

Συστήματα Πολυμέσων και Εικονική Πραγματικότητα

Εργασία 2007-2008

Ομάδα Κατανόησης Πολυμέσων
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

1 Εισαγωγικές Παρατηρήσεις

Η παρακάτω εργασία αποτελεί *προαιρετικό* μέρος του μαθήματος Συστήματα Πολυμέσων και Εικονική Πραγματικότητα και η εκτέλεσή της συνεισφέρει 2,3 ή και 4 επιπλέον μονάδες στην τελική βαθμολογία.

Η εργασία θα πρέπει να εκτελεστεί σε ομάδες μέχρι και δύο ατόμων.

Η εργασία αποτελείται από 3 ενότητες. Η υλοποίηση της εργασίας μπορεί κατ' επιλογή να περιλάβει την πρώτη ενότητα, την πρώτη και τη δεύτερη, κ.ο.κ. συνεισφέροντας τις αντίστοιχες μονάδες για κάθε ενότητα. Δεν μπορεί όμως να εκτελεστεί μία ενότητα χωρίς να έχει ορθά εκτελεστεί η προηγούμενή της.

Η προτεινόμενη εργασία στοχεύει στην υλοποίηση ενός κωδικοποιητή/ αποκωδικοποιητή video κατά το open source πρότυπο Theora του Xiph.org foundation.

2 Διάρθρωση και Παραδοτέα

Η εργασία θα περιλαμβάνει τις συναρτήσεις που περιγράφονται στη συνέχεια, υλοποιημένες σε matlab. Θα συνοδεύεται υποχρεωτικά από γραπτή αναφορά η οποία θα περιγράφει τον τρόπο χρήσης των προγραμμάτων και θα επιδεικνύει ενδεικτικά αποτελέσματα.

Για τον έλεγχο του κωδικοποιητή/ αποκωδικοποιητή που θα κατασκευαστεί θα χρησιμοποιηθεί ενδεικτική ακολουθία ασυμπίεστων εικόνων σε μορφή RGB-CCIR

2.1 Theora Library (2 μονάδες)

Σκοπός του πρώτου παραδοτέου είναι η δημιουργία μιας βιβλιοθήκης συναρτήσεων που θα χρησιμοποιηθούν αργότερα για την κατασκευή του κωδικοποιητή. Κάθε συνάρτηση της βιβλιοθήκης συνοδεύεται και από την αντίστροφή της που ανακατασκευάζει την είσοδο της πρώτης και που θα αποτελέσει μέρος του αποκωδικοποιητή.

Προεπεξεργασία

Η ακολουθία των εικόνων εισόδου περνά αρχικά από ένα στάδιο προεπεξεργασίας με σκοπό τη μετατροπή της σε κατάλληλο format, αλλάζοντας το χρωματικό χώρο σε YCrCb και επιλέγοντας την υποδειγματοληψία χρώματος.

Κατασκευάστε τη συνάρτηση

$$[frame] = preproc(frameRGB, PF)$$

και την αντίστροφή της

$$[frameRGB] = postproc(frame, PF)$$

όπου

frame: Το επεξεργασμένο frame. Δομή (struct) του matlab που αποτελείται από τρία στοιχεία : frame.Y, frame.Cr, frame.Cb τα οποία αντιστοιχούν στο Y, Cr, Cb component του frame.

frameRGB: Το frame προς επεξεργασία. Είναι πίνακας της μορφής $M \times N \times 3$ (όπου M και N είναι το ύψος και το πλάτος του frame αντίστοιχα σε pixels). Σύμφωνα με την ονοματολογία του προτύπου $M = FMBH \times 16$ και $N = FMBW \times 16$.

PF: Ο τρόπος υποδειγματοληψίας χρώματος. Ακέραιος αριθμός με δυνατές τιμές: '0', '2', '3', για τα formats 4:2:0, 4:2:2 και 4:4:4 αντίστοιχα (όπως φαίνεται στον πίνακα 6.4 του προτύπου). Αντίστροφα, η postproc ανακατασκευάζει την είσοδο σε format RGB 4:4:4.

Εκτίμηση και Αντιστάθμιση Κίνησης

Για την εκτίμηση κίνησης κατασκευάστε τη συνάρτηση

$$[resFrame, MVECTS, MBMODES, BCODED] = motEst(frame, pFrame, gFrame, PF, mbSearchOffset, bSearchOffset, errThres)$$

και την αντίστροφή της

$$[frame] = iMotEst(resFrame, MVECTS, MBMODES, BCODED, pFrame, gFrame, PF)$$

όπου

resFrame: Δομή με τα Y, Cr, Cb components του frame σφάλματος πρόβλεψης.

