

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

## Τμήμα Πληροφορικής



### Συστήματα Πολυμέσων

Τίτλος εργασίας	Εργασία Συστήματα Πολυμέσων εαρινό εξάμηνο ακαδημαϊκού έτους 2021-20222
Μέλη ομάδας / Αριθμοί μητρώου	Νικήτας Χατζής Π19183
	Παναγιώτης Μουρλάς Π19108
	Αντώνης Δαρμής Π19040
	Βαχράμ Μανγκογιάν Π19092
Ημερομηνία παράδοσης	15/07/2022

# Περιεχόμενα

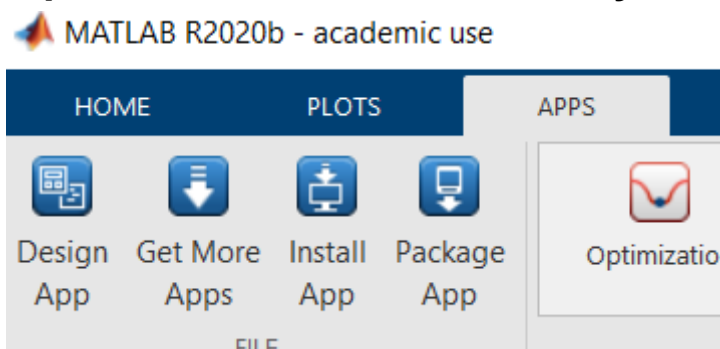
- Requirements
- Εκκίνηση
- Θέμα 1 i)
- Θέμα 1 ii)
- Θέμα 2
- Πηγές

# Requirements

Απαραίτητα για τη λειτουργία των προγραμμάτων είναι τα:

- MATLAB
- Image Processing Toolbox (MathWorks)
- Signal Processing Toolbox (MathWorks)
- Communications Toolbox (MathWorks)

Τα toolbox αυτά μπορείτε να τα κατεβάσετε από το **Add-On Explorer** του MATLAB, πατώντας **APPS** → **Get More Apps**



## **Εκκίνηση**

Κάντε unzip το αρχείο **polymesa\_p19183\_p19108\_p19040\_p19092.zip** που παραδόθηκε στο gunet2 σε κάποιο φάκελο. Στη συνέχεια μεταβείτε στο φάκελο αυτό και κάντε unzip (δεξί κλικ → Extract here) τα αρχεία **source2022.zip** και **auxiliary2022.zip** που ζητούνται από την εκφώνηση.

## Θέμα 1 i)

Τα αρχεία που αντιστοιχούν σε αυτό το θέμα είναι στον φάκελο “/Thema1a”. Το βίντεο που χρησιμοποιείται είναι το “/Videos/Untitled.mp4”.

**Εκφώνηση:** Έστω video της επιλογής σας διάρκειας 5 s – 15 s. Υποθέστε ότι το Frame 1 είναι πάντα I frame και ότι τα επόμενα πλαίσια είναι P frames. Κάθε πλαίσιο P προβλέπεται χωρίς αντιστάθμιση κίνησης από το προηγούμενο πλαίσιο. Υπολογίστε και απεικονίστε την ακολουθία εικόνων σφάλματος και κωδικοποιήστε την χωρίς απώλειες. Υλοποιήστε τον κωδικοποιητή/αποκωδικοποιητή.

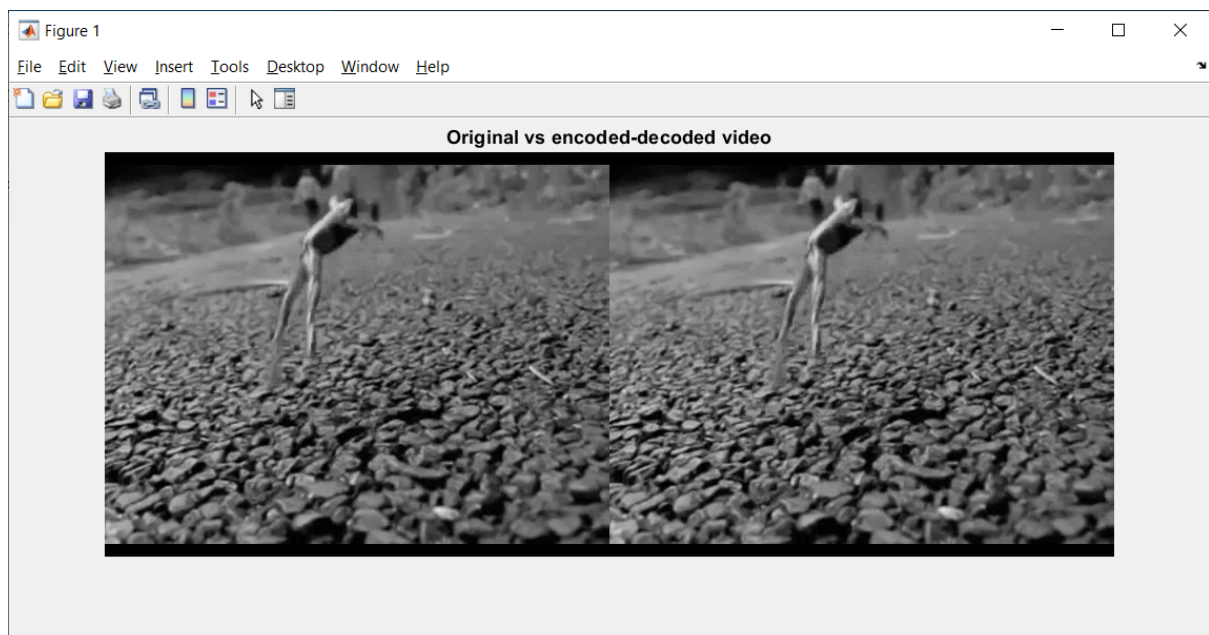
**Preprocessing:** Αρχικά, το πρόγραμμα διαβάζει το βίντεο ως ακολουθία πλαισίων και μετατρέπει τα πλαίσια αυτά από έγχρωμα σε grayscale, για να διευκολυνθεί η διαδικασία.

