή εκτίμηση της κίνησης ή μικρή κατανομή bit στον κωδικοποιητή.
- ❖ **Θόρυβος κουνουπιού:** Αυτό το πρόβλημα μπορεί να παρατηρηθεί στις άκρες κειμένων ή λογοτύπων που παρουσιάζονται στην οθόνη, αλλά και άλλων αντικειμένων που έχουν αιχμηρές άκρες. Οι αιχμηρές άκρες προκαλούν όρους υψηλής συχνότητας του μετασχηματισμού DCT, οι οποίοι κβαντίζονται λανθασμένα και απεικονίζονται στην οθόνη δημιουργώντας αυτό το πρόβλημα.



- ❖ **Βρώμικο παράθυρο:** Το συγκεκριμένο πρόβλημα εμφανίζεται σα λωρίδα ή θόρυβος που παραμένει σταθερός καθώς αντικείμενα κινούνται από κάτω του. Σ' αυτήν την περίπτωση, ο κωδικοποιητής μπορεί να μη στέλνει αρκετά bit για να κωδικοποιηθεί το παραμένον λάθος στα Ρ και Β πλαίσια.
- ❖ **Κυματιστός θόρυβος:** Αυτό το πρόβλημα γίνεται ορατό κατά τη διάρκεια αργών πλάνων σε σκηνές με πολλές λεπτομέρειες, όπως το πλήθος σε ένα στάδιο. Η άκομψη κβάντιση των όρων υψηλής συχνότητας προκαλείται από τέτοιες εικόνες.

Έτσι, συγκεκριμένοι τύποι κίνησης δεν ταιριάζουν στο MPEG-2 μοντέλο και παρουσιάζονται προβληματικοί. Αυτοί οι τύποι κίνησης περιλαμβάνουν:

- ❖ Ζουμ.
- ❖ Περιστροφές.
- ❖ Συμπαγή κινούμενα αντικείμενα.

Επίσης, συγκεκριμένα στοιχεία εικόνας δεν μπορούν να προβλεφθούν σωστά. Αυτά τα στοιχεία είναι:

- ❖ Σκιές.
- ❖ Αλλαγές στη φωτεινότητα.
- ❖ Περιοχές με πολλές λεπτομέρειες.
- ❖ Αθροιστικός θόρυβος.

Παρόλα αυτά, προσπάθειες γίνονται για να ελαχιστοποιηθούν αυτές οι ατέλειες. Ένα από τα δυνατά σημεία του MPEG-2, είναι ότι επιτρέπει ποικιλία και καινοτομίες στη σχεδίαση του κωδικοποιητή.

#### **4.6.3 Το πρότυπο κωδικοποίησης MPEG-4.**

Το MPEG-4 είναι ένα πρότυπο ISO/IEC που αναπτύχθηκε από την MPEG το 1998 και το οποίο έχει επιλεχθεί ως ένα από τα πρότυπα για μετάδοση πολυμέσων για τα ασύρματα δίκτυα 3ης και 4ης γενιάς. Το πρότυπο αυτό αναπτύχθηκε με σκοπό να γίνει το επόμενο πρότυπο των πολυμεσικών εφαρμογών, το οποίο θα υποστηρίζει πολύ χαμηλούς ρυθμούς μετάδοσης ψηφίων. Σε αντίθεση με το MPEG-1 και το MPEG-2 που προηγήθηκαν και που επικεντρώνονταν σε αποδοτικότερη συμπίεση (compression efficiency), το MPEG-4 δίνει έμφαση σε νέες λειτουργίες. Επιπρόσθετα, δίνει στους χρήστες την ευκαιρία να επεξεργάζονται με διάφορους τρόπους τόσο τις ηχητικές πληροφορίες όσο και τις πληροφορίες μιας σκηνής video, αλλά και να συγχωνεύουν το συνθετικό και το φυσικό ήχο, καθώς και τις πληροφορίες video. Παρέχει υψηλά επίπεδα ευελιξίας και ελαστικότητας, έτσι ώστε να μπορεί να εκμεταλλευθεί τις αναπτυσσόμενες τεχνολογίες, ειδικά στα ασύρματα επικοινωνιακά συστήματα. Επιπλέον, υποστηρίζει ένα ευρύ φάσμα ρυθμών μετάδοσης ψηφίων που εκτίνεται από 5 έως 64 kbits/s για κινητές εφαρμογές και μέχρι 4 Mbits/s για εφαρμογές τηλεόρασης. Το πρότυπο εισάγει νέες λειτουργίες video οι οποίες περιγράφονται στο MPEG-4 Proposal Package Description (MPEG AOE Group, 1995) και οι οποίες μπορούν να ομαδοποιηθούν σε τρεις κατηγορίες:

- ❖ **Λειτουργία αλληλεπίδρασης βασισμένης στο περιεχόμενο (content-based interactivity), αποδοτικότητας της κωδικοποίησης (coding efficiency):** Η λειτουργία αλληλεπίδρασης βασισμένης στο περιεχόμενο περιλαμβάνει μια σειρά από διαδικασίες, όπως επεξεργασία του περιεχομένου, διόρθωση της ακολουθίας των bit, υβριδική φυσική αλλά και τεχνητή κωδικοποίηση δεδομένων, και βελτιωμένη, χρονικά τυχαία πρόσβαση. Στα δύο προηγούμενα πρότυπα MPEG, υπήρχε η δυνατότητα για επεξεργασία της ακολουθίας δεδομένων video μόνο στο αρχικό επίπεδο. Επομένως, η συμπίεσμένη σειρά video έπρεπε να αποκωδικοποιηθεί στην αρχική της μορφή προτού κωδικοποιηθεί στην επιθυμητή συμπίεσμένη ακολουθία ψηφίων. Εδώ, υπάρχει η δυνατότητα για πρόσβαση και επεξεργασία των οπτικών αντικειμένων στο πεδίο που αυτά έχουν ήδη συμπίεστεί. Αυτό έχει ως αποτέλεσμα την ελάττωση της πολυπλοκότητας, των απαιτήσεων σε εύρος ζώνης και της καθυστέρησης, αλλά και καμία απώλεια σε ποιότητα. Επίσης παρέχει τα μέσα για

αρμονική αλληλεπίδραση των αντικειμένων video διαφόρων προελεύσεων, τα οποία μπορεί να είναι είτε φυσικά, είτε τεχνητά. Έτσι, καθίσταται δυνατό να κωδικοποιηθεί μια σειρά video με τέτοιο τρόπο ώστε να επιτρέπεται η ξεχωριστή αποκωδικοποίηση και ανακατασκευή των αντικειμένων σε μια σκηνή, αλλά και η επεξεργασία της αρχικής σκηνής μέσω απλών χειρισμών στην ακολουθία των ψηφίων. Τελικά, σε ότι αφορά στη λειτουργία αλληλεπίδρασης βασισμένης στο περιεχόμενο, παρέχει επαρκείς μεθόδους για τυχαία πρόσβαση, εντός ενός περιορισμένου χρονικού διαστήματος αλλά και με καλή ανάλυση σε τμήματα (π.χ. πλαίσια ή αντικείμενα) μιας σειράς video. Διαθέτει εργαλεία και αλγορίθμους που προσφέρουν καλύτερη συμπίεση δίχως να επηρεάζουν την ποιότητα του video. Έτσι, καθίσταται κατάλληλο για χρήση σε εφαρμογές όπως video-συνδιάσκεψη και ροή video πραγματικού χρόνου σε δίκτυα κινητής τηλεφωνίας. Επιπλέον, το MPEG-4 παρέχει μεθόδους για αποδοτική κωδικοποίηση των πολλαπλών όψεων μιας σκηνής.

- ❖ **Λειτουργία καθολικής πρόσβασης (universal access):** Η λειτουργία καθολικής πρόσβασης που προτείνεται στο πρότυπο MPEG-4 καθορίζει τη σθεναρότητα σε επιρρεπή σε σφάλματα περιβάλλοντα και την κλιμάκωση που βασίζεται στο περιεχόμενο. Το πρόβλημα με πολλούς αλγορίθμους συμπίεσης είναι ότι, όταν αυξάνεται ο λόγος συμπίεσης, αυξάνεται, επίσης, και ο κίνδυνος σφάλματος στα ψηφία της μεταδιδόμενης ακολουθίας. Τα επικοινωνιακά κανάλια των ασύρματων δικτύων υπόκεινται σε θόρυβο, διαλείψεις, ράδιο-σκίαση καθώς και σε παρεμβολές, παράγοντες, δηλαδή, που κάνουν δύσκολη την αξιόπιστη μετάδοση μιας κωδικοποιημένης ακολουθίας video. Το MPEG-4 παρέχει σθεναρότητα έναντι σφαλμάτων για εφαρμογές αποθήκευσης, όπως video κατ' απαίτηση, καθώς και για επικοινωνιακές εφαρμογές, όπως video συνδιασκέψεις σε επιρρεπή σε σφάλματα ετερογενή περιβάλλοντα. Εν ολίγοις το MPEG-4 είναι ένα πρότυπο παγκόσμιας εμβέλειας για τη δημιουργία πολυμέσων, τη διάθεση και την αναπαραγωγή τους στο internet .

Το MPEG-1 έφερε επανάσταση στη βιομηχανία των CD, το MPEG-2 σχεδιάστηκε για την ανάπτυξη των DVD και το MPEG-4 έφερε επανάσταση στο διαδίκτυο. Χρησιμοποιείται στα κινητά μέχρι και τη δορυφορική τηλεόραση. Το MPEG-4 σχεδιάστηκε μέσα στα πλαίσια του διεθνούς οργανισμού προτυποποίησης (ISO), που προδιέγραψε τα γνωστά πρότυπα MPEG-1, MPEG-2. Το MPEG-4 είναι περιλαμβάνει εκατοντάδες ερευνητές και τεχνικούς. Τα πεδία στα οποία λειτουργεί το MPEG-4 είναι δύο:

- ❖ Ψηφιακές τηλεοράσεις.
- ❖ Αλληλεπιδραστικά πολυμέσα και γραφικά.

Το MPEG-4 είναι ανοικτό πρότυπο και είναι δύσκολο άλλη τεχνολογία να αναπτυχθεί σε τέτοιο βάθος και να υποστηριχτεί από τόσες εταιρείες, καθώς είναι δουλειά πολλών χρόνων μοιρασμένη από εκατοντάδες εταιρείες. Είναι το μόνο πρότυπο, που μπορεί να εκμεταλλευτεί τις δυνατότητες, που προσφέρει η ψηφιακή επανάσταση. Αναπτύσσει την συμπίεση του ήχου και του video επιτρέποντας τη μεταφορά των δεδομένων στο δίκτυο με υψηλή ποιότητα. Προσφέρει ένα πρότυπο για διάφορες μορφές πολυμέσων, όπως είναι το κείμενο, φωτογραφίες, κινούμενα σχέδια δυσδιάστατα και τρισδιάστατα αντικείμενα. Προσφέρει επίσης μια ποικιλία προφίλ για εξειδικευμένες εφαρμογές, έτσι ώστε οι χρήστες να διευκολύνονται, καθώς περιορίζονται στην απλή εφαρμογή των προφίλ αυτών και δεν είναι ανάγκη να προβούν σε εξειδικευμένες ενέργειες. Η τεχνολογία του MPEG-4 μεταφέρει εύκολα στο διαδίκτυο τις διάφορες εφαρμογές των πολυμέσων σε όχι απαραίτητα υψηλές ταχύτητες.