MVECTS: Ο πίνακας με τα motion vectors που αντιστοιχούν στα blocks της εικόνας. Είναι ένα cell-array $1 \times NBS$ (όπου NBS ο αριθμός blocks στο frame) και κάθε στοιχείο MVECTS(bi) είναι μια δομή με στοιχεία MVECTS(bi).MVX¹ και MVECTS(bi).MVY τα οποία εκφράζουν την οριζόντια και κατακόρυφη συνιστώσα των motion vectors για κάθε block. Σημειώστε ότι ο πίνακας MVECTS είναι φλύαρος γιατί εκτός από την περίπτωση INTER_MV_FOUR τα blocks του ίδιου macroblock έχουν τα ίδια motion vectors.

MBMODES: Περιλαμβάνει πληροφορία για τον τρόπο κωδικοποίησης των macroblocks της εικόνας. Πίνακας από ακέραιους με διαστάσεις $1 \times NMBS$ (όπου NMBS ο αριθμός των macroblock στο frame). Κάθε στοιχείο του παίρνει τιμές στο διάστημα [0, 7]. Για την αντιστοιχία τους με τα modes της κωδικοποίησης δείτε τους πίνακες 7.18 και 7.46 του προτύπου.

BCODED: Πίνακας που υποδηλώνει το αν κωδικοποιήθηκε ή όχι κάθε block της εικόνας. Έχει διαστάσεις $1 \times NBS$ και παίρνει τιμές 0 ('δεν κωδικοποιήθηκε') ή 1('κωδικοποιήθηκε'). Η απόφαση για τη μη κωδικοποίηση ενός block έχει νόημα μόνο για blocks που έχουν προβλεφθεί από το previous frame και έχουν μηδενικά motion vectors και λαμβάνεται ελέγχοντας αν το σφάλμα πρόβλεψης έχει μικρότερη ενέργεια² από το κατώφλι σφάλματος (errThres). Προσέξτε ότι ο πίνακας αυτός είναι σε coded order. Αυτό σημαίνει ότι η εικόνα χωρίζεται σε superblocks 16x16, αρχίζοντας τη μέτρηση από κάτω αριστερά, ενώ μέσα σε κάθε superblock, τα blocks διατάσσονται με hilbert order. Επίσης παρατηρήστε ότι η δεικτοδότηση αφορά και τα chroma blocks της εικόνας. Για περισσότερες πληροφορίες δείτε στη σελ. 7 του προτύπου.

frame: Δομή με τα Y, Cr, Cb components του τρέχοντος frame.

pFrame: Δομή με τα Y, Cr, Cb components του προηγούμενου frame.

gFrame: Δομή με τα Y, Cr, Cb components του golden frame.

PF: Ο τρόπος υποδειγματοληψίας χρώματος.

mbSearchOffset: Η αναζήτηση του αντίστοιχου macroblock στην εικόνα αναφοράς γίνεται με υποψήφια motion vectors που βρίσκονται στο διάστημα [-mbSearchOffset, mbSearchOffset] για κάθε κατεύθυνση.

bSearchOffset: Ειδικά στην περίπτωση της κωδικοποίησης INTER_MV_FOUR η αναζήτηση ενός καλύτερου αντίστοιχου block για κάθε luminance block του macroblock γίνεται σε μια περιοχή [-bSearchOffset, bSearchOffset] για κάθε κατεύθυνση ως προς τη θέση που είχε αρχικά επιλεγεί σε επίπεδο macroblock.

errThres: Το κατώφλι σφάλματος που χρησιμοποιείται για την απόφαση κωδικοποίησης ή όχι ενός block.

Μετασχηματισμός DCT και Zig-zag scanning

Κατασκευάστε τη συνάρτηση

$$[dctBlockCoeffs] = blockDCT(block)$$

¹Η μεταβλητή bi δεικτοδοτεί τα blocks εντός του τρέχοντος frame και παίρνει τιμές από 1 ως NBS. Υπονοείται ότι η δεικτοδότηση αυτή ακολουθεί το coded order, όπως περιγράφεται στην ενότητα 2.3 του προτύπου.

²άθροισμα τετραγώνων των τιμών του σφάλματος πρόβλεψης.