**Method:** Χωρίζουμε το βίντεο σε πλαίσια. Για κάθε πλαίσιο, προβλέπουμε πως είναι ίδιο με το προηγούμενο ( $x(n + 1) \equiv x(n)$ ) και υπολογίζουμε τη διαφορά τους. Ο κωδικοποιητής χωρίζει κάθε πλαίσιο διαφορών σε macroblocks και για κάθε macroblock κάνει run-length encoding και huffman σε κάθε επιμέρους 8x8 block. Ο decoder υλοποιεί την ανάποδη διαδικασία. Αποκωδικοποιεί τις διαφορές με huffman decoding και run-length decoding και στη συνέχεια ανακατασκευάζει κάθε πλαίσιο διαφορών ενώνοντας τα επιμέρους 8x8 block του σε macroblocks, τα οποία επίσης ενώνει μετά για να ανακατασκευάσει το πλαίσιο. Στο τέλος ξεκινώντας από το αρχικό πλαίσιο, προστίθενται οι διαφορές για να δημιουργηθούν τα επόμενα.

**Results:** Πριν γίνει το encoding, καθώς είναι ιδιαίτερα χρονοβόρο, το πρόγραμμα εμφανίζει τις εικόνες διαφορών. Επίσης, αφού τελειώσει το decoding (που είναι επίσης χρονοβόρο), το πρόγραμμα εμφανίζει τις ανακατασκευασμένες εικόνες διαφορών. Επιλέξαμε να μην εμφανίζεται μια decoded εικόνα διαφορών αμέσως μετά την αποκωδικοποίηση της (πριν αποκωδικοποιηθεί η επόμενη) διότι η καθυστέρηση που προκύπτει από την διαδικασία δημιουργεί μια καθυστέρηση μεταξύ της προβολής 2 ανακατασκευασμένων πλαισίων διαφορών. Συνεπώς, το πρόγραμμα αποκωδικοποιεί όλες τις εικόνες διαφορών και τις προβάλλει μετά. Τέλος, ξεκινώντας από το πρώτο frame, προσθέτει κάθε φορά τη διαφορά για να ανακατασκευάσει το επόμενο frame και το προβάλλει, ανακατασκευάζοντας έτσι το αρχικό βίντεο.



(Αρχική και ανακατασκευασμένη διαφορά δίπλα δίπλα)



(Αρχικό και ανακατασκευασμένο βίντεο δίπλα δίπλα)

**Conclusions:** Το ανακατασκευασμένο βίντεο είναι το ίδιο με το αρχικό, επειδή η κωδικοποίηση γίνεται χωρίς απώλειες.

**Αρχεία και εκτέλεση:** Το script του θέματος αυτού είναι το “**thema1a.m**” που βρίσκεται στον φάκελο “/Thema1a”. Ο φάκελος περιέχει επίσης βοηθητικές συναρτήσεις που χρησιμοποιούνται κατά την εκτέλεση. Τα αρχεία “**runlengthenc.m**” και “**runlengthdec.m**” υλοποιούν run-length encoding και decoding αντίστοιχα ενώ στα αρχεία “**codeFrame.m**” και “**decodeFrame.m**” υλοποιούν τη διαδικασία του encoding και του decoding που περιγράφηκε παραπάνω για ένα frame. Τα αρχεία “**encode.m**” και “**decode.m**” αποτελούν τον κωδικοποιητή και τον αποκωδικοποιητή αντίστοιχα και αξιοποιούν τις παραπάνω συναρτήσεις.

## **Θέμα 1 ii)**

**Τα αρχεία που αντιστοιχούν σε αυτό το θέμα βρίσκονται στο φάκελο “/Thema1b”. Το βίντεο που χρησιμοποιείται είναι το “/Videos/Untitled.mp4”.**

**Εκφώνηση:** Έστω video της επιλογής σας διάρκειας 5 s – 15 s. Υποθέστε ότι το Frame 1 είναι πάντα I frame και ότι τα επόμενα πλαίσια είναι P frames. Υλοποιήστε την τεχνική αντιστάθμισης κίνησης για την συμπίεση της ακολουθίας πλαισίων χρησιμοποιώντας αντιστάθμιση κίνησης σε macroblocks 32x32, ακτίνα αναζήτησης  $k=16$  και τεχνική σύγκρισης macroblocks της επιλογής σας. Να επιταχυνθεί η διαδικασία υλοποιώντας ιεραρχική αναζήτηση. Υπολογίστε, αποθηκεύστε και απεικονίστε την ακολουθία εικόνων πρόβλεψης και εικόνων σφαλμάτων. Υλοποιήστε τον κωδικοποιητή/αποκωδικοποιητή

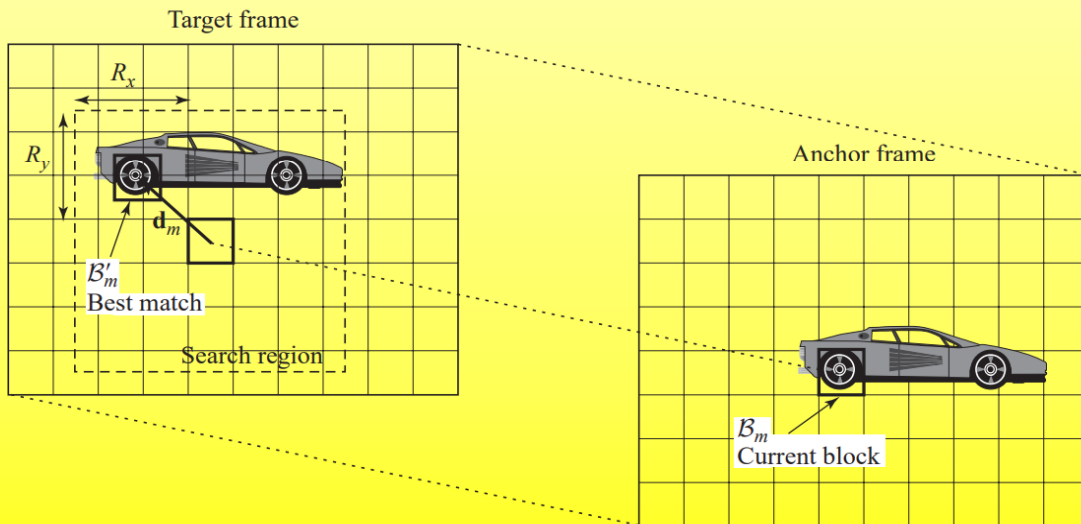
**Preprocessing:** Αρχικά, το πρόγραμμα διαβάζει το βίντεο ως ακολουθία πλαισίων και μετατρέπει τα πλαίσια αυτά από έγχρωμα σε grayscale, για να διευκολυνθεί η διαδικασία.