#### **4.6.3.1. Επανάσταση στην λειτουργικότητα.**

Το MPEG-4 αποτελεί επανάσταση στη λειτουργία των πολυμέσων. Εκτός των άλλων μπορεί να δουλεύει σε υπάρχουσες δομές, όπως σε περιβάλλοντα λειτουργίας του MPEG-2. Αυτή είναι μια πολύ σημαντική λειτουργία, γιατί με αυτό τον τρόπο δεν χρειάζεται να απορρίπτουμε παλιότερες τεχνολογίες, αλλά απλώς να τις βελτιώνουμε. Ακόμα με την κοινή χρήση του MPEG-

2 και του MPEG-4 υπάρχει μια συμβατότητα του παλιού με το καινούριο και με τον τρόπο αυτό αποφεύγουμε την απώλεια δεδομένων. Η σύνδεση των προτύπων υποδηλώνει και τα προηγούμενα πρότυπα είναι διαχρονικά και μπορούν να ανταπεξέλθουν σε σύγχρονες ανάγκες. Επίσης οι τεχνικές συμπίεσης είναι περίτεχνες, δημιουργώντας μια παρουσίαση των πολυμέσων πιο αποτελεσματική και εύκολη στη χρήση. Το πρότυπο αυτό προσφέρει επαναστατικά εργαλεία για το περιεχόμενο, με το οποίο μεταφέρεται το ελάχιστο από τις γενικές και άχρηστες πληροφορίες. Αυτό επιτρέπει χωριστή αλληλεπίδραση με το κάθε αντικείμενο και σωστή μεταφορά σε όλα τα δίκτυα, όπως για παράδειγμα οι υπότιτλοι σε μια ταινία, οι οποίοι μπορεί να είναι αναγκαίοι για την κατανόηση μιας ταινίας, παραμένουν της ίδιας ποιότητας ακόμα και σε υψηλή συμπίεση. Επιλέγουμε λοιπόν πιο μέρος της πληροφορίας επιθυμούμε να διατηρηθεί σε καλύτερη ποιότητα. Έτσι τα δεδομένα, που δεν είναι αναγκαία, δέχονται μια μεγαλύτερη συμπίεση για την εξοικονόμηση του απαραίτητου χώρου. Αυτός είναι άλλωστε ήταν και ο σκοπός του προτύπου εξ αρχής.

#### **4.6.3.2. Αποδοχή του MPEG-4.**

Το MPEG-4 είναι ένα πρότυπο που γίνεται αποδεκτό από τα περισσότερα λογισμικά για την αναπαραγωγή πολυμέσων. Είναι πρότυπο που χρησιμοποιείται ευρέως στο Internet, κυρίως στις ψηφιακές βιβλιοθήκες πολυμέσων, ειδικά σε μορφή video. Έχει ήδη γίνει καθιερωμένο πρότυπο για όλες τις συσκευές, που χρησιμοποιούν πολυμεσικές εφαρμογές μικρής χωρητικότητας αλλά αξιόλογης ποιότητας, όπως είναι τα κινητά τρίτης γενιάς. Χρησιμοποιείται σε λογισμικά της Microsoft αλλά και πολλών άλλων εταιρειών. Εκτελείται σε εφαρμογές όπως Quicktime 6 και το Media Player της Microsoft που είναι από τα πιο διαδεδομένα και πιο εύχρηστα προγράμματα στο χώρο των υπολογιστών. Χρησιμοποιείται ακόμα και σε φορητές συσκευές των υπολογιστών, όπως είναι τα PDA, συσκευές ιδιαίτερα χρήσιμες για την εύκολη μεταφορά προσωπικών δεδομένων. Το MPEG-4 δεν περιορίζεται μόνο στο χώρο των υπολογιστών αλλά εμπλέκεται σε όλες τις συσκευές ψυχαγωγίας. Χρησιμοποιείται σε ψηφιακές τηλεοράσεις και τις ψηφιακές κάμερες σε συνδυασμό με τα προϋπάρχοντα πρότυπα MPEG-1 και MPEG-2.

#### **4.6.3.3. Προφίλ και επίπεδα του MPEG-4.**

Το MPEG-4 αποτελείται από ένα μεγάλο αριθμό εργαλείων, τα οποία δεν είναι χρήσιμα σε όλες τις εφαρμογές. Υπάρχουν τα λεγόμενα προφίλ για την οργάνωση των εργαλείων αυτών, όπως το MPEG-4 Advanced Simple Visual Profile, όπου περιέχει ¼ αποκατάσταση της κίνησης, B-frames, αλλά δεν περιέχει σχηματοποιημένο κωδικοποιημένο video. Τα προφίλ αυτά επιτρέπουν στους χρήστες να διαλέξουν το κατάλληλο από μια ποικιλία εργαλείων για να εκτελέσουν τη λειτουργία που αυτοί επιθυμούν. Τα προφίλ υπάρχουν σε ένα αριθμό επιπέδων, που παρέχουν έναν τρόπο να περιορίσουν την πολυπλοκότητα των υπολογιστών.

#### **4.6.3.4. Τα βασικά μέρη του MPEG-4.**

Το MPEG-4 αποτελείται από στενά συνδεδεμένα μέρη που μπορούν να εφαρμοστούν χωριστά ή να συνδυαστούν με άλλα μέρη. Η δομή του έχει ως εξής:

- ❖ Συστήματα(Systems)
- ❖ Οπτική μορφή(Visual)
- ❖ Ήχο(Audio)
- ❖ Προσαρμογή(Conformance):Διευκρινίζει πως να δοκιμάσουμε μια εφαρμογή του MPEG-4.
- ❖ Αναφορά Λογισμικού(Reference Software):Μπορεί να χρησιμοποιηθεί για να ξεκινήσουμε την εφαρμογή του προτύπου.
- ❖ DMIF (Delivery Multimedia Integration Framework): Προδιαγράφει μια διεπαφή μεταξύ εφαρμογής και δικτύου/ αποθετηρίου.
- ❖ Προδιαγραφή ενός βελτιωμένου κωδικοποιητή video.

- ❖ Η μεταφορά δεδομένων είναι η βασική αρχή του προτύπου αλλά δεν είναι προδιαγραμμένη. Το 8 μέρος προδιαγράφει πως να χαρτογραφήσουμε τη ροή δεδομένων του MPEG-4 σε μεταφορά στο internet.

#### **4.6.3.5. Πλεονεκτήματα του MPEG-4.**

Ο καλύτερος τρόπος για να καταλάβουμε τις δυνατότητες του MPEG-4 είναι με τη σύγκριση με το MPEG-2. Στο MPEG-2 το περιεχόμενο δημιουργείται από πολλές πηγές όπως κινούμενο video , γραφικά, κείμενο. Μετά τη συγκρότηση τους σε ένα σχέδιο από εικονοστοιχεία, κωδικοποιούνται όλα, ανεξαρτήτως μορφής σε εικονοστοιχεία κινούμενου video. Από την πλευρά του καταναλωτή η αποκωδικοποίηση είναι ένα άμεσο εγχείρημα. Έχεις τη δυνατότητα να προσθέσεις γραφικά και άλλα στοιχεία αλλά δεν υπάρχει τρόπος να τα σβήσεις. Από την άλλη, το MPEG-4 είναι δυναμικό. Διαφορετικά αντικείμενα μπορούν να κωδικοποιηθούν και να μεταδοθούν χωριστά από τον κωδικοποιητή και σε δική τους ροή. Η σύνθεση τους πραγματοποιείται μετά την αποκωδικοποίηση. Για να επιτευχθεί αυτή η σύνθεση το MPEG-4 περιλαμβάνει μια ειδική γλώσσα περιγραφής σκηνών, αποκαλούμενη BIFS (Binary Format for Scenes), η οποία, δεν περιγράφει μόνο το πότε και πού τα αντικείμενα θα εμφανιστούν στη σκηνή, αλλά και συμπεριφορά κάτω υπό συνθήκες, όπως για παράδειγμα αντικείμενα που αντιδρούν σε καταστάσεις. Όλα τα αντικείμενα μπορούν να κωδικοποιηθούν με δικό τους σχήμα κωδικών ενώ το video κωδικοποιείται σαν video , το κείμενο σαν κείμενο, τα γραφικά σαν γραφικά αντί να θεωρούνται όλα τα εικονοστοιχεία σαν κινούμενο video.

**Κεφάλαιο 5<sup>ο</sup>**  
**Πειραματικό Μέρος**  
**και Εφαρμογές.**

**5.1. Εισαγωγή στο Matlab.**

Το MATLAB είναι ένα λογισμικό πακέτο για υψηλής απόδοσης αριθμητικούς υπολογισμούς (numerical computations). Παρέχει στο χρήστη ένα διαδραστικό περιβάλλον με χιλιάδες ενσωματωμένες συναρτήσεις, κατάλληλες για την υλοποίηση απαιτητικών υπολογιστικών αναλύσεων, γραφημάτων καθώς επίσης και για την παραγωγή διαφόρων animations. Επιπλέον, το MATLAB προσφέρει τη δυνατότητα επέκτασης σε ποικίλα πεδία εφαρμογών με τη αξιοποίηση την υψηλού επιπέδου γλώσσας προγραμματισμού, την οποία διαθέτει σε όλες τις εκδόσεις του. Για λόγους πληρότητας, να αναφερθεί ότι το όνομα MATLAB προέρχεται από τις λέξεις MATrix και LABoratory. Το MATLAB αποτελεί ένα εξελιγμένο υπολογιστικό εργαλείο, το οποίο μπορεί να βρει εφαρμογή σε διάφορους τομείς της επιστήμης αλλά βέβαια και της πράξης, όπως για παράδειγμα τη μηχανική, την ιατρική, τις θετικές επιστήμες (Μαθηματικά – Φυσική), την οικονομία καθώς και γενικά τη βιομηχανική παραγωγή. Μάλιστα, το φάσμα των εφαρμογών του συγκεκριμένου πακέτου λογισμικού διευρύνεται συνεχώς και περισσότερο, αναδεικνύοντας με αυτό τον τρόπο τις πολλαπλές δυνατότητες του, όπως:

- ❖ Υψηλή απόδοση και ταχύτητα υπολογιστικών αναλύσεων.
- ❖ Δυνατότητα προσομοίωσης φυσικών συστημάτων.
- ❖ Δυνατότητα υλοποίησης αλγορίθμων.
- ❖ Δυνατότητα αμφίδρομης επικοινωνίας με πληθώρα άλλων προγραμμάτων και εφαρμογών.
- ❖ Υψηλής ποιότητας γραφικές απεικονίσεις και animations.
- ❖ Δυνατότητα σύνδεσης με διάφορες συσκευές καταγραφής.
- ❖ Φιλικότητα προς το χρήστη και διαδραστικός χαρακτήρας.