και την αντίστροφή της

$$[block] = iBlockDCT(dctBlockCoeffs)$$

όπου

dctBlockCoeffs: Οι DCT coefficients του block 8x8 σε διάταξη zig-zag. Πίνακας με διαστάσεις 1×64 .

block: Το block της εισόδου. Πίνακας 8x8.

Για την υλοποίηση αυτού του βήματος συνιστάται η χρήση της συνάρτησης dct2 της matlab. Υπενθυμίζεται ότι ο μετασχηματισμός εφαρμόζεται σε επίπεδο block 8x8.

Κβαντισμός

Για τον κβαντισμό των blocks κατασκευάστε τις συναρτήσεις

$$[blockCOEFFS] = blockQuantize(dctBlockCoeffs, BM, DCSCALE, DCSCALE, qti, qi)$$

$$[DQC] = iBlockQuantize(blockCOEFFS, BM, DCSCALE, DCSCALE, qti, qi)$$

όπου

blockCOEFFS: Τα σύμβολα κβαντισμού των DCT coefficients του block.

dctBlockCoeffs: Οι DCT coefficients εισόδου σε διάταξη zig-zag.

BM: Ο πίνακας κβαντισμού σε διάταξη zig-zag και έχει διαστάσεις 1×64 .

DCSCALE: Πίνακας 1×64 με τις κλίμακες κβαντισμού για τον DC συντελεστή. Η κλίμακα κβαντισμού του συγκεκριμένου block επιλέγεται, από αυτόν τον πίνακα, ως DCSCALE(qi).

ACSCALE: Πίνακας 1×64 με τις κλίμακες κβαντισμού για τους AC συντελεστές. Η κλίμακα κβαντισμού του συγκεκριμένου block επιλέγεται, από αυτόν τον πίνακα, ως ACSCALE(qi).

DQC: Οι ανακατασκευασμένοι DCT coefficients του block σε διάταξη zig-zag.

qti: Quantization type index. Αριθμός που παίρνει τις τιμές '0' ή '1' ανάλογα με το αν το block είναι κωδικοποιημένο κατά INTRA ή όχι, αντίστοιχα.

qi: Quantization index. Αριθμός που χρησιμοποιείται για να επιλεγούν οι πίνακες κβαντισμού ac και dc και κυμαίνεται από το 1 ως το 64 (για λόγους δεκτοδότησης στο matlab). Πρόκειται για έναν αριθμό που εκφράζει την ποιότητα της κβάντισης. Όσο μικρότερος είναι, τόσο πιο 'χονδρικοί' είναι οι πίνακες κβαντισμού κάτι που οδηγεί σε μειωμένη ποιότητα, αλλά και σε μεγαλύτερη συμπίεση. Για περισσότερες πληροφορίες δείτε στη σελίδα 13 του προτύπου.

Ο κβαντισμός γίνεται με διαίρεση των συντελεστών DCT με τα αντίστοιχα στοιχεία ενός πίνακα QMAT και στη συνέχεια στρογγυλοποίηση στον πλησιέστερο ακέραιο στο διάστημα $[-580, 580]$. Έτσι

$$blockCOEFFS(ci) = fix\left(\frac{dctBlockCoeffs(ci)}{QMAT(ci)}\right)$$

Οι τιμές που είναι εκτός του διαστήματος $[-580, 580]$ αποκρίνονται.

Ο αποκβαντισμός γίνεται με την αντίστροφη διαδικασία:

$$DQC(ci) = blockCOEFFS(ci) \times QMAT(ci)$$

Οι τιμές του QMAT δίνονται από τη σχέση:

$$QMAT(ci) = \max(QMIN, \min\left(fix\left(\frac{QSCALE \times BM(ci)}{100}\right) \times 4, 4096\right))$$

Ο συντελεστής QSCALE αντιστοιχεί στα DCSCALE(qi) και ACSCALE(qi) ανάλογα με το είδος του συντελεστή ενώ το QMIN υπολογίζεται σύμφωνα με τον πίνακα 6.18 του προτύπου.

Υπενθυμίζεται ότι, κατά τον κβαντισμό, το αποτέλεσμα που προκύπτει από το ακέραιο μέρος της διαίρεσης με το βήμα κβαντισμού είναι σύμβολο και όχι νούμερο (τιμή φωτεινότητας ή χρωματικότητας).