**Method:** Πρώτα γίνεται η διαδικασία της κωδικοποίησης (encoding). Ο encoder είναι βασισμένος στον ψευδοκώδικα που μας παρέχει το βιβλίο του μαθήματος στη σελίδα 265. Για το πρώτο frame γίνεται κωδικοποίηση ενώ για όλα τα επόμενα προηγείται η αντιστάθμιση κίνησης.

Υλοποιήσαμε την τεχνική της αντιστάθμισης κίνησης με εξαντλητική αναζήτηση, που αναζητά το καλύτερο ταίριασμα σε όλα τα block γύρω από ένα macroblock με κάποια ακτίνα αναζήτησης  $k$ . Χρησιμοποιήσαμε  $k=16$ , όπως επισημαίνεται στην εκφώνηση.



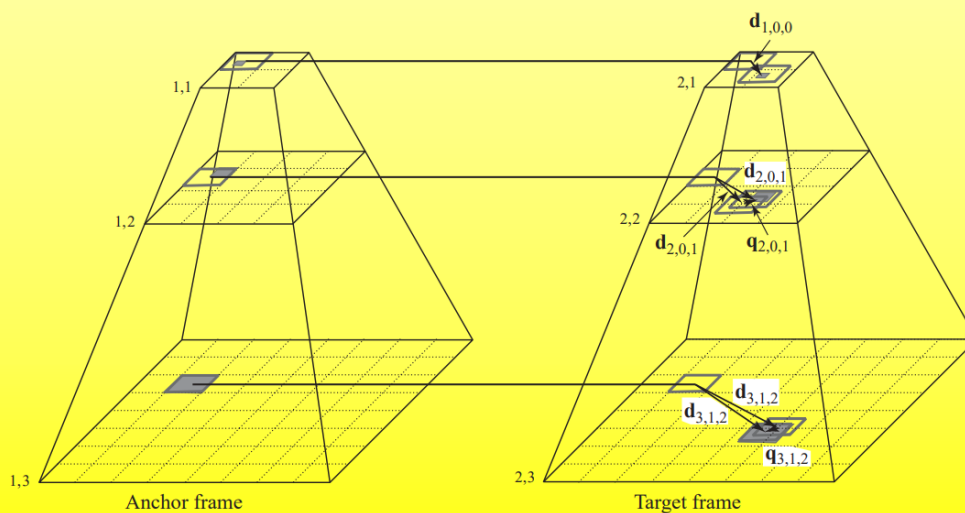
# Exhaustive Block Matching Algorithm (EBMA)

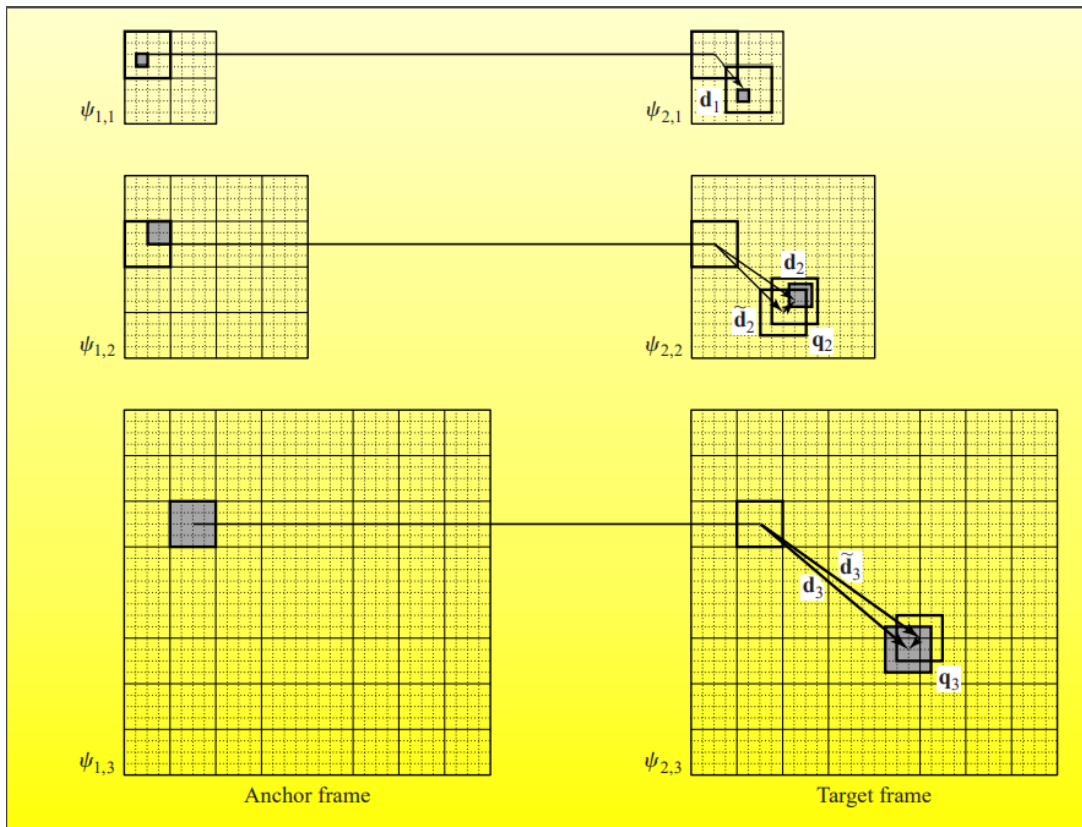


(Πηγή: Διαφάνειες Yao Wang [9])

Στη συνέχεια υλοποιήσαμε την τεχνική της ιεραρχικής αναζήτησης, που μειώνει σημαντικά τον αριθμό των Mean Absolute Difference (MAD) πράξεων, άρα και το χρόνο εκτέλεσης. Συγκεκριμένα χρησιμοποιήσαμε 4 επίπεδα στο HBMA μας.