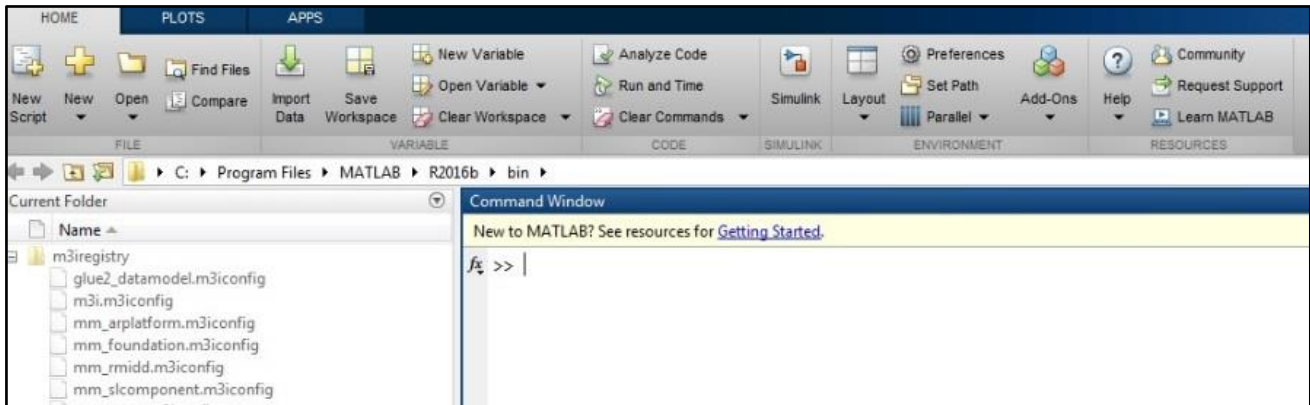
Οι ενσωματωμένες συναρτήσεις του λογισμικού παρέχουν τα απαραίτητα πακέτα εργαλείων για υπολογισμούς γραμμικής άλγεβρας, ανάλυσης δεδομένων, επεξεργασία σημάτων, αριθμητικές λύσεις κανονικών διαφορικών εξισώσεων. Οι περισσότερες από τις προαναφερόμενες συναρτήσεις εφαρμόζουν την πλέον πρόσφατη και εξελιγμένη γνώση στο κάθε τομέα επιστήμης (state-of-the-art algorithms). Επίσης ο χρήστης έχει τη δυνατότητα να αναπτύξει τις δικές του συναρτήσεις, κάνοντας χρήση της δικής του γλώσσας προγραμματισμού. Από τη στιγμή που θα αναπτυχθούν οι συναρτήσεις αυτές, λειτουργούν ως ενσωματωμένες συναρτήσεις του εν λόγω λογισμικού. Επίσης παρέχονται από το MATLAB πολλές προαιρετικές εργαλειοθήκες, οι οποίες προορίζονται για την ανάπτυξη ειδικών εφαρμογών, όπως συμβολικοί υπολογισμοί (symbolic computation), επεξεργασία εικόνων (image processing), στατιστική (statistics), σχεδιασμός ελέγχου συστημάτων (control system design), νευρωνικά δίκτυα (neural networks), ασαφή λογική (fuzzy logic). Η λίστα με τις διαθέσιμες εργαλειοθήκες συνεχώς διευρύνεται (στην τελευταία έκδοση του λογισμικού ο αριθμός των εργαλειοθηκών έχει ξεπεράσει τις 50). Η βασική δομική μονάδα του MATLAB είναι ο πίνακας (matrix) και ο θεμελιώδης τύπος δεδομένου είναι το διάνυσμα (array). Όλα τα υπόλοιπα στοιχεία του λογισμικού πακέτου συντίθεται από τι προαναφερόμενες βασικές μονάδες. Σε αυτό το σημείο να αναφερθεί ότι στο MATLAB δεν υπάρχει ανάγκη για δήλωση των διαστάσεων ενός πίνακα, ο οποίος χρησιμοποιείται σε μια εφαρμογή. Το λογισμικό, αυτό είναι με τέτοιο τρόπο δομημένο, ώστε όλοι οι υπολογισμοί να μετατρέπονται ουσιαστικά σε υπολογισμούς μεταξύ πινάκων και διανυσμάτων.

**5.2. Χρήση παραδειγμάτων και επεξήγησή τους.**

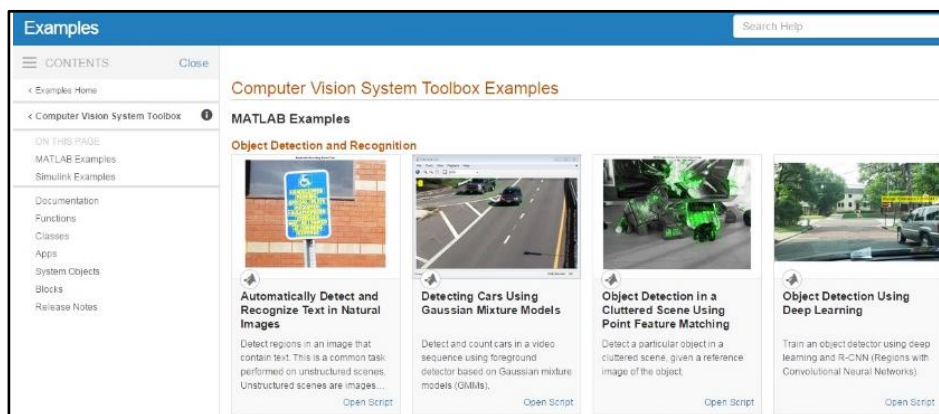
Για το πειραματικό μέρος της εργασίας αυτής, χρησιμοποιήθηκε το λογισμικό Matlab R2016b. Σε πρώτο στάδιο, μεταφερόμαστε στο Command Window, ή αλλιώς Παράθυρο εντολών και αποτελεί το βασικό παράθυρο. Χαρακτηρίζεται από το σύμβολο προτροπής (>>, MATLAB command prompt). Το σύνολο των εντολών, συμπεριλαμβανομένων και των εντολών, που αναπτύσσει ο ίδιος ο χρήστης, πληκτρολογούνται στο Παράθυρο εντολών, πάντα με τη χρήση του συμβόλου της προτροπής (μπροστά από την κάθε εντολή).

Επίσης, στο τμήμα αυτό της επιφάνειας εργασίας του MATLAB, πραγματοποιείται η εισαγωγή των απαραίτητων δεδομένων για μια εφαρμογή, καθώς χρησιμεύει και για την εξαγωγή των επιζητούμενων αποτελεσμάτων.

Αρχικά πληκτρολογούμε την εντολή `cd` στο Παράθυρο Εντολών, η οποία εμφανίζει τον τρέχοντα φάκελο, στον οποίο βρισκόμαστε.



Εν συνεχεία, πληκτρολογούμε στο παράθυρο εντολών, την εντολή `visiondemos`, όπως παρατηρούμε και παρακάτω, καθώς μας μεταφέρει στο Computer Vision System Toolbox του Matlab, που περιέχει μία πληθώρα από παραδείγματα προσομοίωσης Video.



### 5.3. Ανάλυση Παραδειγμάτων και Εφαρμογές στην Καθημερινότητα.

#### Detecting Cars Using Gaussian Mixture Models.

##### Επεξήγηση Παραδείγματος.

Το συγκεκριμένο παράδειγμα μας παρουσιάζει πώς να ανιχνεύουμε και να μετράμε τα αυτοκίνητα σε μία ακολουθία video, χρησιμοποιώντας ανιχνευτή προσκηνίου, βασιζόμενο στα Gaussian Μοντέλα Μείγματος. Ανιχνεύοντας και μετρώντας τον αριθμό των αυτοκινήτων, μπορεί να χρησιμοποιηθεί στο να αναλυθούν τα μοντέλα κυκλοφορίας. Επιπρόσθετα, η ανίχνευση, αποτελεί ένα πρώτο βήμα, κυρίως στο να επεξεργαζόμαστε πιο εξελιγμένες εργασίες,

όπως η παρακολούθηση ή ακόμη και η κατηγοριοποίηση των οχημάτων ανά του είδους τους. Το παρόν παράδειγμα, επεξηγεί πώς να χρησιμοποιούμε τον ανιχνευτή προσκηνίου, και την ανάλυση κυψελίδων στην ανίχνευση και στο μέτρημα των αυτοκινήτων σε μία αλληλουχία video. Θεωρείται πως η κάμερα είναι ακίνητη. Το παράδειγμα επικεντρώνεται στην ανίχνευση των αντικειμένων.

Κώδικας που χρησιμοποιήθηκε.

### **%% Step 1 - Import Video and Initialize**

#### **Foreground Detector**

```
foregroundDetector =
vision.ForegroundDetector('NumGaussians',
    3, ... 'NumTrainingFrames', 50);
videoReader =
vision.VideoFileReader('visiontraffic.avi');
for i = 1:150
    frame = step(videoReader);
    foreground = step(foregroundDetector,
        frame);
end
figure; imshow(frame); title('Video Frame');
figure; imshow(foreground);
title('Foreground');
```

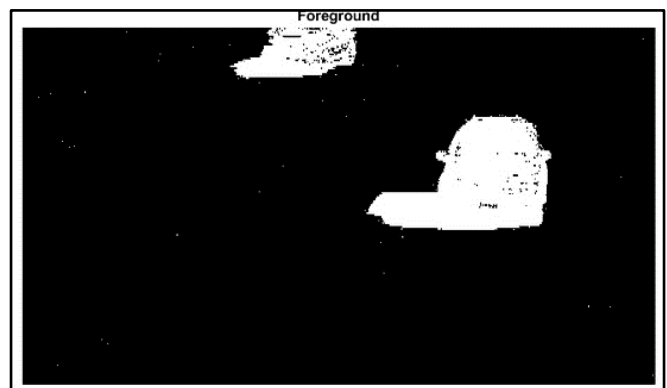
### **%% Step 2 - Detect Cars in an Initial Video**

#### **Frame**

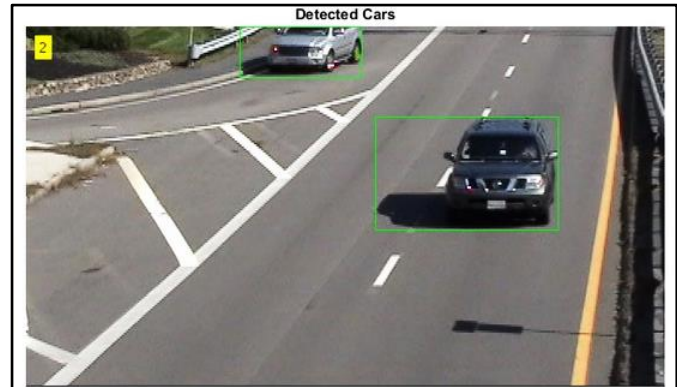
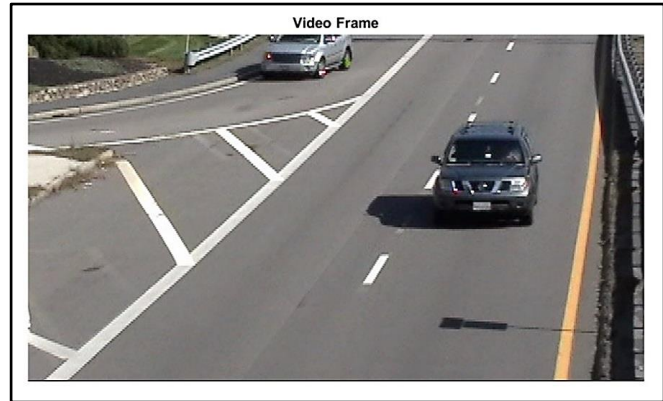
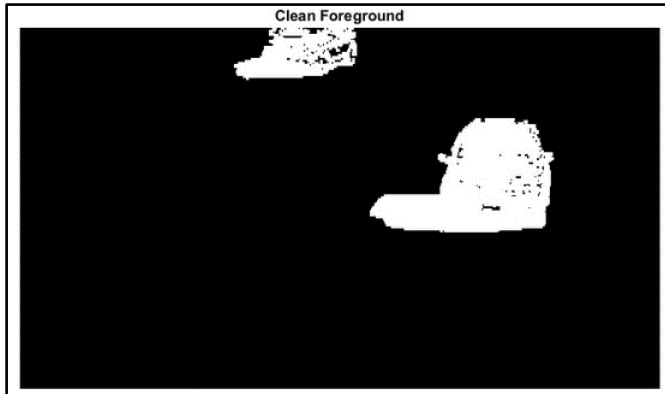
```
se = strel('square', 3);
filteredForeground = imopen(foreground,
    se);
figure; imshow(filteredForeground);
title('Clean Foreground');
blobAnalysis =
vision.BlobAnalysis('BoundingBoxOutputPort'
    , true, ...
'AreaOutputPort', false, 'CentroidOutputPort',
    false, ...
'MinimumBlobArea', 150);
bbox = step(blobAnalysis,
    filteredForeground);
```

```
result = insertShape(frame, 'Rectangle', bbox,
    'Color', 'green');
numCars = size(bbox, 1);
result = insertText(result, [10 10], numCars,
    'BoxOpacity', 1, ...
    'FontSize', 14);
figure; imshow(result); title('Detected Cars');
%% Step 3 - Process the Rest of Video
Frames
videoPlayer = vision.VideoPlayer('Name',
    'Detected Cars');
videoPlayer.Position(3:4) = [650,400];
se = strel('square', 3);
while ~isDone(videoReader)
    frame = step(videoReader);
    foreground = step(foregroundDetector,
        frame);
    filteredForeground = imopen(foreground,
        se);
    bbox = step(blobAnalysis,
        filteredForeground);
    result = insertShape(frame, 'Rectangle', bbox,
        'Color', 'green');
    numCars = size(bbox, 1);
    result = insertText(result, [10 10], numCars,
        'BoxOpacity', 1, ...
        'FontSize', 14);
    step(videoPlayer, result);
end
release(videoReader);
displayEndOfDemoMessage(mfilename)
```