Run Length Encoding των συντελεστών DCT

Το πρότυπο χρησιμοποιεί δύο κατηγορίες μηκών διαδρομής. Η πρώτη κωδικοποιεί α) μεμονωμένους συντελεστές DCT β) n-άδες μηδενικών όπως εμφανίζονται σε διάταξη zig-zag και γ) n-άδες μηδενικών ακολουθούμενες από έναν κβαντισμένο (μη-μηδενικό) συντελεστή DCT. Η δεύτερη κατηγορία κωδικοποιεί τα σύμβολα τερματισμού των blocks (EOB), όπως θα δούμε παρακάτω.

Τα μήκη διαδρομής ονομάζονται tokens και κάθε token χαρακτηρίζεται από μια τριάδα ακεραίων:

$$(ti, token_value, extra_bits)$$

Όπου ti (token index): δηλώνει τη θέση του πρώτου συντελεστή που αφορά το token. Στην περίπτωση της πρώτης κατηγορίας, ο συνδυασμός του token value και των extra bits σημαίνει το συνδυασμό μήκους διαδρομής μηδενικών ή/και συντελεστών που ακολουθούν (βλέπε πίνακα 7.38 του προτύπου). Στην περίπτωση της δεύτερης κατηγορίας, ο συνδυασμός του token value και των extra bits σημαίνει το πλήθος των EOBs που κωδικοποιεί το token (βλέπε πίνακα 7.33 του προτύπου).

Ο υπολογισμός των tokens της πρώτης κατηγορίας γίνεται σε επίπεδο block.

Τα tokens που αντιστοιχούν σε σύμβολα μηδενικών στο τέλος του block παραλείπονται. Η λίστα των tokens συμπληρώνεται με ένα προσωρινό σύμβολο EOB.

Στη συνέχεια, τα tokens που έχουν παραχθεί από τα blocks ολόκληρου του frame διατάσσονται κατά αύξουσα σειρά των ti τους.

Αν στη διάταξη αυτή ανιχνευθούν ακολουθίες από EOB, αυτές κωδικοποιούνται με tokens της δεύτερης κατηγορίας.

Η αποκωδικοποίηση μήκους διαδρομής ενός frame γίνεται παράλληλα για όλα τα blocks και για κάθε ένα από αυτά φυλάσσεται στη μνήμη ένα current token index (TIS(bi)). Βλέπε ενότητα 7.7 του προτύπου.

Παράδειγμα: Έστω ότι έχουμε να κωδικοποιήσουμε τα εξής blockCOEFFS:

$$b1 : [2, \underbrace{0 \dots 0}_8, 3, 0, 4, -5, 0, 1, \underbrace{0 \dots 0}_{49}]$$

$$b2 : [-3, \underbrace{0 \dots 0}_3, 3, 4, 5, \underbrace{0 \dots 0}_{30}, 5, \underbrace{0 \dots 0}_{26}] \text{ και}$$

$$b3 : [7, \underbrace{0 \dots 0}_{10}, 9, 420, \underbrace{0 \dots 0}_8, -1, \underbrace{0 \dots 0}_{42}]$$

Στο πρώτο βήμα υπολογίζονται τα tokens για κάθε block ³:

block element	token
2	(0,11,b)
$\underbrace{0 \dots 0}_8$	(1,7,b111)
3	(9,13,b0)
0	(10,7,b000)
4	(11,14,b0)
-5	(12,15,b1)
0, 1	(13,23,b0)
$\underbrace{0 \dots 0}_{49}, EOB$	(15,EOB)

³τα extra bits κάθε token αναπαρίστανται σε δυαδική μορφή. Έτσι το b101 συμβολίζει τη δυαδική λέξη 101 ενώ το b σημαίνει ότι δεν υπάρχουν extra bits

block element	token
-3	(0,13,b1)
$\underbrace{0 \dots 0}_3, 3$	(1,31,b011)
4	(5,14,b0)
5	(6,15,b0)
$\underbrace{0 \dots 0}_{30}$	(7,8,b011101)
5	(37,15,b0)
$\underbrace{0 \dots 0}_{26}, EOB$	(38,EOB)

b2:

block element	token
7	(0,17,b00)
$\underbrace{0 \dots 0}_{10}, 3$	(1,8,b001001)
9	(11,18,b000)
420	(12,22,b0101011111)
$\underbrace{0 \dots 0}_8, -1$	(13,28,b110)
$\underbrace{0 \dots 0}_{42}, EOB$	(22,EOB)

b3:

Στο δεύτερο βήμα διατάσσονται σύμφωνα με το ti τους και για το ίδιο ti σύμφωνα με τον αύξοντα αριθμό block, bi . Οπότε, θεωρώντας για λόγους συντομίας ότι το frame περιέχει μόνο αυτά τα τρία blocks, προκύπτει ο κάτωθι πίνακας:

no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
ti	0	0	0	1	1	1	5	6	7	8	9	10	11	11	12	12	13	14	22	37	38
bi	1	2	3	1	2	3	2	2	2	1	1	1	1	3	1	3	3	1	3	2	2

Ανιχνεύοντας τώρα ακολουθίες από EOBs σε αυτή τη λίστα, παρατηρούμε ότι οι στήλες 18 και 19 αντιστοιχούν σε δύο διαδοχικά EOBs ενώ η στήλη 21 περιέχει ένα μεμονωμένο EOB. Κατά συνέπεια, εφαρμόζοντας την κωδικοποίηση του πίνακα 7.33, τα δύο διαδοχικά EOB tokens κωδικοποιούνται ως (1,b) και το τελευταίο ως (0,b).

Με βάση τα παραπάνω κατασκευάστε τη συνάρτηση υπολογισμού των κωδικοποιημένων κατά huffman συμβόλων μήκους διαδρομής tokens για τους κβαντισμένους συντελεστές DCT.

$$[tokenStream] = runLengthFrame(COEFFS, BCODED)$$

και την αντίστροφή της

$$[COEFFS] = irunLengthFrame(tokenStream, BCODED)$$

όπου

$tokenStream$: Πίνακας από bits που περιέχει τα κωδικοποιημένα tokens για το συγκεκριμένο frame. Επίσης περιέχει τις τιμές των $htiL$ και $htiC$, που θα χρησιμοποιηθούν για να υπολογίσουμε στην αποκωδικοποίηση τους επιλεγέντες πίνακες κβαντισμού (βλ. σελ 94-95 του προτύπου), τοποθετημένες ακριβώς πριν από τα στοιχεία που αντιστοιχούν σε $ti=0$ ($htiL_DC$ και $htiC_DC$) και $ti=1$ ($htiL_AC$ και $htiC_AC$).

$COEFFS$: Τα σύμβολα κβαντισμού των DCT coefficients. Cell array που περιέχει τα $blockCOEFFS$ για τα κωδικοποιημένα blocks και κενούς πίνακες για τις θέσεις που αντιστοιχούν σε μη-κωδικοποιημένα blocks ($BCODED(bi)=0$).

$BCODED$: Πίνακας που υποδηλώνει το αν κωδικοποιήθηκε ή όχι κάθε block της εικόνας. Χρησιμοποιείται για να παρακάμπτονται τα blocks που δεν κωδικοποιήθηκαν.

Επίσης, κατασκευάστε τη συνάρτηση

$$[vlcToken] = vLC(token, ti, blockType)$$

και την αντίστροφή της

$$[token] = iVLC(vlcToken, ti, blockType)$$

όπου

vlcToken: Πίνακας από bits. Ένα token κωδικοποιημένο με huffman.

ti: Πίνακας με τις τιμές των token indices (ti).

token: Δομή της μορφής token.tokenValue και token.extraBits και περιέχει δύο στοιχεία: τον ακέραιο token value και μια δυαδική συμβολοσειρά με τα extra bits στο φορμά της υποσημείωσης.

Για την κωδικοποίηση μεταβλητού μήκους διαδρομής, η συνάρτηση runLengthFrame θα καλεί την υποσυνάρτηση vlc και θα προσθέτει το αποτέλεσμα της στο tokenStream.

Αντίστοιχα, στην αποκωδικοποίηση, η συνάρτηση iVLC καλείται από την iRunLengthFrame.

Οι συναρτήσεις αυτές πρέπει να (απο-)κωδικοποιούν τα σύμβολα μήκους διαδρομής σύμφωνα με τους κώδικες που περιλαμβάνονται στους πίνακες της ενότητας B.4 του προτύπου οι οποίοι πρέπει να είναι προσβάσιμοι σαν global μεταβλητές.

Παρατηρήσεις:

1. Για την υλοποίησή μας θα θεωρήσουμε (για λόγους απλούστευσης) ότι οι προς κωδικοποίηση εικόνες έχουν διαστάσεις πολλαπλάσιες του 32. Με αυτόν τον τρόπο εξασφαλίζουμε ότι περιέχουν ακέραιο αριθμό από superblocks, ενώ έτσι επίσης το picture region ταυτίζεται με το frame (βλ. σελ. 7-10 του προτύπου).

