ΤΜΗΜΑ ΗΛΕΚΤΡΩΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
&
ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«ΣΥΜΠΙΕΣΗ ΕΙΚΟΝΑΣ ΜΕ ΤΗΝ ΧΡΗΣΗ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ»

Εισηγητής και Επιβλέπων καθηγητής: Πέτρος Σταυρουλάκης

Μέλη της τριμελούς επιτροπής: Μιχάλης Ζερβάκης
Πέτρος Σταυρουλάκης
Μανώλης Χριστοδούλου

Δημήτριος Α. Αραμπατζής

ΧΑΝΙΑ 1999

Περιεχόμενα

Πρόλογος

Η συμπίεση εικόνας είναι ένα σημαντικό αντικείμενο στην ψηφιακή επεξεργασία εικόνας και ο ρόλος της γίνεται ακόμα πιο σημαντικός όταν ψηφιακές εικόνες πρέπει να μεταδοθούν ή να αποθηκευτούν με έναν ικανοποιητικό τρόπο. Σ' αυτή την εργασία παρουσιάζεται μια μέθοδος στην ένας Block Truncation κωδικοποιητής (BTC) συνδέεται σειριακά με έναν Vector Quantizer (VQ), ο οποίος είναι βασισμένος σε ένα νευρωνικό δίκτυο. Ο προτεινόμενος κωδικοποιητής είναι πολύ ελκυστικός για μεταδόσεις σε πραγματικό χρόνο εξαιτίας της απλότητας και της απόδοσής του. Διατηρεί σημαντικά χαρακτηριστικά της εικόνας, ενώ με την διαδοχική τοποθέτηση του VQ πετυχαίνουμε υψηλούς λόγους συμπίεσης της τάξεως του 0.5 bpp, χωρίς να αυξάνεται σημαντικά ο χρόνος κωδικοποίησης.

Στο 2ο κεφάλαιο παρουσιάζεται η συμπίεση εικόνας και το εύρος των εφαρμογών της, όπως επίσης και οι γενικές κατηγορίες των lossy και lossless αλγορίθμων συμπίεσης. Μια γενική εισαγωγή για τα νευρωνικά δίκτυα γίνεται στο κεφάλαιο 3, ενώ το κεφάλαιο 4 ασχολείται με τον βασικό BTC αλγόριθμο όπως και με κάποιες τροποποιήσεις του που θα χρησιμοποιηθούν στον προτεινόμενο κωδικοποιητή. Παρουσιάζονται επίσης και τα αποτελέσματα των προσομοιώσεων του αλγορίθμου. Στο 5ο κεφάλαιο παρουσιάζεται ο βασισμένος σε νευρωνικό δίκτυο vector quantizer και στο 6ο η περιγραφή και τα αποτελέσματα του προτεινόμενου κωδικοποιητή.

Τέλος στο 7ο αναφέρονται μερικά συμπεράσματα και κάποιες κατευθύνσεις για περαιτέρω εργασίες της προτεινόμενης μεθόδου.

Θεωρώ χρέος μου να ευχαριστήσω τον καθηγητή του Πολυτεχνείου Κρήτης και διευθυντή του Ινστιτούτου Τηλεπικοινωνιακών Συστημάτων, κ. Πέτρο Σταυρουλάκη για την ανάθεση και επίβλεψη της εργασίας αυτής. Όπως επίσης θα ήθελα να να ευχαριστήσω τους Κώστα Αγγελλάκη, Γίωργο Μαραγκάκη και Χάρη Σανδαλίδη που με τις εύστοχες παρεμβάσεις τους βοήθησαν στην διαικπεραίωση της εργασίας αυτής.

1. Εισαγωγή

Η επιδίωξη της συμπίεσης δεδομένων (data compression) είναι να αφαιρέσει τον πλεονασμό που ίσως υπάρχει και έτσι να μειώσει τον χρόνο μετάδοσης τους ή τον χώρο αποθήκευσής τους. Η συμπίεση δεδομένων είναι η διαδικασία κωδικοποίησης δεδομένων, η οποία μειώνει τις απαιτήσεις αποθήκευσης (storage requirements). Ένα τέτοιο σχήμα αυξάνει τη διεκπεραιωτικότητα (throughput), βελτιώνει την δικτυακή ασφάλεια (network security) και απαλλάσσει τους προγραμματιστές από την διαδικασία ικανοποιητικού πακεταρίσματος των δεδομένων. Η συμπίεση δεδομένων λειτουργεί σε λογικό επίπεδο και έτσι δεν ξεπερνιέται τεχνολογικά από την ραγδαία τεχνολογική πρόοδο των μεσών αποθήκευσης και των δικτύων.