Στιγμιότυπα από το Video.







### Lane Departure Warning System.

#### Επεξήγηση Παραδείγματος.

Το δεύτερο παράδειγμα, με ονομασία «Lane Departure Warning System» μας παρουσιάζει πώς να ανιχνεύουμε σηματοδότες οδοστρωμάτων, σε μία ροή video, και στο πώς να δώσουμε έμφαση στη γραμμή, στην οποία οδηγείται το όχημα. Η παραπάνω πληροφορία μπορεί να χρησιμοποιηθεί με σκοπό να προβλέψουμε μία ακούσια αναχώρηση, από τη γραμμή, και να δοθεί μία έγκαιρη προειδοποίηση. Επιπρόσθετα, παρατηρούμε πως, ανιχνεύονται και παρακολουθούνται σηματοδότες οδοστρώματος, κατά τη διάρκεια της ροής video, και ανιχνεύει-παρακολουθεί εάν ο οδηγός του οχήματος μετακινείται από τη γραμμή. Παράλληλα, μας υποδεικνύει πώς να χρησιμοποιούμε τις εξής μεθόδους;

- ❖ Μετασχηματισμός Hough.
- ❖ Σύστημα HoughLines.
- ❖ Σύστημα LocalMaximaFinder.

Με τη βοήθεια των ανωτέρω μεθόδων, δημιουργούμε ανίχνευση γραμμής, καθώς και αλγορίθμους παρακολούθησης. Το παράδειγμα, εισάγει τον αλγόριθμο, ακολουθώντας τα παρακάτω βήματα:

- ❖ Ανιχνεύει τους δείκτες των λωριδών του οδοστρώματος, στο τρέχον video πλαισίου.
- ❖ Ταιριάζει τις τωρινές λωρίδες του οδοστρώματος με αυτές που είχαν ανιχνευθεί, στο προηγούμενο video πλαισίου.
- ❖ Αναζητά τον αριστερό και το δεξί δείκτη λωρίδας.
- ❖ Τοποθετεί ένα μήνυμα προειδοποίησης, εάν το όχημα μετακινείται κατά μήκος ενός από τους δείκτες λωρίδας κυκλοφορίας.

Προκειμένου να επεξεργαστούν ακολουθίες video χαμηλής ποιότητας, όπου οι δείκτες λωριδών, μπορεί να είναι δύσκολο να εντοπιστούν ή να κρύβονται πίσω από αντικείμενα, το παράδειγμα περιμένει έναν δείκτη λωρίδας να εμφανίζεται σε πολλαπλά πλαίσια πριν θεωρήσει ότι ο δείκτης είναι έγκυρος. Τέλος, χρησιμοποιεί την ίδια διαδικασία για να αποφασίσει πότε θα ξεκινήσει να αγνοεί ένα δείκτη λωρίδας.

Κώδικας που χρησιμοποιήθηκε.

```

        DrawPoly = 1;
        NumRows = 120;
        MaxLaneNum = 20;
        ExpLaneNum = 2;
        Rep_ref = zeros(2, MaxLaneNum);
        Count_ref = zeros(1, MaxLaneNum);
        TrackThreshold = 75;
        LaneColors = single([0 0 0;1 1 0; 1 1 0; 1 1 1;1
            1 1]);
        frameFound = 5;
        frameLost = 20;
        startIdxRho_R = 415;
        NumRhos_R = 11;
        startIdxTheta_R = 1;
        NumThetas_R = 21;
        startIdxRho_L = 380;
        NumRhos_L = 36;
        startIdxTheta_L = 146;
        NumThetas_L = 21;
        offset = int32([0, NumRows, 0, NumRows]);

        OutMsgPre = OutMsg;
        Broken = false;
        warningTextColors = {[1 0 0], [1 0 0], [0 0 0], [0
            0 0]};
        while ~isDone(hVideoSrc)
            RGB = step(hVideoSrc);
            Imlow = RGB(NumRows+1:end, :, :);
            Imlow = rgb2gray(Imlow);
            I = imfilter(Imlow, [-1 0 1], 'replicate','corr');
            I(I < 0) = 0;
        );

        Line = [Rho(Idx1(:, 1)); Theta(Idx1(:, 2))];
        Enable = [ones(1,Count1) zeros(1,
            ExpLaneNum-Count1)];
        [Rep_ref, Count_ref] =
        videolanematching(Rep_ref, Count_ref, ...
        MaxLaneNum, ExpLaneNum, Enable, Line, ...
        TrackThreshold, frameFound+frameLost);
        Pts = step(hHoughLines1, Rep_ref(2,:),
            Rep_ref(1,:), Imlow);
        [TwoValidLanes, NumNormalDriving,
            TwoLanes, OutMsg] = ...
        videodeparturerearning(Pts, Imlow,
            MaxLaneNum, Count_ref, ...
            NumNormalDriving, OutMsg);
        1 = Normal driving, 2 = Left lane departure

```

```

        hVideoSrc =
        vision.VideoFileReader('viplanedeparture.mp4')
            ;
        hHoughLines1 =
        vision.HoughLines('SineComputation',
            'Trigonometric function');
        hHoughLines3 =
        vision.HoughLines('SineComputation',
            'Trigonometric function');
        warnText = {sprintf('Right\nDeparture'), "",
            sprintf(' Left\n Departure')};
        warnTextLoc = [120 170;-1 -1; 2 170];
        lineText = {'', ...
            sprintf('Yellow\nBroken'),
            sprintf('Yellow\nSolid'), ...
            sprintf('White\nBroken'),
            sprintf('White\nSolid')};
        hVideoOut = vision.VideoPlayer;
        Frame = 0;
        NumNormalDriving = 0;
        OutMsg = int8(-1);
        I(I > 1) = 1;
        th = multithresh(I);
        [H, Theta, Rho] = hough(I > th);
        Theta = Theta * pi / 180;
        H1 = H;
        H1(:, 1:12) = 0;
        H1(:, end-12:end) = 0;
        Idx1 = houghpeaks(H1, ExpLaneNum,
            'NHoodSize', [301 81], 'Threshold', 1);
        Count1 = size(Idx1,1)

        YCbCr =
        rgb2ycbcr(double(RGB(NumRows+1:240, :, :)));
        ColorAndTypeIdx =
        videodetectcolorandtype(TwoLanes, YCbCr);

        % INVALID_COLOR_OR_TYPE = int8(0);
        % YELLOW_BROKEN = int8(1); YELLOW_SOLID
            = int8(2);
        % WHITE_BROKEN = int8(3); WHITE_SOLID =
            int8(4).
        Frame = Frame + 1;
        if Frame >= 5
            TwoLanes1 = TwoLanes + [offset; offset];
            if DrawPoly && TwoValidLanes
                if TwoLanes(4,1) >= 239
                    Templ = TwoLanes1(3:4, 1);

```

```

else
  Templ = [0 239]';
end
if TwoLanes(4,2) >= 239
  Tempr = TwoLanes1(3:4, 2);
else
  Tempr = [359 239]';
end
Pts_poly = [TwoLanes1(:,1); Templ; Tempr; ...

```

```

txtColor =
single(warningTextColors(mod(Frame-1,4)+1));
  RGB = insertText(RGB,txtLoc,txt,'TextColor',
    txtColor, ...
    'FontSize',20, 'BoxOpacity', 0);
  for ii=1:2
  txtLoc = TwoLanes1([1 2], ii) + int32([0 -35]);
  lineTxt = lineText{ColorAndTypeIdx(ii)};
  txtColor = LaneColors(ColorAndTypeIdx(ii), :);

```

```

ShowThirdLane = Broken && (OutMsg~=1);
  if ShowThirdLane
    if OutMsg == 0
      Idx2 =
houghpeaks(H(startIdxRho_R:startIdxRho_R+N
  umRhos_R-1, ...
startIdxTheta_R:startIdxTheta_R+NumThetas_R
  -1), ...
  'NHoodSize', [7 7], 'Threshold', 1);
  Rhor = Rho(Idx2(:,1) + startIdxRho_R);
  Thetar = Theta(Idx2(:,2) + startIdxTheta_R);
  ThirdLane = step(hHoughLines3, Thetar, Rhor,
    Imlow);
    else
      Idx3 =
houghpeaks(H(startIdxRho_L:startIdxRho_L+N
  umRhos_L-1, ...
startIdxTheta_L:startIdxTheta_L+NumThetas_L-
  1),...
  'NHoodSize', [7 7], 'Threshold', 1);
name)

```

```

TwoLanes1(3:4,2); TwoLanes1(1:2,2)];
  RGB =
insertShape(RGB,'FilledPolygon',Pts_poly.',...
  'Color',[0 1 1],'Opacity',0.2);
  end
  RGB = insertShape(RGB,'Line',TwoLanes1',...
  'Color',{'yellow','magenta'});
  txt = warnText{OutMsg+1};
  txtLoc = warnTextLoc(OutMsg+1, :);