2. Για τις συναρτήσεις κωδικοποίησης/αποκωδικοποίησης huffman (vLC-iVLC) θα χρειαστούν οι πίνακες του παραρτήματος B.4 (σελ. 145) του προτύπου⁴, ενώ μέσα σε αυτές μπορείτε να καλείτε τις συναρτήσεις του matlab (huffmanenco, huffmandeco, huffmandict).

2.2 Theora Processing (+1 μονάδα)

Σε αυτό το βήμα θα χρησιμοποιηθούν οι προηγούμενες συναρτήσεις για την κατασκευή ενός codec, ο οποίος θα είναι δομημένος σε επίπεδα, όπως αυτά ορίζονται από το πρότυπο. Συγκεκριμένα, υπάρχουν τα επίπεδα κωδικοποίησης Sequence και Frame, τα οποία θα χειρίζονται οι αντίστοιχες συναρτήσεις.

2.2.1 Δομή Αποθήκευσης

Το αποτέλεσμα της διαδικασίας encodeTheora θα αποθηκευθεί σε δομή τύπου struct του matlab. Αυτή παρουσιάζεται στους πίνακες που ακολουθούν.

Δομή	Στοιχείο	Σχόλια
SeqEntity	SeqHeader (1) FrameEntityArray(1,*)	Sequence Header Πίνακας από δομές τύπου FrameEntity
FrameEntity	FrameHeader (1) FrameData (1)	Frame Header Τα πραγματικά δεδομένα του Frame

⁴Θα ήταν καλή ιδέα, αντί να τους αντιγράψει καθέννας από το πρότυπο, να συνεννοηθείτε και να μοιράσετε τη δουλειά.

Δομή	Στοιχεία	Σχόλια
SeqHeader	IDHeader (1) CommentHeader (1) SetupHeader (1)	Identification Header Comment Header Setup Header
IDHeader	VMAJ*, VMIN*, VREV*, FMBW, FMBH NSBS, NBS, NMBS, PICW, PICH, PICX* PICY*, FRN*, FRD*, PARN*, PARD*, CS, PF NOMBR*, QUAL*, KFGSHIFT*	Όπως αυτά περιγράφονται στις σελίδες 40-41 του προτύπου.
CommentHeader	VENDOR, NCOMMENTS COMMENTS	Βλέπε σελ. 47 του προτύπου. Cell array που περιέχει NCOMMENTS συμβολοσειρές σχολίων
SetupHeader	LFLIMS*, ACSCALE, DCSCALE, NBMS BMS, NQRS, QRSIZES, QRBMS, HTS	Βλέπε σελ. 58-59 του προτύπου.
FrameHeader	FTYPE, NQIS, QIS	Βλέπε σελ. 61 του προτύπου.
FrameData	BCODED MBMODES MVECTS QIIS vlcStream	Όπως στο πρώτο μέρος της εργασίας Βλέπε σελ. 79 του προτύπου Το αποτέλεσμα της κωδικοποίησης μεταβλητού μήκους λέξης

Να σημειωθεί ότι οι μεταβλητές που χρησιμοποιούνται στη δική μας, απλουστευμένη, υλοποίηση υπάρχουν αλλά μπορούν να αφηθούν κενές σε αυτό το επίπεδο καθώς αγνοούνται στην αποκωδικοποίηση. Αυτές οι μεταβλητές σημειώνονται με έναν αστερίσκο (*).

Επίσης, στη δική μας υλοποίηση χρησιμοποιούμε μόνο ένα quantization index (qi) για όλη την ακολουθία. Συνεπώς το στοιχείο NQIS του FrameHeader είναι πάντα 1, ενώ ο πίνακας QIS περιέχει μόνο ένα στοιχείο: το ίδιο το qi.

2.2.2 Ορισμοί Συναρτήσεων και Δομών Δεδομένων

Κύριες Συναρτήσεις

```
function SeqEntity=encodeTheora(fName, fExtension, startFrame, ...
    GoPFrames, stopFrame, seqHeader, outFName, qi)
```

fName: Το πρόθεμα της σειράς των αρχείων εικόνων.

fExtension: Η επέκταση της σειράς των αρχείων εικόνων.

startFrame: Ο αύξων αριθμός της πρώτης εικόνας.