# Hierarchical Block Matching Algorithm (HBMA)





(Πηγή: Διαφάνειες Yao Wang [9])

Μετά όμως η κωδικοποίηση των πλαισίων (codeFrame στον ψευδοκώδικα και στα αρχεία μας) δεν ακολουθεί αυτούσια την διαδικασία του JPEG που αναφέρεται στον ψευδοκώδικα, αλλά μια απλούστερη, τροποποιημένη έκδοση του. Τα βήματα που ακολουθεί είναι τα εξής:

1. Εφαρμογή Discrete Cosine Transform (DCT) σε κάθε 8x8 block κάθε επιμέρους macroblock
2. Εφαρμογή μάσκας στους πάνω αριστερά συντελεστές σε κάθε 8x8 block κάθε επιμέρους macroblock

1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

(Πηγή: MathWorks [\[14\]](#))

3. Κβάντιση των DCT συντελεστών σε κάθε 8x8 block κάθε επιμέρους macroblock με χρήση του 8x8 πίνακα κβάντισης του JPEG

$$F_Q(u, v) = \text{ceil}(F(u, v)/Q(u, v)) \text{ όπου } Q(u, v) \text{ η τιμή στον πίνακα κβάντισης.}$$

(σελίδα 226 του βιβλίου του μαθήματος)

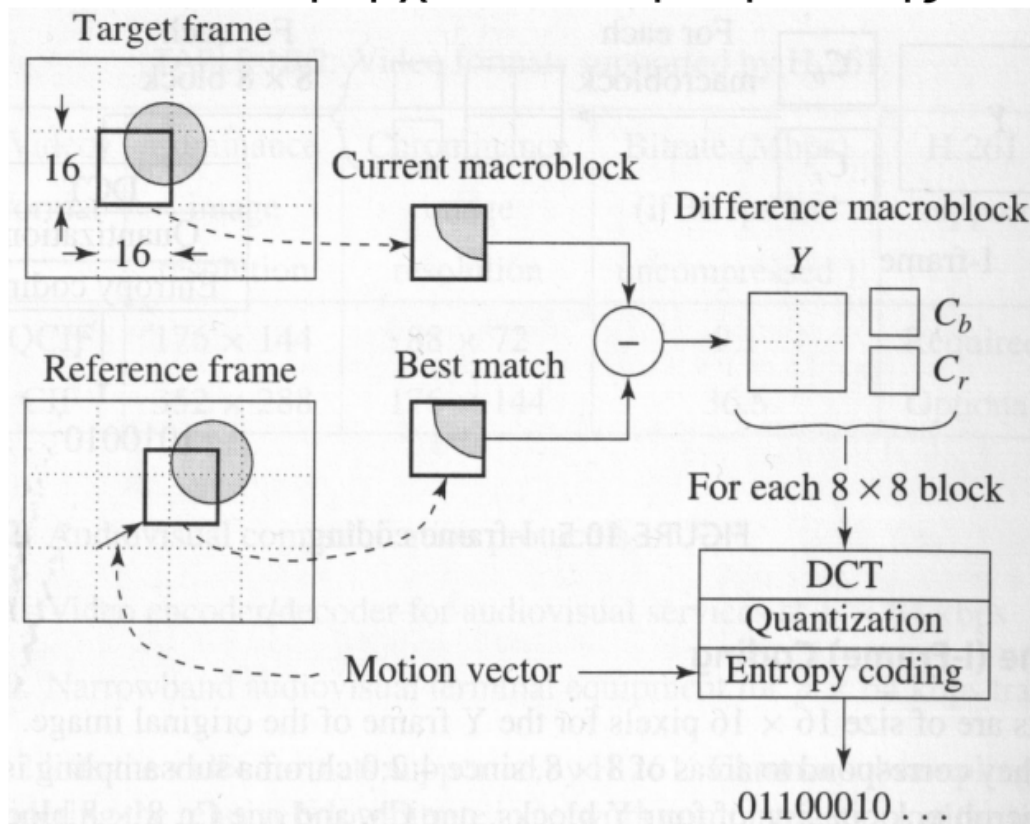
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(πίνακας κβάντισης JPEG - πηγή: [\[15\]](#))

4. Zig-zag σάρωση των κβαντισμένων συντελεστών σε κάθε 8x8 block κάθε επιμέρους macroblock
5. Run-length encoding σε κάθε επιμέρους ακολουθία που προέκυψε από το βήμα 4
6. Huffman κωδικοποίηση της εξόδου του βήματος 5

Για το πρώτο πλαίσιο του βίντεο, η παραπάνω διαδικασία χρησιμοποιείται μόνη της. Για τα υπόλοιπα πλαίσια, προηγείται η διαδικασία της αντιστάθμισης κίνησης, που επιχειρεί να προβλέψει το τρέχον πλαίσιο από το προηγούμενο. Για κάθε τέτοιο predicted frame, υπολογίζεται η διαφορά του από το πραγματικό πλαίσιο, η οποία στη συνέχεια κωδικοποιείται με τη διαδικασία που περιγράφηκε παραπάνω.

## Βασική αρχιτεκτονική συμπίεσης



(σχήμα συμπίεσης από τις διαφάνειες)

Τα διανύσματα κίνησης που επίσης προκύπτουν από την διαδικασία της αντιστάθμισης κίνησης, κωδικοποιούνται με τον εξής τρόπο:

1. Run-length encoding (RLE) στα  $dx$  και  $dy$  αντίστοιχα.

Έχει υλοποιηθεί δηλαδή μια διαδικασία παρόμοια με αυτή που περιγράφεται και στις διαφάνειες του μαθήματος. Μια διαφορά είναι πως τα διανύσματα κίνησης τα κωδικοποιούμε μόνο με RLE και όχι και με huffman για να μειωθεί η καθυστέρηση.