```

```

  RGB =
insertText(RGB,txtLoc,lineTxt,'TextColor',txtCol
  or, ...
  'FontSize',14, 'BoxOpacity', 0);
  end
  if OutMsgPre ~= OutMsg
  ColorType = ColorAndTypeIdx(2-(OutMsg ==
    2));
  Broken = ColorType == 2 || ColorType == 4;
  end

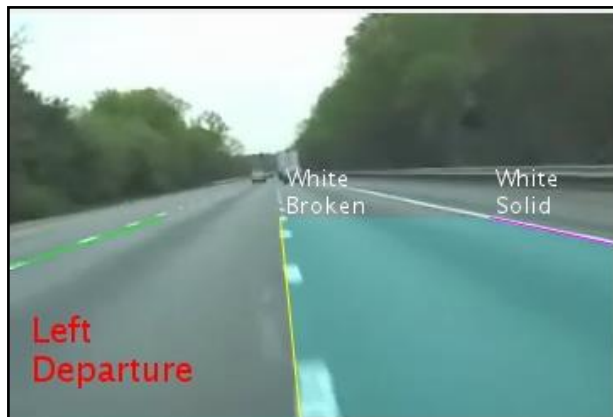
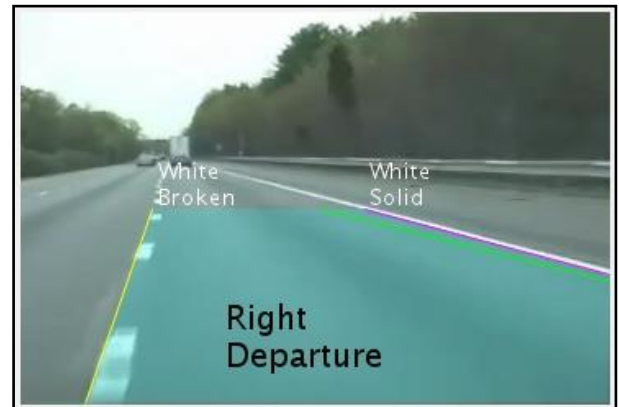
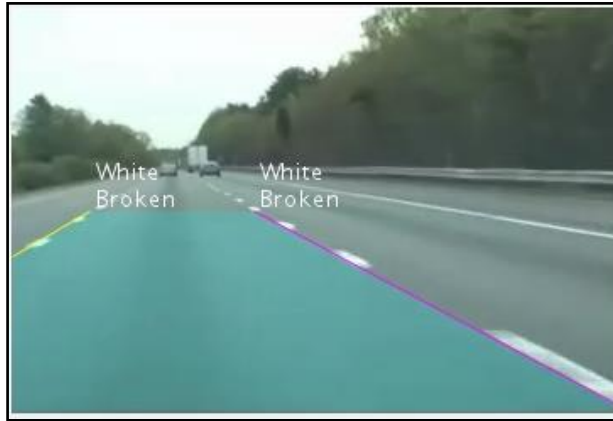
```

```

  Rhol = Rho(Idx3(:,1) + startIdxRho_L);
  Thetal = Theta(Idx3(:,2) + startIdxTheta_L);
  ThirdLane = step(hHoughLines3, Thetal, Rhol,
    Imlow);
    end
  OutThirdLane =
videoexclude3rdlane(ThirdLane,
  ShowThirdLane,...
  TwoLanes, TwoValidLanes, YCbCr);
  OutThirdLane = OutThirdLane(:) + offset(:);
  RGB =
insertShape(RGB,'Line',OutThirdLane.','Color','g
  reen');
  end
  end
  OutMsgPre = OutMsg
step(hVideoOut, RGB);
  end
  release(hVideoSrc);
displayEndOfDemoMessage(mfile

```

Στιγμιότυπα από το Video.



**Tracking pedestrians from a moving car.**

Επεξήγηση Παραδείγματος.

Σε αυτήν την περίπτωση, χρησιμοποιούμε το παράδειγμα του πώς να παρακολουθήσουμε πεζούς, χρησιμοποιώντας μία κάμερα, η οποία είναι τοποθετημένη σε ένα κινούμενο όχημα. Επιπρόσθετα, το παράδειγμα αυτό, μας υποδεικνύει την ευελιξία ενός συστήματος ανίχνευσης, προσαρμοσμένο σε μία κινούμενη κάμερα, η οποία είναι ιδανική για τις εφαρμογές της ασφάλειας στην αυτοκινητοβιομηχανία.

Κώδικας για το παραπάνω παράδειγμα.

```

function
PedestrianTrackingFromMovingCameraExample()
    videoFile = 'vippedtracking.mp4';
    scaleDataFile = 'pedScaleTable.mat';
    obj = setupSystemObjects(videoFile,
        scaleDataFile);
    tracks = initializeTracks();
    nextId = 1;
    option.ROI = [40 95 400 140];
    option.scThresh = 0.3;
    option.gatingThresh = 0.9;
    option.gatingCost = 100;
    option.costOfNonAssignment = 10;
    option.timeWindowSize = 16;
    option.confidenceThresh = 2;
    option.ageThresh = 8;
    option.visThresh = 0.6;
    
```

```

    cont = ~isDone(obj.reader);
    while cont
        frame = readFrame();
        [centroids, bboxes, scores] = detectPeople();
        predictNewLocationsOfTracks();
        [assignments, unassignedTracks,
        unassignedDetections] = ...
        detectionToTrackAssignment();
        updateAssignedTracks();
        updateUnassignedTracks();
        deleteLostTracks();

        createNewTracks();
        displayTrackingResults();
        cont = ~isDone(obj.reader) &&
        isOpen(obj.videoPlayer);
    end
    
```

```

function obj =
setupSystemObjects(videoFile,scaleDataFile)
    obj.reader =
        vision.VideoFileReader(videoFile,
            'VideoOutputDataType', 'uint8')

    obj.videoPlayer =
        vision.VideoPlayer('Position', [29, 597, 643,
            386]);

    'id', {}, ...
    'color', {}, ...
    'bboxes', {}, ...
    'scores', {}, ...
    'kalmanFilter', {}, ...
    'age', {}, ...
    'totalVisibleCount', {}, ...
    'confidence', {}, ...

    resizeMode = 1.5;
    frame = imresize(frame, resizeMode,
        'Antialiasing',false);
    [bboxes, scores] = detectPeopleACF(frame,
        option.ROI, ...
        'Model','caltech',...
        'WindowStride', 2,...
        'NumScaleLevels', 4, ...
        'SelectStrongest', false);
    height = bboxes(:, 4) / resizeMode;
    y = (bboxes(:,2)-1) / resizeMode + 1;
    yfoot = min(length(obj.pedScaleTable),
        round(y + height));
    estHeight = obj.pedScaleTable(yfoot);
    invalid = abs(estHeight-
        height)>estHeight*option.scThresh;
    bboxes(invalid, :) = [];
    scores(invalid, :) = [];
    [bboxes, scores] =
        selectStrongestBbox(bboxes, scores, ...
        'RatioType', 'Min', 'OverlapThreshold', 0.6);
    if isempty(bboxes)
        centroids = [];
    else
        centroids = [(bboxes(:, 1) + bboxes(:, 3) / 2),
            ...
            (bboxes(:, 2) + bboxes(:, 4) / 2)];
    end
end
function predictNewLocationsOfTracks()
    for i = 1:length(tracks)

```

```

    Id = load(scaleDataFile, 'pedScaleTable');
    obj.pedScaleTable = Id.pedScaleTable;
    end
    function tracks = initializeTracks()
        tracks = struct(...

```

```

        'predPosition', {});
    end
    function frame = readFrame()
        frame = step(obj.reader);
    end
    function [centroids, bboxes, scores] =
        detectPeople()

```

```

        bbox = tracks(i).bboxes(end, :)
        predictedCentroid =
            predict(tracks(i).kalmanFilter);
        tracks(i).predPosition = [predictedCentroid -
            bbox(3:4)/2, bbox(3:4)];
    end
end
function [assignments, unassignedTracks,
    unassignedDetections] = ...
    detectionToTrackAssignment()
    predBboxes =
        reshape([tracks(:).predPosition], 4, []);
    cost = 1 - bboxOverlapRatio(predBboxes,
        bboxes);
    cost(cost > option.gatingThresh) = 1 +
        option.gatingCost;
    [assignments, unassignedTracks,
        unassignedDetections] = ...
        assignDetectionsToTracks(cost,
            option.costOfNonAssignment);
    end
    function updateAssignedTracks()
        numAssignedTracks = size(assignments, 1);
        for i = 1:numAssignedTracks
            trackIdx = assignments(i, 1);

            detectionIdx = assignments(i, 2);

            centroid = centroids(detectionIdx, :);

```

```

        bbox = bboxes(detectionIdx, :);
        correct(tracks(trackIdx).kalmanFilter,
                centroid);
    T = min(size(tracks(trackIdx).bboxes,1), 4);
    w = mean([tracks(trackIdx).bboxes(end-
        T+1:end, 3); bbox(3)]);
    h = mean([tracks(trackIdx).bboxes(end-
        T+1:end, 4); bbox(4)]);
    tracks(trackIdx).bboxes(end+1, :) = [centroid
        - [w, h]/2, w, h];
    tracks(trackIdx).age = tracks(trackIdx).age +
        1;

```

```

        idx = unassignedTracks(i);
        tracks(idx).age = tracks(idx).age + 1;
        tracks(idx).bboxes = [tracks(idx).bboxes;
            tracks(idx).predPosition];
        tracks(idx).scores = [tracks(idx).scores; 0];
        T = min(option.timeWindowSize,
            length(tracks(idx).scores));
        score = tracks(idx).scores(end-T+1:end);
        tracks(idx).confidence = [max(score),
            mean(score)];
            end
            end
        function deleteLostTracks()
            if isempty(tracks)
                return;
            end
            end
        function createNewTracks()
            unassignedCentroids =
            centroids(unassignedDetections, :);
            unassignedBboxes =
            bboxes(unassignedDetections, :);
            unassignedScores =
            scores(unassignedDetections);
            for i = 1:size(unassignedBboxes, 1)
                centroid = unassignedCentroids(i,:);
                bbox = unassignedBboxes(i, :);
                score = unassignedScores(i);
                kalmanFilter =
                configureKalmanFilter('ConstantVelocity', ...
                    centroid, [2, 1], [5, 5], 100);

```

```

        tracks(trackIdx).scores =
        [tracks(trackIdx).scores; scores(detectionIdx)];
        tracks(trackIdx).totalVisibleCount = ...
        tracks(trackIdx).totalVisibleCount + 1;
        T = min(option.timeWindowSize,
            length(tracks(trackIdx).scores));
        score = tracks(trackIdx).scores(end-T+1:end);
        tracks(trackIdx).confidence = [max(score),
            mean(score)];
            end
            end
        function updateUnassignedTracks()
            for i = 1:length(unassignedTracks)

```

```

                newTrack = struct(...
                    'id', nextId, ...
                    'color', 255*rand(1,3), ...
                    'bboxes', bbox, ...
                    'scores', score, ...
                    'kalmanFilter', kalmanFilter, ...
                    'age', 1, ...
                    'totalVisibleCount', 1, ...
                    'confidence', [score, score], ...
                    'predPosition', bbox);
                tracks(end + 1) = newTrack;
                %#ok<AGROW>
                nextId = nextId + 1;
            end
            end
        function displayTrackingResults()
            displayRatio = 4/3;
            frame = imresize(frame, displayRatio);
            if ~isempty(tracks),
                ages = [tracks(:).age]';
                confidence = reshape([tracks(:).confidence],
                    2, []);
                maxConfidence = confidence(:, 1);
                avgConfidence = confidence(:, 2);
                opacity = min(0.5, max(0.1, avgConfidence/3));
                noDispInds = (ages < option.ageThresh &
                    maxConfidence < option.confidenceThresh) |
                    ...
                    (ages < option.ageThresh / 2);
                for i = 1:length(tracks)
                    if ~noDispInds(i)
                        bb = tracks(i).bboxes(end, :);
                        bb(:,1:2) = (bb(:,1:2)-1)*displayRatio + 1;
                        bb(:,3:4) = bb(:,3:4) * displayRatio;
                        frame = insertShape(frame, ...