GoPFrames: Ο αριθμός των frames κωδικοποιούνται σαν ομάδα (Group of Pictures - GoP), με αναφορά στο ίδιο Golden (INTRA) frame. Θεωρούμε, δηλαδή, ότι κάθε INTRA frame ακολουθείται από (GoPFrames-1) INTER-frames, ενώ μετά ξεκινά ένα νέο GoP κ.ο.κ.

stopFrame: Ο αύξων αριθμός της τελευταίας εικόνας.

seqHeader: Μια δομή τύπου seqHeader όπως ορίστηκε παραπάνω. Περιέχει τα μεταδεδομένα που θα χρειαστούν στην κωδικοποίηση.

outFName: Το όνομα του αρχείου (.mat στο οποίο αποθηκεύεται η έξοδος).

qi: Ο δείκτης στους πίνακες κβαντισμού DCSCALE, ACSCALE, όπως περιγράφεται στη συνάρτηση blockQuantize του πρώτου μέρους. Αποτελεί μέτρο της ποιότητας κωδικοποίησης και χρησιμοποιείται για τον έλεγχο μεγέθους/ποιότητας. Στο πρότυπο μπορεί να είναι διαφορετικός για κάθε frame, αλλά εμείς επιλέγουμε να είναι σταθερός για όλη την ακολουθία εικόνων.

SeqEntity: Δομή τύπου SeqEntity που περιέχει το κωδικοποιημένο βίντεο.

```
function FrameEntityArray=encodeSeq(fName, fExtension, startFrame, ...
    GoPFrames, stopFrame, seqHeader, qi)
```

fName: Το πρόθεμα της σειράς των αρχείων εικόνων.

fExtension: Η επέκταση της σειράς των αρχείων εικόνων.

startFrame: Ο αύξων αριθμός της πρώτης εικόνας.

GoPFrames: Ο αριθμός των frames κωδικοποιούνται σαν ομάδα.

stopFrame: Ο αύξων αριθμός της τελευταίας εικόνας.

qi: Ο δείκτης στους πίνακες κβαντισμού DCSCALE, ACSCALE.
FrameEntityArray: Πίνακας από δομές τύπου GoPEntity.

```
function FrameData=encodeFrame(frame, gFrame, pFrame, SeqHeader, qi)
```

frame: Δομή με τα Y, Cr, Cb components του frame προς κωδικοποίηση.
gFrame: Δομή με τα Y, Cr, Cb components του τρέχοντος golden frame.
pFrame: Δομή με τα Y, Cr, Cb components του τρέχοντος previous frame.
qi: Ο δείκτης στους πίνακες κβαντισμού DCSCALE, ACSCALE.
SeqHeader: Δομή τύπου SeqHeader.

```
function decodeTheora(fName, outFName)
```

fName: Το όνομα του αρχείου τύπου mat όπου είναι αποθηκευμένη η δομή του βίντεο.
outFName: Η βάση του ονόματος με το οποίο θα αποθηκευτούν οι αποκωδικοποιημένες RGB εικόνες. Το όνομα θα έχει τη μορφή [όνομα βάσης, αριθμός καρέ, '.tiff'].

```
function decodeSeq(SeqEntity, outFName)
```

SeqEntity: Δομή τύπου SeqEntity.
outFName: Η βάση του ονόματος με το οποίο θα αποθηκευτούν οι αποκωδικοποιημένες RGB εικόνες. Το όνομα θα έχει τη μορφή [όνομα βάσης, αριθμός καρέ, '.tiff'].

```
function frame=decodeFrame(FrameEntity, gFrame, pFrame, SeqHeader)
```

FrameEntity: Δομή τύπου FrameEntity.
frame: Η αποκωδικοποιημένη εικόνα. Δομή με τα Y,Cr,Cb components.
gFrame: Δομή με τα Y, Cr, Cb components του τρέχοντος golden frame.
pFrame: Δομή με τα Y, Cr, Cb components του τρέχοντος previous frame.
SeqHeader: Δομή τύπου SeqHeader.

2.3 Theora Evaluation (+1 μονάδα)

Το τελευταίο μέρος της εργασίας είναι μια αξιολόγηση της συμπίεσης που έχει επιτευχθεί.

Τα κομμάτια που περιέχουν την κυρίως πληροφορία του κωδικοποιημένου βίντεο είναι τα motion vectors και φυσικά τα tokens των κβαντισμένων dct συντελεστών.

Για να υπολογίσει κανείς την εντροπία τους, χρειάζεται να μετρήσει τις σχετικές συχνότητες εμφάνισης όλων των συμβόλων που αυτά παράγουν. Συγκεκριμένα, τα σύμβολα των motion vectors είναι όλοι οι εμφανιζόμενοι συνδυασμοί των ζευγαριών (x, y). Ο αριθμός των πιθανών συνδυασμών εξαρτάται από το εύρος των τιμών που μπορούν να πάρουν τα x και y (βλ. μεταβλητές mbSearchOffset, bSearchOffset στη συνάρτηση motEst). Αντίστοιχα, τα πιθανά σύμβολα στα tokens είναι όλοι οι συνδυασμοί των δυάδων token value, extra bits που προκύπτουν από την runLengthFrame.