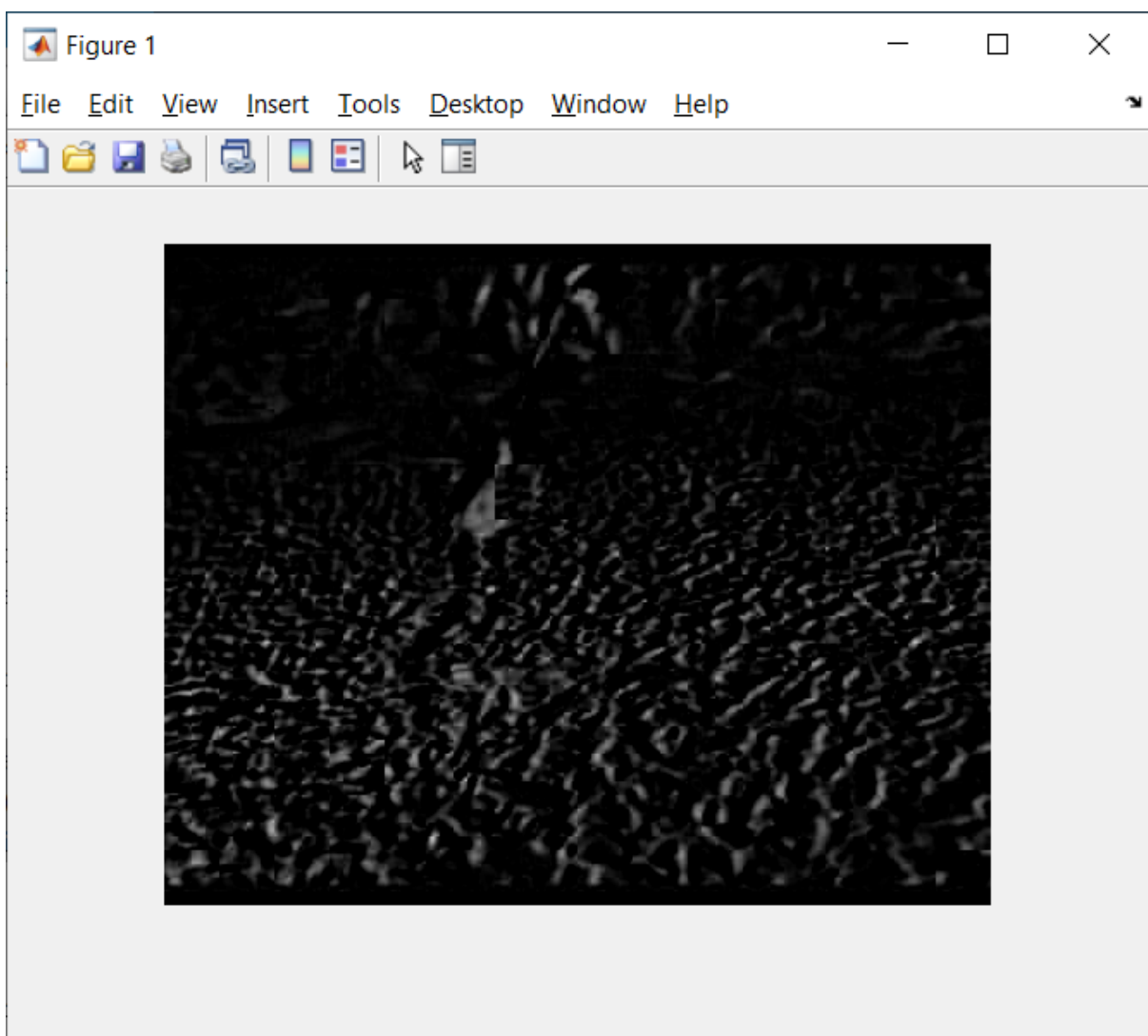
- Εκμετάλλευση του μεγάλου πλεονασμού δεδομένων σε μία ακολουθία εικόνων video:
  - Χρονικός: Πρόβλεψη με αντιστάθμιση κίνησης
  - Χωρικός: Εφαρμογή DCT μετασχηματισμού στο λάθος πρόβλεψης
  - Κβάντιση των συντελεστών DCT
  - Zigzag σάρωση των κβαντισμένων τελεστών
  - Κωδικοποίηση μήκους διαδρομής
  - Huffman Κωδικοποίηση της εξόδου του προηγούμενου βήματος και του διανύσματος κίνησης του ορθογώνιου τμήματος (macroblock)

(διαδικασία που περιγράφεται στις διαφάνειες)

Ο decoder έπειτα υλοποιεί την ανάποδη διαδικασία, βασισμένη στον ψευδοκώδικα στη σελίδα 266 του βιβλίου του μαθήματος. Ωστόσο, επειδή η διαδικασία δεν είναι αρκετά γρήγορη ώστε να μπορεί να γίνεται αποκωδικοποίηση και αμέσως να εμφανίζεται το αποκωδικοποιημένο πλαίσιο, τα αποκωδικοποιημένα πλαίσια αποθηκεύονται και εμφανίζονται ως βίντεο στο τέλος του προγράμματος.

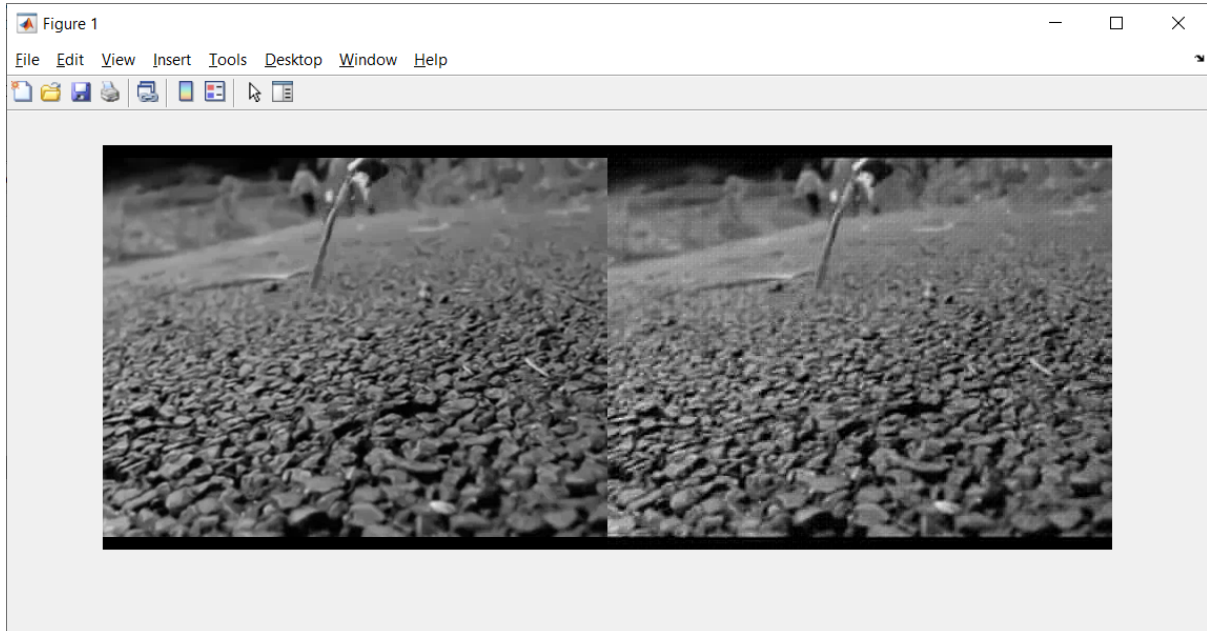
1. Αποκωδικοποίηση Huffman
2. Run-length decoding
3. Ανακατασκευή μπλοκ με την αντίστροφη διαδικασία του zigzag scan.
4. Αντίστροφη κβάντιση  
$$F'(u, v) = F_Q(u, v) * Q(u, v)$$
 όπου  $Q(u, v)$  η τιμή στον πίνακα κβάντισης.
5. Αντίστροφος DCT μετασχηματισμός