```

```

'FilledRectangle', bb, ...
'Color', tracks(i).color, ...
'Opacity', opacity(i));
frame = insertObjectAnnotation(frame, ...
    'rectangle', bb, ...
    num2str(avgConfidence(i)), ...
    'Color', tracks(i).color);
end
end

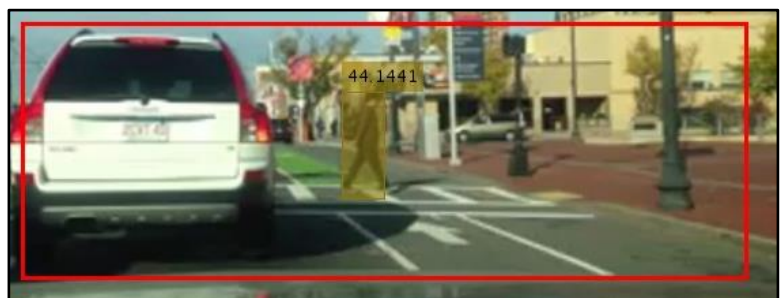
```

```

end
frame = insertShape(frame, 'Rectangle',
    option.ROI * displayRatio, ...
    'Color', [255, 0, 0], 'LineWidth', 3);
step(obj.videoPlayer, frame);
end
displayEndOfDemoMessage(mfilename)
end

```

### Στιγμιότυπα από το Video.



### **Motion Based Multiple Object Tracking**

#### Επεξήγηση Παραδείγματος.

Αυτό το παράδειγμα δείχνει πώς μπορούμε να εκτελέσουμε αυτόματη ανίχνευση και παρακολούθηση κίνησης των κινούμενων αντικειμένων σε ένα βίντεο από μια στατική κάμερα. Η ανίχνευση κινούμενων αντικειμένων και η παρακολούθηση βάσει κίνησης αποτελούν σημαντικά συστατικά πολλών εφαρμογών ηλεκτρονικής όρασης, όπως η αναγνώριση δραστηριότητας, η παρακολούθηση της κυκλοφορίας και η ασφάλεια αυτοκινήτου. Το πρόβλημα της παρακολούθησης αντικειμένων με κίνηση μπορεί να χωριστεί σε δύο μέρη:

- ❖ Ανίχνευση κινούμενων αντικειμένων σε κάθε πλαίσιο.
- ❖ Συνδυασμός των ανιχνεύσεων που αντιστοιχούν στο ίδιο αντικείμενο με την πάροδο του χρόνου.

Η ανίχνευση κινούμενων αντικειμένων χρησιμοποιεί έναν αλγόριθμο αφαίρεσης υποβάθρου βασισμένο σε μοντέλα με μείγμα Gaussian. Οι μορφολογικές λειτουργίες εφαρμόζονται στην έμπροσθεν μάσκα προσκηνίου για την εξάλειψη του θορύβου. Τέλος, η ανάλυση blob ανιχνεύει ομάδες συνδεδεμένων pixel, οι οποίες είναι πιθανό να αντιστοιχούν σε κινούμενα αντικείμενα. Η συσχέτιση των ανιχνεύσεων με το ίδιο αντικείμενο βασίζεται αποκλειστικά στην κίνηση. Η κίνηση κάθε κομματιού υπολογίζεται από ένα φίλτρο Kalman. Το φίλτρο χρησιμοποιείται για την πρόβλεψη της θέσης του κομματιού σε κάθε καρέ και για τον προσδιορισμό της πιθανότητας κάθε ανίχνευσης να αντιστοιχεί σε κάθε κομμάτι. Η συντήρηση της τροχιάς γίνεται μια σημαντική πτυχή αυτού του παραδείγματος. Σε οποιοδήποτε δεδομένο πλαίσιο, ορισμένες ανιχνεύσεις μπορούν να αντιστοιχιστούν σε κομμάτια, ενώ άλλες ανιχνεύσεις και ίχνη μπορεί να παραμείνουν ανενεργές. Τα επιλεγμένα κομμάτια ενημερώνονται χρησιμοποιώντας τις αντίστοιχες ανιχνεύσεις. Τα κομμάτια που δεν έχουν επισημανθεί είναι αόρατα. Μια ανίχνευση ξεκινά ένα νέο κομμάτι. Κάθε κομμάτι συνεχίζει να μετράει τον αριθμό των διαδοχικών καρέ, όπου παρέμεινε αδιάθετος. Εάν ο αριθμός υπερβαίνει ένα καθορισμένο όριο, το παράδειγμα υποθέτει ότι το αντικείμενο έφυγε από το πεδίο προβολής και διαγράφει το κομμάτι.

Κώδικας που χρησιμοποιήθηκε:

```
function multiObjectTracking()
    obj = setupSystemObjects();
    tracks = initializeTracks(); % Create an empty
        array of tracks.
    nextId = 1; % ID of the next track
    while ~isDone(obj.reader)
        frame = readFrame();
        [centroids, bboxes, mask] =
            detectObjects(frame);
        predictNewLocationsOfTracks();
        [assignments, unassignedTracks,
            unassignedDetections] = ...
            detectionToTrackAssignment();
        updateAssignedTracks();
        updateUnassignedTracks();
        deleteLostTracks();
        createNewTracks();
        displayTrackingResults();
    end
    function obj = setupSystemObjects()
        obj.reader =
            vision.VideoFileReader('atrium.mp4');
        obj.videoPlayer =
            vision.VideoPlayer('Position', [20, 400, 700,
                400]);
```

```
'consecutiveInvisibleCount', {});
    end
    function frame = readFrame()
        frame = obj.reader.step();
    end
```

```
        obj.maskPlayer =
            vision.VideoPlayer('Position', [740, 400, 700,
                400]);
        obj.detector =
            vision.ForegroundDetector('NumGaussians',
                3, ...
                'NumTrainingFrames', 40,
                'MinimumBackgroundRatio', 0.7);
        obj.blobAnalyser =
            vision.BlobAnalysis('BoundingBoxOutputPort'
                , true, ...
                'AreaOutputPort', true, 'CentroidOutputPort',
                true, ...
                'MinimumBlobArea', 400);
    end
    function tracks = initializeTracks()
        tracks = struct(...
            'id', {}, ...
            'bbox', {}, ...
            'kalmanFilter', {}, ...
            'age', {}, ...
            'totalVisibleCount', {}, ..
```

```
function [centroids, bboxes, mask] =
    detectObjects(frame)
    mask = obj.detector.step(frame);
    mask = imopen(mask, strel('rectangle', [3,3]));
    mask = imclose(mask, strel('rectangle', [15,
        15]));
```



```

mask = imfill(mask, 'holes');
[~, centroids, bboxes] =
obj.blobAnalyser.step(mask);
end
function predictNewLocationsOfTracks()
for i = 1:length(tracks)
bbox = tracks(i).bbox;
predictedCentroid =
predict(tracks(i).kalmanFilter);
predictedCentroid =
int32(predictedCentroid) - bbox(3:4) / 2;
tracks(i).bbox = [predictedCentroid,
bbox(3:4)];
end
end
function [assignments, unassignedTracks,
unassignedDetections] = ...
detectionToTrackAssignment()
nTracks = length(tracks);
nDetections = size(centroids, 1);
cost = zeros(nTracks, nDetections);
for i = 1:nTracks
cost(i, :) = distance(tracks(i).kalmanFilter,
centroids);
end
costOfNonAssignment = 20;
[assignments, unassignedTracks,
unassignedDetections] = ...
assignDetectionsToTracks(cost,
costOfNonAssignment);

end
invisibleForTooLong = 20;
ageThreshold = 8;
ages = [tracks(:).age];
totalVisibleCounts =
[tracks(:).totalVisibleCount];
visibility = totalVisibleCounts ./ ages;
lostInds = (ages < ageThreshold & visibility
< 0.6) | ...
[tracks(:).consecutiveInvisibleCount] >=
invisibleForTooLong;
tracks = tracks(~lostInds);
end

newTrack = struct(...
'id', nextId, ...
'bbox', bbox, ...
'kalmanFilter', kalmanFilter, ...
'age', 1, ...

```

```

end
function updateAssignedTracks()
numAssignedTracks = size(assignments, 1);
for i = 1:numAssignedTracks
trackIdx = assignments(i, 1);
detectionIdx = assignments(i, 2);
centroid = centroids(detectionIdx, :);
bbox = bboxes(detectionIdx, :);
correct(tracks(trackIdx).kalmanFilter,
centroid);
tracks(trackIdx).bbox = bbox;
tracks(trackIdx).age = tracks(trackIdx).age +
1;
tracks(trackIdx).totalVisibleCount = ...
tracks(trackIdx).totalVisibleCount + 1;
tracks(trackIdx).consecutiveInvisibleCount =
0;
end
end
function updateUnassignedTracks()
for i = 1:length(unassignedTracks)
ind = unassignedTracks(i);
tracks(ind).age = tracks(ind).age + 1;
tracks(ind).consecutiveInvisibleCount = ...
tracks(ind).consecutiveInvisibleCount + 1;
end
end
function deleteLostTracks()
if isempty(tracks)
return;

function createNewTracks()
centroids =
centroids(unassignedDetections,:);
bboxes = bboxes(unassignedDetections, :);
for i = 1:size(centroids, 1)
centroid = centroids(i,:);
bbox = bboxes(i, :);
kalmanFilter =

configureKalmanFilter('ConstantVelocity', ...

centroid, [200, 50], [100, 25], 100);

'totalVisibleCount', 1, ...
'consecutiveInvisibleCount', 0);
tracks(end + 1) = newTrack;
nextId = nextId + 1;
end

```

```

end
function displayTrackingResults()
    frame = im2uint8(frame);
    mask = uint8(repmat(mask, [1, 1, 3])) .* 255;
    minVisibleCount = 8;
    if ~isempty(tracks)
        reliableTrackInds = ...
            [tracks(:).totalVisibleCount] >
                minVisibleCount;
        reliableTracks = tracks(reliableTrackInds);
        if ~isempty(reliableTracks)
            bboxes = cat(1, reliableTracks.bbox);
            ids = int32([reliableTracks(:).id]);
            labels = cellstr(int2str(ids));
            predictedTrackInds = ...
                [reliableTracks(:).consecutiveInvisibleCount]
                    > 0;

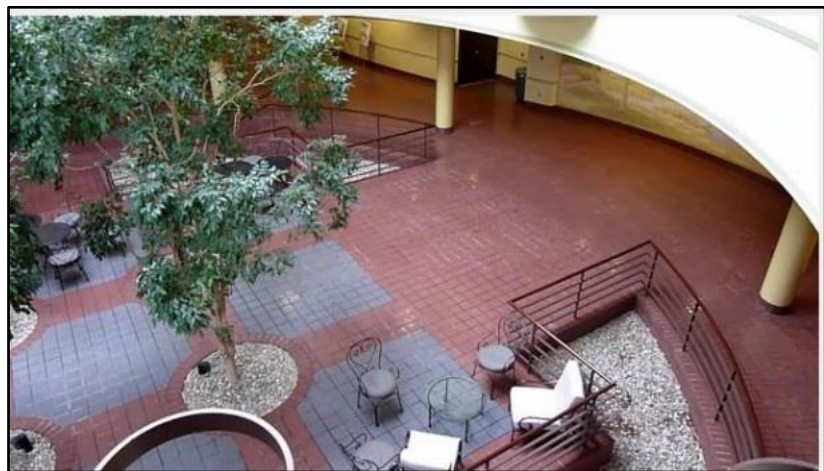
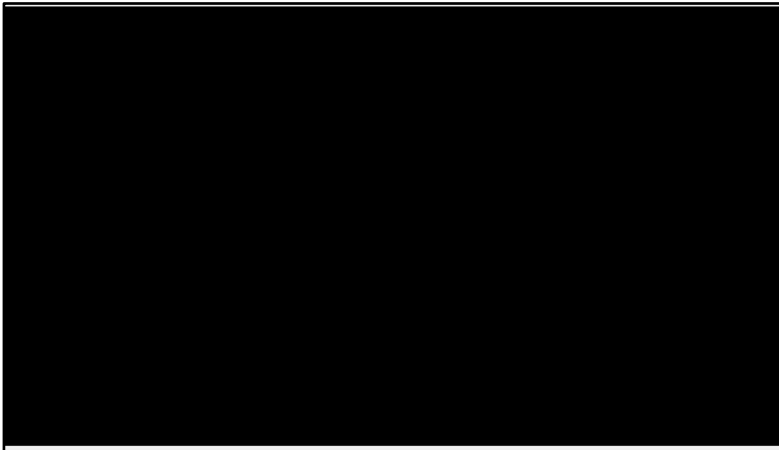
```