Για τον υπολογισμό της εντροπίας, θα δημιουργηθούν δύο global μεταβλητές που θα περιέχουν μετρητές εμφάνισης κάθε συμβόλου: mvCounts: cell-array διάστασης 2x(αρ. συμβόλων). Η πρώτη στήλη περιλαμβάνει όλα τα δυνατά σύμβολα σε μορφή συμβολοσειράς και η δεύτερη τον αριθμό εμφάνισής τους.

tokenCounts: Όμοια με την mvCounts, που όμως αναφέρεται στα σύμβολα των tokens.

Στο σημείο αυτό θα γίνουν μερικές τροποποιήσεις στις συναρτήσεις motEst και runLengthFrame, οι οποίες, επιπλέον από τις λειτουργίες τους, θα επιδρούν στις παραπάνω global μεταβλητές, αυξάνοντας κατά ένα τον αντίστοιχο μετρητή για κάθε σύμβολο που συναντούν.

Επιπλέον, για την αξιολόγηση της διαδικασίας, κατασκευάστε τη συνάρτηση:

```
[tSize, rSize, huffRatio] = huffStats(SeqEntity)
```


SeqEntity: Το κωδικοποιημένο βίντεο του δεύτερου βήματος σε μορφή SeqEntity.

tSize: Το θεωρητικό ελάχιστο μέγεθος των motion vectors και των tokens. Υπολογίζεται από το άθροισμα της εντροπίας τους.

rSize: Το πραγματικό μέγεθος των motion vectors και των tokens, μετά την κωδικοποίηση huffman.

huffRatio: Ο λόγος του θεωρητικού προς το μέγεθος που επιτεύχθηκε μέσω huffman. Μας δείχνει την επίδοση του huffman (πόσο καλά προσεγγίζει την κωδικοποίηση εντροπίας) και ισούται με

$$huffRatio = \frac{tSize}{rSize}$$

Η συνάρτηση αυτή μετρά την εντροπία των motion vectors και των tokens χρησιμοποιώντας τους μετρητές των μεταβλητών tokenCounts και mvCounts (των οποίων και η ύπαρξη στη μνήμη προϋποτίθεται) και το ελάχιστο μέγεθος της ακολουθίας πολλαπλασιάζοντας με τον αριθμό των motion vectors και των tokens αντίστοιχα και προσθέτοντας τα extra bits για κάθε token. Το πραγματικό μέγεθος αντιστοιχεί στο μήκος των ακολουθιών, όπως αυτές είναι αποθηκευμένες σε κάθε FrameEntity της ακολουθίας.

Για τον υπολογισμό της συμπίεσης θα ληφθούν υπόψη και οι headers. Το μέγεθός τους θα εκτιμηθεί προσεγγιστικά, ανάλογα με τον τρόπο συμπίεσης της ακολουθίας και σύμφωνα με το κεφάλαιο 6 του προτύπου.

Για το σκοπό αυτό, θα κατασκευαστεί η παρακάτω συνάρτηση η οποία θα καλεί την huffStats και θα υπολογίζει το βαθμό συμπίεσης λαμβάνοντας υπόψη και τους headers.

```
[theoraSize, rawSize, theoraRatio, tSize, rSize, huffRatio] = ...  
theoraStats(SeqEntity)
```

SeqEntity: Το κωδικοποιημένο βίντεο του δεύτερου βήματος σε μορφή SeqEntity.

theoraSize: Το μέγεθος του συμπιεσμένου βίντεο.

rawSize: Το μέγεθος του ασυμπιεστού βίντεο.

theoraRatio: Ο λόγος της συμπίεσης που επιτεύχθηκε μέσω του theora και ισούται με

$$theoraRatio = \frac{theoraSize}{rawSize}$$

tSize: Το θεωρητικό ελάχιστο μέγεθος των motion vectors και των tokens.

rSize: Το πραγματικό μέγεθος των motion vectors και των tokens, μετά την κωδικοποίηση huffman.

huffRatio: Ο λόγος του θεωρητικού ελάχιστου προς το μέγεθος που επιτεύχθηκε μέσω huffman