**Results:** Αφού ολοκληρωθεί η διαδικασία του encoding και του decoding, το πρόγραμμα εμφανίζει πρώτα τα πλαίσια διαφορών σε ένα παράθυρο



(πλαίσιο διαφορών)

και έπειτα μια δίπλα-δίπλα σύγκριση του αρχικού βίντεο με το ανακατασκευασμένο βίντεο



(original και ανακατασκευασμένο πλαίσιο δίπλα δίπλα)

Επίσης, τα πλαίσια διαφορών αποθηκεύονται στο αρχείο **“errorframes.avi”** και τα ανακατασκευασμένα-συμπιεσμένα πλαίσια αποθηκεύονται στο αρχείο **“compressedframes.avi”**.

**Conclusions:** Το αποκωδικοποιημένο βίντεο δεν είναι ίδιο με το αρχικό. Αυτό συμβαίνει επειδή η διαδικασία κωδικοποίησης σε αυτή τη περίπτωση είναι με απώλειες, αλλά και επειδή χρησιμοποιείται η διαδικασία της αντιστάθμισης κίνησης για την πρόβλεψη του επόμενου πλαισίου. Οι διαφορές φαίνονται καλύτερα εάν γίνει zoom in.

**Αρχεία και εκτέλεση:** Το script του θέματος αυτού είναι το **“thema1b.m”** που βρίσκεται στον φάκελο **“/Thema1b”**. Ο φάκελος αυτός περιέχει επίσης **πολλές** βοηθητικές συναρτήσεις που υλοποιούν κάθε επιμέρους λειτουργία. Περισσότερες λεπτομέρειες για τον τρόπο λειτουργίας τους βρίσκονται στα ίδια τα αρχεία. Συγκεκριμένα:

#### Αντιστάθμιση κίνησης

- **computeMotionVector.m** - Υπολογισμός διανυσμάτων κίνησης με εξαντλητική αναζήτηση
- **computeMotionVectorHS.m** - Υπολογισμός διανυσμάτων κίνησης με ιεραρχική αναζήτηση

- **compensateFrame.m** - Δημιουργία αντισταθμισμένου πλαισίου, δοθέντος τα διανύσματα κίνησης και το reference frame.
- **MAD.m** - Υπολογισμός της μετρικής MAD για 2 μπλοκ ίδιου μεγέθους.
- **min\_cost\_mb.m** - Υπολογισμός dx και dy του καλύτερου ταιριάσματος-macroblock δεδομένου ενός πίνακα με κόστη MAD που προέρχεται από εξαντλητική αναζήτηση
- **check8NewPositions.m** - Υπολογισμός MAD κόστους για τις 8 νέες θέσεις που εξετάζονται στα κατώτερα επίπεδα της ιεραρχικής αναζήτησης.
- **min\_cost\_mb\_2.m** - Υπολογισμός dx και dy του καλύτερου ταιριάσματος-macroblock δεδομένου ενός πίνακα με κόστη MAD που προέρχονται από ιεραρχική αναζήτηση

#### Κωδικοποίηση/Αποκωδικοποίηση

- **encode\_video.m** - Αποτελεί τον κωδικοποιητή. Βασίστηκε στον ψευδοκώδικα της σελίδας 265 του βιβλίου.
- **codeFrame.m** - Κωδικοποιεί ένα πλαίσιο με τη διαδικασία που περιγράφηκε νωρίτερα.
- **zigzag.m** - Ζιγκ-ζαγκ σάρωση ενός nxm πίνακα και μετατροπή του σε μονοδιάστατη ακολουθία.
- **runlengthenc.m** - Υλοποιεί την κωδικοποίηση μήκους διαδρομής όμοια με το θέμα 1 ι).
- **decode\_video** - Αποτελεί τον αποκωδικοποιητή. Βασίστηκε στον ψευδοκώδικα της σελίδας 266 του βιβλίου.
- **decodeFrame.m** - Υλοποιεί τη διαδικασία της αποκωδικοποίησης για ένα πλαίσιο.
- **izigzag.m** - Μετατροπή μονοδιάστατης ακολουθίας σε δισδιάστατο πίνακα με ανάποδη σάρωση zigzag.
- **runlengthdec.m** - Αποκωδικοποίηση μήκους διαδρομής, όμοια με το θέμα 1 ι).