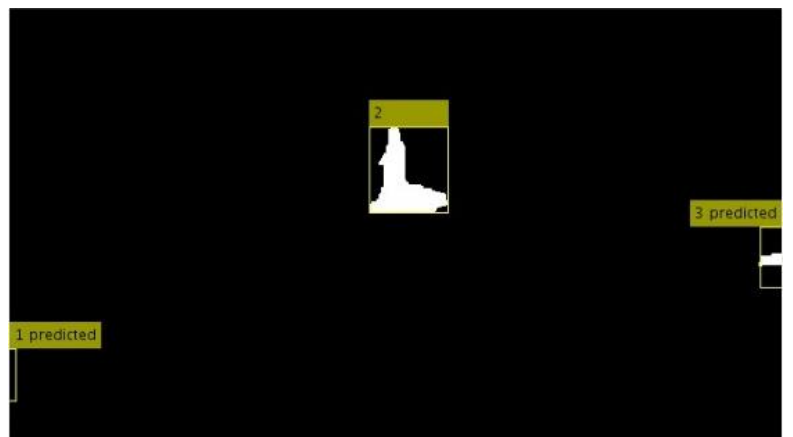
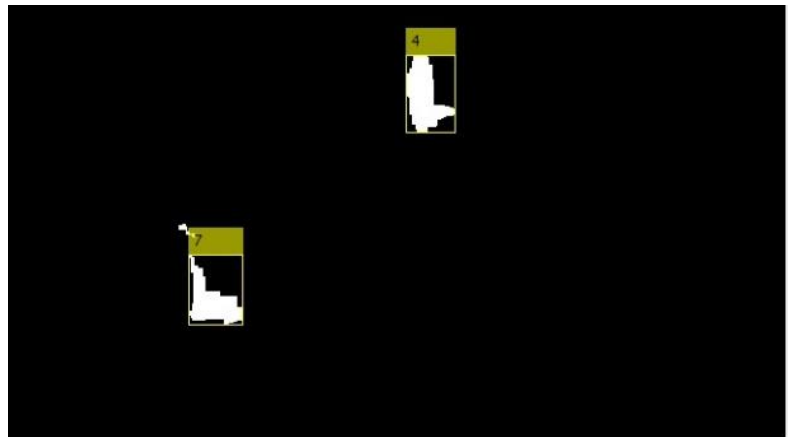
```

isPredicted = cell(size(labels));
isPredicted(predictedTrackInds) = {'
    predicted'};
labels = strcat(labels, isPredicted);
frame = insertObjectAnnotation(frame,
    'rectangle', ...
        bboxes, labels);
mask = insertObjectAnnotation(mask,
    'rectangle', ...
        bboxes, labels);
end
end
obj.maskPlayer.step(mask);
obj.videoPlayer.step(frame);
end
displayEndOfDemoMessage(mfilename)
end

```

Στιγμιότυπα από το Video.



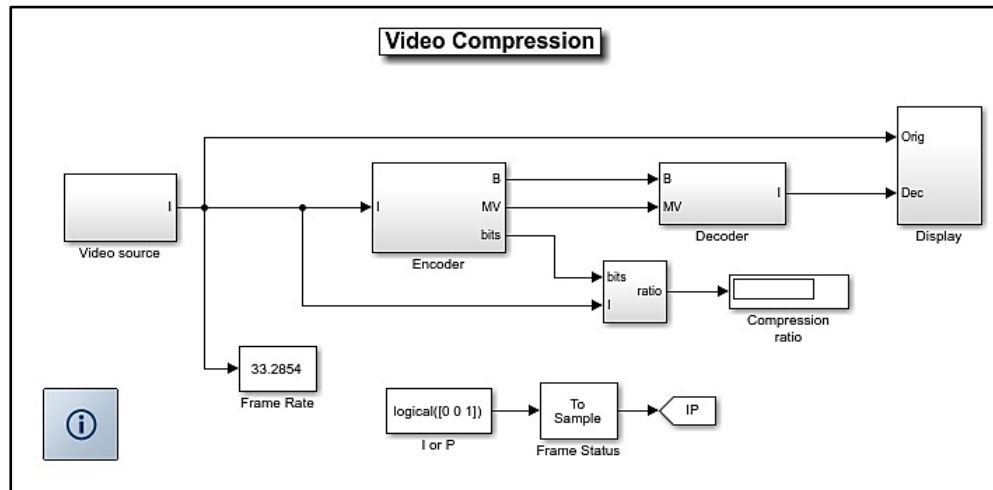


**Video Compression – Συμπίεση Video.**Επεξήγηση Παραδείγματος.

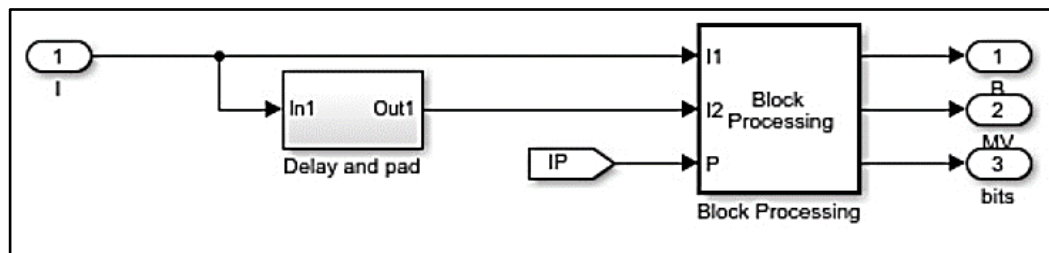
Αυτό το παράδειγμα δείχνει τον τρόπο συμπίεσης ενός video, χρησιμοποιώντας τεχνικές αντιστάθμισης κίνησης και τεχνικές διακριτού μετασχηματισμού συνημιτόνου (DCT). Αρχικά, υπολογίζει τους φορείς κίνησης μεταξύ διαδοχικών πλαισίων και τις χρησιμοποιεί για να μειώσει τις περιττές πληροφορίες. Έπειτα, διαιρεί κάθε πλαίσιο σε υποπλάγια και εφαρμόζει το διακριτό μετασχηματισμό συνημιτόνου σε κάθε υποτάξη. Τέλος, εφαρμόζει μια τεχνική ποσοτικοποίησης για την επίτευξη περαιτέρω συμπίεσης. Το υποσύστημα αποκωδικοποιητή εκτελεί την αντίστροφη διαδικασία για να ανακτήσει το αρχικό video.

Παράδειγμα Προσομοίωσης Video.

Η παρακάτω αναπαράσταση, παρουσιάζει το μοντέλο συμπίεσης του video, στη γενική του μορφή.

Υποσύστημα Κωδικοποιητή.

Το μπλοκ επεξεργασίας στέλνει 16-με-16 δευτερεύοντα αρχεία κάθε πλαισίου βίντεο στο υποσύστημα του μπλοκ επεξεργασίας μπλοκ για επεξεργασία. Μέσα σε αυτό το υποσύστημα, το μοντέλο εφαρμόζει μια τεχνική αντιστάθμισης κίνησης και το DCT στη ροή βίντεο. Απορρίπτοντας πολλούς συντελεστές υψηλής συχνότητας στην έξοδο DCT, στο παράδειγμα μειώνεται ο ρυθμός μετάδοσης bit του βίντεο εισόδου.

**Video Stabilization.**Επεξήγηση Video.

Αυτό το παράδειγμα δείχνει πώς αφαιρείται το αποτέλεσμα της κίνησης της κάμερας από μια ροή video. Σε πρώτο στάδιο, καθορίζουμε τον στόχο για τον εντοπισμό. Στην περίπτωση αυτή, είναι το πίσω μέρος ενός αυτοκινήτου και της πινακίδας. Δημιουργούμε επίσης μια δυναμική περιοχή αναζήτησης, η θέση της οποίας καθορίζεται από την τελευταία γνωστή τοποθεσία προορισμού. Στη συνέχεια, αναζητούμε τον στόχο μόνο σε αυτήν την περιοχή αναζήτησης, πράγμα που μειώνει τον αριθμό των υπολογισμών που απαιτούνται για την εύρεση του στόχου. Σε κάθε επόμενο πλαίσιο video, καθορίζουμε πόσο έχει μετακινηθεί ο στόχος σε σχέση με το προηγούμενο πλαίσιο. Χρησιμοποιούμε αυτές τις πληροφορίες για να αφαιρέσουμε τις ανεπιθύμητες κινήσεις της κάμερας μετάφρασης και να δημιουργήσουμε ένα σταθεροποιημένο video.

Κώδικας που χρησιμοποιήθηκε:

```
filename = 'shaky_car.avi';
hVideoSource = vision.VideoFileReader(filename,
    ...
    'ImageColorSpace', 'Intensity',...
    'VideoOutputDataType', 'double');
hTM = vision.TemplateMatcher('ROIInputPort',
    true, ...
    'BestMatchNeighborhoodOutputPort', true);
hVideoOut = vision.VideoPlayer('Name', 'Video
    Stabilization');
hVideoOut.Position(1) =
    round(0.4*hVideoOut.Position(1));
hVideoOut.Position(2) =
    round(1.5*(hVideoOut.Position(2)));
hVideoOut.Position(3:4) = [650 350];
pos.template_orig = [109 100];
pos.template_size = [22 18];
pos.search_border = [15 10];
pos.template_center = floor((pos.template_size-
    1)/2);
```

```
Target = zeros(18,22);
firstTime = true;
while ~isDone(hVideoSource)
    input = step(hVideoSource);
    if firstTime
        Idx = int32(pos.template_center_pos);
        MotionVector = [0 0];
        firstTime = false;
    else
```

```
Stabilized = imtranslate(input, Offset, 'linear');
Target = Stabilized(TargetRowIndices,
    TargetColIndices);
Stabilized(:, BorderCols) = 0;
Stabilized(BorderRows, :) = 0;
TargetRect = [pos.template_orig-Offset,
    pos.template_size];
```

```
txt = sprintf('%+05.1f,%+05.1f', Offset);
input = insertText(input(:,1),[191
    215],txt,'FontSize',16, ...
    'TextColor', 'white', 'BoxOpacity',
    0);
```

```
pos.template_center_pos = (pos.template_orig +
    pos.template_center - 1);
fileInfo = info(hVideoSource);
W = fileInfo.VideoSize(1);
H = fileInfo.VideoSize(2);
BorderCols = [1:pos.search_border(1)+4 W-
    pos.search_border(1)+4:W];
BorderRows = [1:pos.search_border(2)+4 H-
    pos.search_border(2)+4:H];
sz = fileInfo.VideoSize;
TargetRowIndices = ...pos.template_orig(2)-
    1:pos.template_orig(2)+pos.template_size(2)-2;
TargetColIndices = ...pos.template_orig(1)-
    1:pos.template_orig(1)+pos.template_size(1)-2;
SearchRegion = pos.template_orig -
    pos.search_border - 1;
Offset = [0 0];
```

```
IdxPrev = Idx;
ROI = [SearchRegion,
    pos.template_size+2*pos.search_border];
Idx = step(hTM, input, Target, ROI);
MotionVector = double(Idx-IdxPrev);
end
[Offset, SearchRegion] = updatesearch(sz,
    MotionVector, ...
```

```
SearchRegionRect = [SearchRegion,
    pos.template_size + 2*pos.search_border];
input = insertShape(input, 'Rectangle',
    [TargetRect; SearchRegionRect],...
    'Color', 'white');
```

```
step(hVideoOut, [input(:,1) Stabilized]);
end
release(hVideoSource);
displayEndOfDemoMessage(mfilename)
```

Στιγμιότυπο από Video.