## Θέμα 2

**Το αρχείο που αντιστοιχεί σε αυτό το θέμα είναι το `thema2.m`. Το βίντεο που χρησιμοποιείται είναι το `"/Videos/GreenTiger.mp4"`.**

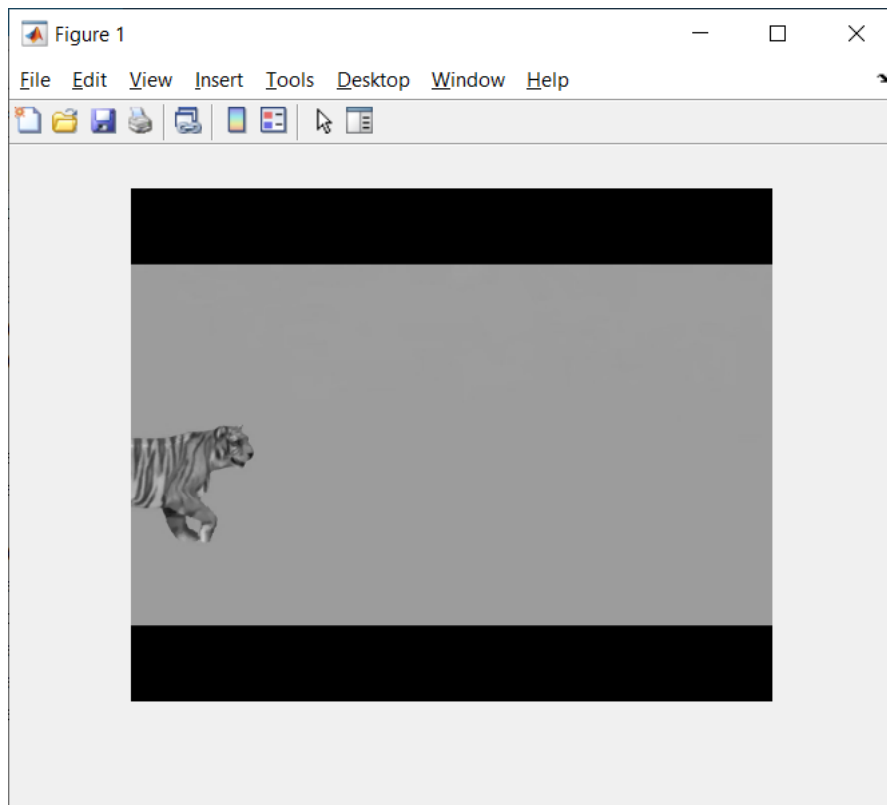
**Εκφώνηση:** Σε βίντεο της επιλογής σας, διάρκειας 5s – 10s, στο οποίο υπάρχει ήπια κίνηση αντικειμένου και κάμερας, επιλέξτε ένα αντικείμενο και εξαφανίστε το αλγοριθμικά. Δηλαδή, δημιουργήστε και αποθηκεύστε ένα νέο βίντεο στο οποίο δεν θα υπάρχει το αντικείμενο που επιλέξατε. Για τον σκοπό αυτόν, αξιοποιήστε την τεχνική αντιστάθμισης κίνησης. Υλοποιήστε και τεκμηριώστε το σχετικό σύστημα.

**Preprocessing:** Αρχικά, το πρόγραμμα διαβάζει το βίντεο ως ακολουθία πλαισίων μέσω του VideoReader και μετατρέπει τα πλαίσια αυτά από έγχρωμα σε grayscale, για να διευκολυνθεί η διαδικασία.

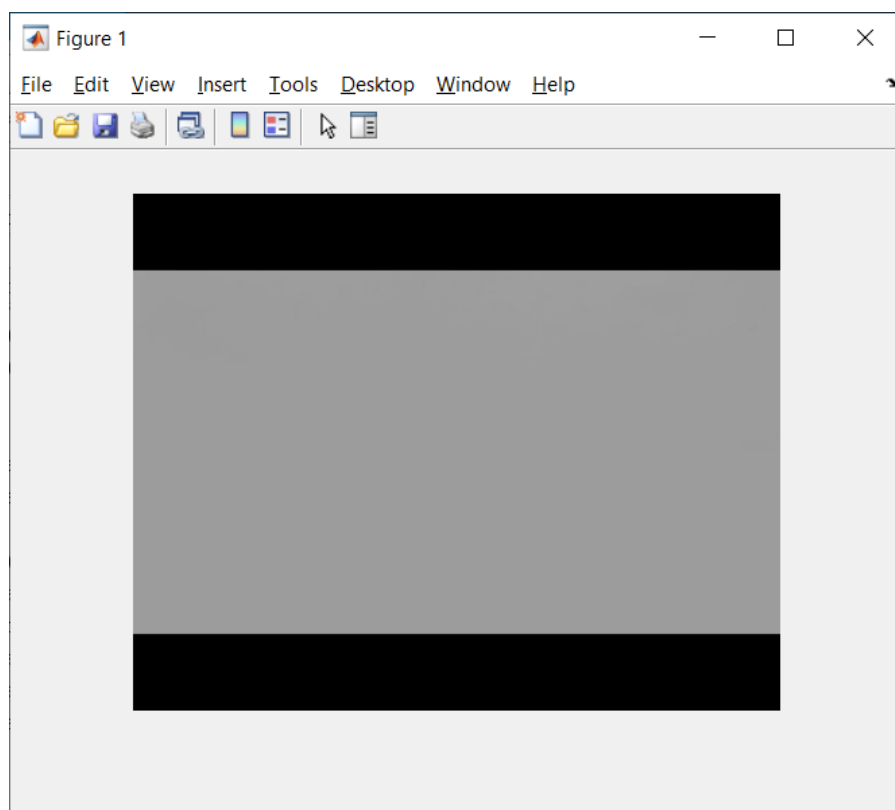
**Μεθοδολογία:** Η μεθοδολογία βασίστηκε στην εκφώνηση της άσκησης 8.18, σελ. 295 του βιβλίου. Στο βίντεο που επιλέχθηκε, το πρώτο πλαίσιο περιέχει μόνο το παρασκήνιο. Στη συνέχεια, όταν εμφανίζεται κίνηση, τα διανύσματα κίνησης που υπολογίζονται για τα μακρομπλόκ στα οποία υπάρχει αυτή η κίνηση του αντικειμένου είναι μη μηδενικά. Ο αλγόριθμός μας αρχικά αξιοποιεί τον κώδικα του προηγούμενου ερωτήματος για τον υπολογισμό των διανυσμάτων κίνησης. Στη συνέχεια ωστόσο, αντί να χρησιμοποιήσει τα διανύσματα αυτά για το ταίριασμα των μακρομπλόκ με το reference frame, ελέγχει την τιμή τους. Αν είναι μηδενικά, χρησιμοποιεί τα macroblock του ίδιου του frame, ενώ εάν είναι μη μηδενικά, που σημαίνει ότι υπάρχει κίνηση, αντικαθιστά το αντίστοιχο macroblock του frame με αυτό του reference frame, που περιέχει μόνο το παρασκήνιο. Αυτό έχει ως αποτέλεσμα να εξαφανίζονται όλα τα κινούμενα αντικείμενα.

**Results:** Το πρόγραμμα εμφανίζει το ανακατασκευασμένο βίντεο στο οποίο έχει εξαφανιστεί η κίνηση. Το βίντεο αυτό αποθηκεύεται στο αρχείο **`RemovedTiger.avi`** στον ίδιο φάκελο.





(original frame)



(frame without the moving object)

**Conclusions:** Τα κινούμενα αντικείμενα εξαφανίζονται με τέλειο τρόπο στην περίπτωση του επιλεγμένου βίντεο, το οποίο έχει ομοιόμορφο, μονόχρωμο background (green screen).

**Παρατήρηση:** Κατά τη διάρκεια ανάπτυξης του αλγορίθμου για το θέμα αυτό παρατηρήθηκε το εξής φαινόμενο: εάν το background της εικόνας δεν είναι ομοιόμορφο ή μονόχρωμο και περιέχει ένα βαθμό λεπτομέρειας και διαφοροποίησης μεταξύ διπλανών macroblock, η τεχνική αυτή δεν έχει τα επιθυμητά αποτελέσματα. Αυτό συμβαίνει επειδή λόγω της διαφορετικότητας του background του macroblock στο οποίο εμφανίζεται κίνηση με το background των γύρω περιοχών, το καλύτερο ταίριασμα-ελάχιστο MAD με το reference frame παραμένει στο ίδιο macroblock. Αυτό έχει ως αποτέλεσμα το διάνυσμα κίνησης να παραμένει μηδενικό και η κίνηση να μην εντοπίζεται. Μια λύση για το ζήτημα αυτό θα μπορούσε να είναι η επιλογή μιας μέγιστης τιμής για το κόστος MAD. Δηλαδή εάν το ελάχιστο κόστος MAD που βρέθηκε είναι μεγαλύτερο από την τιμή αυτή (δηλαδή υπάρχει διαφοροποίηση στα ταιριασμένα macroblocks του reference frame με το current frame), θεωρούμε πως υπάρχει κίνηση στο νέο frame και να αντικαθίσταται το macroblock αυτό με το αντίστοιχο του reference frame. Ωστόσο η μέγιστη τιμή αυτή δεν είναι εύκολο να βρεθεί και μπορεί να εξαρτάται και από τα χαρακτηριστικά του βίντεο του ίδιου, όπως για παράδειγμα αν περιέχει ανομοιογενές μοτίβο στο φόντο, κίνηση κάμερας, ελαφρώς κινούμενο φόντο (π.χ. λιβάδι με χόρτα που κινούνται από τον αέρα). Για το λόγο αυτό τελικά αποφασίσαμε να λειτουργήσουμε με μονόχρωμο, στατικό φόντο.

**Αρχεία και εκτέλεση:** Το script του θέματος αυτού είναι το “**thema2.m**” που βρίσκεται στον φάκελο “/Thema2”. Ο φάκελος αυτός περιέχει και μερικές βοηθητικές συναρτήσεις, οι περισσότερες των οποίων είναι όμοιες με τις αντίστοιχες στα προηγούμενα ερωτήματα. Η μόνη συνάρτηση που είναι διαφορετική από την αντίστοιχη των προηγούμενων ερωτημάτων είναι η “**compensateFrame.m**”, η οποία σε αυτό το ερώτημα χρησιμοποιεί τη μεθοδολογία που περιγράφηκε νωρίτερα (βλ. [Θέμα 2 - Μεθοδολογία](#)) και στην εκφώνηση της άσκησης 8.18 σελ. 295 του βιβλίου.

## Πηγές:

1. Συστήματα Πολυμέσων, Αλγόριθμοι, Πρότυπα & Εφαρμογές, Parag Havaladar, Gerard Medioni, Επιμέλεια Ελληνικής Έκδοσης Άγγελος Πικράκης
2. MATLAB [www.mathworks.com](http://www.mathworks.com)
3. MATLAB huffman encoder  
<https://www.mathworks.com/help/comm/ref/huffmanenco.html>
4. MATLAB huffman decoder  
<https://www.mathworks.com/help/comm/ref/huffmandeco.html>
5. MATLAB video reader  
<https://www.mathworks.com/help/matlab/ref/videoreader.html>
6. MATLAB video writer  
<https://www.mathworks.com/help/matlab/ref/videowriter.writevideo.html>
7. Διαφάνειες μαθήματος “Συστήματα Πολυμέσων” gunet2  
<https://gunet2.cs.unipi.gr/modules/document/document.php?course=TMC108&openDir=/2009111416362250xdm930>
8. Block algorithms for motion estimation, A. Barjatya  
<https://www.semanticscholar.org/paper/Block-Matching-Algorithms-For-Motion-Estimation-Barjatya/35c2fa7c4c058098288b239adfd7673a41e051e5>
9. Video processing & Communications slides, Yao Wang, NYU  
[https://eeweb.engineering.nyu.edu/~yao/EL6123\\_s16/motion2d\\_part2.pdf](https://eeweb.engineering.nyu.edu/~yao/EL6123_s16/motion2d_part2.pdf)
10. Block matching algorithms, wikipedia  
[https://en.wikipedia.org/wiki/Block-matching\\_algorithm](https://en.wikipedia.org/wiki/Block-matching_algorithm)
11. Βίντεο Θέματος 1 (κατόπιν επεξεργασίας)  
<https://www.youtube.com/watch?v=poxbBDuPfbY>
12. Βίντεο Θέματος 2 (κατόπιν επεξεργασίας)  
<https://www.youtube.com/watch?v=FCgMP7Mmwm4>
13. zigzag-izigzag code  
<https://www.mathworks.com/matlabcentral/fileexchange/15317-zigzag-scan>
14. Παράδειγμα DCT-συμπίεσης MathWorks. Τρέξτε στο terminal του matlab:  
openExample('images/ImageCompressionWithTheDiscreteCosineTransformExample')
15. Πίνακας κβάντισης JPEG  
[https://www.researchgate.net/figure/The-Default-JPEG-Quantization-Matrix-and-then-each-plane-is-individually-divided\\_fig1\\_288182261](https://www.researchgate.net/figure/The-Default-JPEG-Quantization-Matrix-and-then-each-plane-is-individually-divided_fig1_288182261)