### **Video Stabilization Using Point Feature Matching.**

#### Επεξήγηση Παραδείγματος.

Αυτό το παράδειγμα δείχνει πώς να σταθεροποιήσετε ένα βίντεο που καταγράφηκε από μια πλατφόρμα ψυχραιμίας. Ένας τρόπος για να σταθεροποιήσετε ένα βίντεο είναι να παρακολουθείτε ένα σημαντικό χαρακτηριστικό στην εικόνα και να το χρησιμοποιείτε ως σημείο αγκύρωσης, ώστε να ακυρώσετε όλες τις διαταραχές σε σχέση με αυτό. Αυτή η διαδικασία, ωστόσο, πρέπει να ξεκινήσει με τη γνώση του πού ένα τέτοιο χαρακτηριστικό έγκειται στο πρώτο πλαίσιο βίντεο. Σε αυτό το παράδειγμα, διερευνάται μια μέθοδος σταθεροποίησης βίντεο που λειτουργεί χωρίς μια τέτοια εκ των προτέρων γνώση. Αντίθετα, αναζητά αυτόματα το "υπόβαθρο" σε μια ακολουθία βίντεο και χρησιμοποιεί την παρατηρούμενη παραμόρφωση για να διορθώσει την κίνηση της κάμερας. Αυτός ο αλγόριθμος σταθεροποίησης περιλαμβάνει δύο βήματα. Καταρχήν, προσδιορίζουμε τους μετασχηματισμούς συνάφειας εικόνας μεταξύ όλων των γειτονικών πλαισίων μιας ακολουθίας βίντεο χρησιμοποιώντας τη λειτουργία `estimGeometricTransform` που εφαρμόζεται στις αντιστοιχίσεις σημείων μεταξύ δύο εικόνων. Δεύτερον, παραμορφώνουμε τα πλαίσια βίντεο για να επιτύχουμε ένα σταθεροποιημένο βίντεο. Αυτό το παράδειγμα είναι παρόμοιο με το Παράδειγμα Σταθεροποίησης Βίντεο. Η κύρια διαφορά είναι ότι στο παράδειγμα σταθεροποίησης βίντεο δίνεται μια περιοχή που θα παρακολουθεί, ενώ σε αυτό το παράδειγμα δεν δίνεται τέτοια γνώση. Και τα δύο παραδείγματα χρησιμοποιούν το ίδιο βίντεο.

#### Κώδικας που χρησιμοποιήθηκε:

```
filename = 'shaky_car.avi';
hVideoSrc = vision.VideoFileReader(filename,
    'ImageColorSpace', 'Intensity');
imgA = step(hVideoSrc);
```

```
figure;
imshowpair(imgA,imgB,'ColorChannels','red-
cyan');
title('Color composite (frame A = red, frame
B = cyan)');
ptThresh = 0.1;
pointsA = detectFASTFeatures(imgA,
    'MinContrast', ptThresh);
pointsB = detectFASTFeatures(imgB,
    'MinContrast', ptThresh);
figure; imshow(imgA); hold on;
plot(pointsA);
title('Corners in A');
```

```
imgB = step(hVideoSrc);
figure; imshowpair(imgA, imgB, 'montage');
title(['Frame A', repmat(' ', [1 70]), 'Frame B']);
```

```
figure; imshow(imgB); hold on;
plot(pointsB);
title('Corners in B');
[featuresA, pointsA] = extractFeatures(imgA,
    pointsA);
[featuresB, pointsB] = extractFeatures(imgB,
    pointsB);
indexPairs = matchFeatures(featuresA,
    featuresB);
pointsA = pointsA(indexPairs(:, 1), :);
pointsB = pointsB(indexPairs(:, 2), :);
```

```

figure; showMatchedFeatures(imgA, imgB,
    pointsA, pointsB);
    legend('A', 'B');
[tform, pointsBm, pointsAm] =
estimateGeometricTransform(...
    pointsB, pointsA, 'affine');
imgBp = imwarp(imgB, tform, 'OutputView',
    imref2d(size(imgB)));

    H = tform.T;
    R = H(1:2,1:2);
theta = mean([atan2(R(2),R(1)) atan2(-
    R(3),R(4))]);
    scale = mean(R([1 4])/cos(theta));
    translation = H(3, 1:2);
HsRt = [[scale*cos(theta) -sin(theta);
    sin(theta) cos(theta)]; ...
    translation], [0 0 1]'];
tformsRT = affine2d(HsRt);
imgBold = imwarp(imgB, tform,
'OutputView', imref2d(size(imgB)));
imgBsRt = imwarp(imgB, tformsRT,
'OutputView', imref2d(size(imgB)));
    figure(2), clf;
imshowpair(imgBold,imgBsRt,'ColorChannels'
    , 'red-cyan'), axis image;

```

```

H = cvxEstStabilizationTform(imgA,imgB);
HsRt = cvxTformToSRT(H);
Hcumulative = HsRt * Hcumulative;
    imgBp =
imwarp(imgB,affine2d(Hcumulative),'OutputV
    iew',imref2d(size(imgB)));

```

```

movMean = movMean/(ii-2);
    release(hVideoSrc);
    release(hVPlayer);
figure; imshowpair(movMean,
    correctedMean, 'montage');

```

```

pointsBmp = transformPointsForward(tform,
    pointsBm.Location);
    figure;
showMatchedFeatures(imgA, imgBp,
    pointsAm, pointsBmp);
    legend('A', 'B');

```

```

title('Color composite of affine and s-R-t
    transform outputs');
    reset(hVideoSrc);
hVPlayer = vision.VideoPlayer;
movMean = step(hVideoSrc);
    imgB = movMean;
    imgBp = imgB;
correctedMean = imgBp;
    ii = 2;
Hcumulative = eye(3);
while ~isDone(hVideoSrc) && ii < 10
    imgA = imgB; % z^-1
    imgAp = imgBp; % z^-1
    imgB = step(hVideoSrc);
    movMean = movMean + imgB;

```

```

    step(hVPlayer,
imfuse(imgAp,imgBp,'ColorChannels','red-
    cyan'));
correctedMean = correctedMean + imgBp;
    ii = ii+1;
    end
correctedMean = correctedMean/(ii-2);

```

```

title(['Raw input mean', repmat(' ',[1 50]),
    'Corrected sequence mean']);
displayEndOfDemoMessage(mfilename)

```

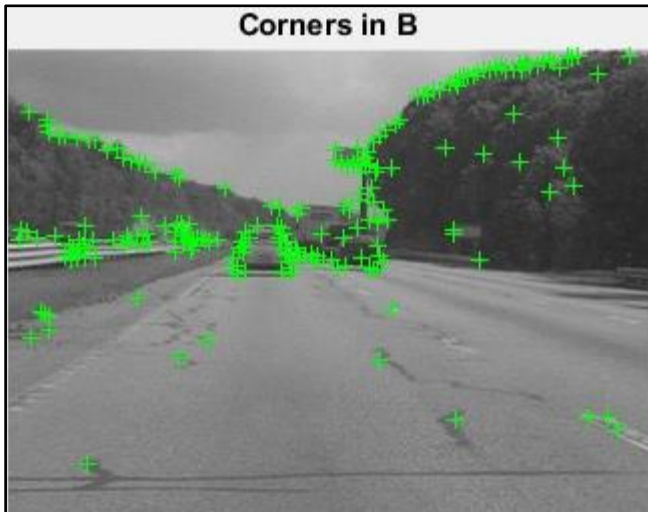
Στιγμιότυπα από το Video.



Color composite of affine and s-R-t transform outputs



Corners in B







**ΠΗΓΕΣ**

- ❖ Μελέτη Και Σύγκριση Μεθόδων Εκτίμησης Υποκειμενικής Ποιότητας Τηλεοπτικής Εικόνας- ΜΑΡΙΑ ΣΙΣΜΑΝΙΔΟΥ
- ❖ Ανάπτυξη και Συγκριτική Μελέτη Μεθόδων Αποτίμησης Ποιότητας Ψηφιακών Εικόνων- ΤΟΛΙΟΠΟΥΛΟΣ ΙΩΑΝΝΗΣ
- ❖ Εκτίμηση της Ποιότητας Εικόνων με τη Χρήση Τεχνητών Νευρωνικών Δικτύων- ΜΠΙΛΙΟΥΡΗ ΙΩΑΝΝΑ
- ❖ Subjective Video Quality Assessment of HTTP Adaptive Streaming Technologies-DAVID C. ROBINSON, YVES JUTRAS, AND VIOREL CRACIUN
- ❖ Μέθοδοι Εκτίμησης QoE Κατά τη Μετάδοση Πολυμέσων- ΚΩΝΣΤΑΝΤΙΝΟΣ ΦΛΟΥΡΗΣ
- ❖ Still Image and Video Compression with Matlab- K.S. THYAGARAJAN
- ❖ User Perceived Video Quality Modelling On Mobile Devices For Vp9 And H265 Encoders-YAO XIAO
- ❖ Ανάπτυξη Αλγορίθμων για Τμηματοποίηση (Segmentation) Ψηφιακής Εικόνας και Υλοποίησή τους σε Περιβάλλον Matlab - ΒΟΥΤΣΚΟΠΟΥΛΟΥ ΕΥΑΓΓΕΛΙΑ
- ❖ Μετάδοση Ψηφιακού Video στο Διαδίκτυο - ΜΑΚΡΗ ΣΤΑΜΑΤΙΚΗ
- ❖ Συμπύεση Video- ΔΗΜΗΤΡΗΣ ΚΟΚΚΑΣ
- ❖ Μετάδοση Υπηρεσιών Video σε Ασύρματα Δίκτυα –ΤΖΟΥΜΑΣ ΕΜΜΑΝΟΥΗΛ
- ❖ Βασικά Στοιχεία για τη Χρήση του MATLAB & Εφαρμογή σε Προβλήματα Κατασκευών- ΚΑΤΣΑΝΟΣ ΕΥΑΓΓΕΛΟΣ
- ❖ Οδηγός Matlab για Αρχάριους-ΧΡΙΣΤΟΣ ΞΕΝΟΦΩΝΤΟΣ
- ❖ Ανάλυση Fourier
- ❖ [https://en.wikipedia.org/wiki/Discrete cosine transform](https://en.wikipedia.org/wiki/Discrete_cosine_transform)
- ❖ <https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node231.html>
- ❖ <https://en.wikipedia.org/wiki/Video>
- ❖ [https://en.wikipedia.org/wiki/Live streaming](https://en.wikipedia.org/wiki/Live_streaming)