Η έρευνα πάνω στην συμπίεση δεδομένων σημείωσε αξιοσημείωτη πρόοδο με την εργασία του Shannon. Σήμερα η έρευνα πάνω στον τομέα αυτό είναι σημαντική για την επιστήμη Υπολογιστών, επειδή εξετάζει ένα πολύ σημαντικό trade-off στην αναπαράσταση δεδομένων (data representation): όσο το δυνατόν μικρότερος χώρος – ταχύτητα. Για να αποθηκεύσουμε ή να μεταδώσουμε δεδομένα πρέπει να διαλέξουμε τον τρόπο αναπαράστασης τους, αφού δεν υπάρχουν κάποιες default αναπαραστάσεις. Έχοντας αυτό στο μυαλό μας, θα προχωρήσουμε στην εξέταση της απεικόνισης δεδομένων, έχοντας στραμμένη την προσοχή μας στην συμπίεση.

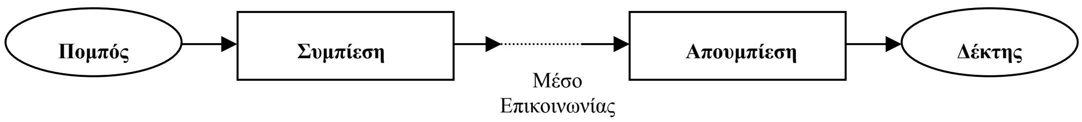
Μπορούμε να δούμε την συμπίεση δεδομένων σαν την μετάφραση ενός μηνύματος μιας απεικόνισης σε μια άλλη απεικόνιση, η οποία χρησιμοποιεί λιγότερο χώρο αποθήκευσης. Η συμπίεση δεδομένων είναι σχετικά ορισμένη και είναι λάθος να αναφερθούμε σε κάποια λεπτομερής απεικόνιση σαν συμπιεσμένη, χωρίς πρώτα να ορίσουμε την ασυμπίεστη απεικόνιση, με την οποία την συγκρίνουμε. Στην πράξη, η απόδοση της συμπίεσης συνήθως εκφράζεται σχετικά με ένα σύνολο από φυσικές απεικονίσεις. Το σύνολο αυτό απεικονίσεων χρησιμοποιεί το ίδιο ποσό χώρου για όλες τις τιμές δεδομένων. Οι συμπιεσμένες απεικονίσεις χρησιμοποιούν λιγότερο κατά μέσο όρο χώρο.

Για να χρησιμοποιηθεί λιγότερος χώρος από ότι μια φυσική απεικόνιση, οι συμπιεσμένες απεικονίσεις χρησιμοποιούν μεταβλητού μήκους κωδικοποιημένες συμβολοσειρές για να απεικονίσουν τις διαφορετικές τιμές δεδομένων. Αυτή η διαφοροποίηση στο μήκος συνήθως αυξάνει τον χρόνο που απαιτήτε για να εκτελεστούν λειτουργίες στις συμπιεσμένες απεικονίσεις. Για παράδειγμα, στοιχεία σε έναν συμπιεσμένο πίνακα δεν μπορούν να προσπελαστούν τυχαία. Για αυτό τον λόγο απαιτούμε να χρησιμοποιούμε τεχνικές συμπίεσης μόνο εκεί όπου το κόστος χώρου είναι μεγάλο σχετικά με το κόστος του χρόνου. Τύποι δεδομένων στους οποίους χρησιμοποιήται η τεχνολογία της συμπίεσης δεδομένων είναι:

- Bit-maps
- Φωνή και μουσική
- Gray-scale και έγχρωμες εικόνες
- Video και animation
- Ιατρικά δεδομένα
- Γραφικά και δεδομένα CAD

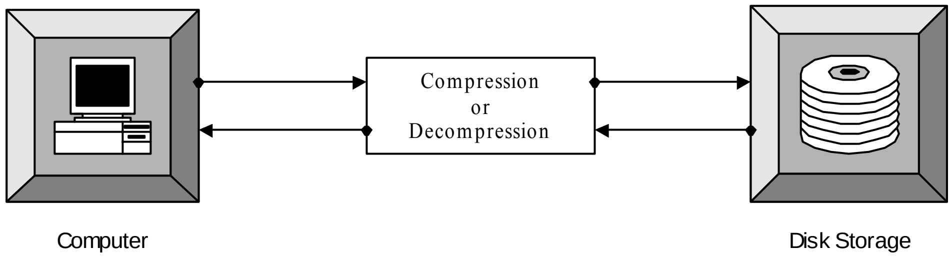
Πριν την συμπίεση δεδομένων δίνεται έμφαση στην ικανοποιητική απεικόνιση αναλογικών σημάτων σε ψηφιακά. Και οι δύο λειτουργίες θα πρέπει να παράγουν το καλύτερο δυνατό αποτέλεσμα από πλευράς πιστότητας υπό τους περιορισμούς του ρυθμού δεδομένων (bit rate) και της πολυπλοκότητα υλοποίησης. Η μετατροπή των σημάτων σε ψηφιακές απεικονίσεις έχει μερικούς στόχους:

- Να ελαχιστοποιήσει την χωρητικότητα που απαιτείται για μετάδοση υψηλής ποιότητας σημάτων όπως φωνή και εικόνες ή ισοδύναμα να πάρει την καλύτερη δυνατή πιστότητα μέσο ενός δοσμένου ψηφιακού κανάλι επικοινωνίας. Έτσι με το να συμπιέζουμε δεδομένα στον πομπό και να αποσυμπιέζουμε στον δέκτη, μπορούμε να αυξήσουμε ικανοποιητικά το bandwidth του μέσου επικοινωνίας (σχήμα 1.1).



Σχήμα 1.1 Συμπίεση σε μέσο επικοινωνίας

- Για την ελαχιστοποίηση του αποθηκευτικό όγκο που απαιτείται για αποθήκευση τέτοιου είδος πληροφορίας σε αποθηκευτικά μέσα ή σε βάσεις δεδομένων. Με μεθόδους software ή hardware, οι οποίες συμπιέζουν ένα κομμάτι δεδομένων όταν αυτά αποθηκεύονται και το αποσυμπιέζουν όταν αυτά ανακτούνται, αυξάνεται η χωρητικότητα των αποθηκευτικών μέσων (σχήμα 1.2).



Σχήμα 1.2 Συμπίεση σε αποθηκευτικό μέσο

- Να προσφέρει τις απλούστερες δυνατές αποδεκτές περιγραφές ενός σήματος και έτσι ελαχιστοποιεί την επακόλουθη πολυπλοκότητα των αλγορίθμων επεξεργασίας σήματος, όπως μετασχηματισμούς και ταξινόμηση.

Αυτή η εργασία ασχολείται με τεχνικές συμπίεσης εικόνας και παρουσιάζει έναν συνδυασμό δύω γνωστών μεθόδων συμπίεσης εικόνας: το Block Truncation Coding και Vector Quantization. Η ανάπτυξη των τεχνικών συμπίεσης δεδομένων και εικόνων τα τελευταία χρόνια έχει οδηγήσει στον σχεδιασμό αποδοτικών κωδικοποιητών και αποκωδικοποιητών, σε διάφορες εφαρμογές όπως: ιατρικές εφαρμογές, βάσεις δεδομένων και τηλεπικοινωνίες, εκμάθηση από απόσταση, τηλεδιάσκεψη σε πραγματικό χρόνο κ.α.

Το status quo σήμερα στην συμπίεση εικόνας είναι γνωστό με το ακρωνύμιο JPEG, το οποίο είναι βασισμένο σε έναν διακριτό μετασχηματισμό συνημίτονου (Discrete Cosine Transform), και για αυτό τον λόγο ανήκει στη γενικότερη κατηγορία των κωδικοποιητών που δουλεύουν στο πεδίο της συχνότητας. Μερικά από τα μειονεκτήματα του JPEG είναι ότι ενεργεί ως ένα low pass φίλτρο, θεωρώντας ότι οι υψηλές συχνότητες είναι όχι τόσο σημαντικές. Αυτή η υπόθεση δεν στέκει εάν η εικόνα έχει απότομες ακμές (sharp edges). Για αυτό το λόγο εξασθενώντας τις υψηλότερες συχνότητες οι βασισμένες στον διακριτό μετασχηματισμό συνημίτονου συναρτήσεις, έχουν αποτελέσματα τα οποία μοιάζουν σαν κυματισμοί που διαδίδονται έξω από τις ακμές. Επιπλέον το υπολογιστικό φορτίο και η πολυπλοκότητα είναι πολύ υψηλά και κάνουν τον αλγόριθμο ακατάλληλο για εφαρμογές πραγματικού χρόνου.

Έχουμε δει ότι ο Block Truncation Coding σε συνδυασμό με το Vector Quantization μπορεί να γίνει η βάση για ένα νέο σύστημα κωδικοποίησης, το οποίο μπορεί να μας απαλλάξει από μερικά από τα παραπάνω προβλήματα. Σ' αυτή την εργασία θα παρουσιαστούν τόσο ο Block Truncation Coding όσο και ο συνδυασμός του με έναν σχεδιασμένο με νευρωνικό δίκτυο Vector Quantizer, ο οποίος αποτελεί την καρδιά του προτεινόμενου συστήματος, το οποίο ανήκει στην γενικότερη κατηγορία των codecs του εργάζονται στο χωρικό πεδίο (spatial domain).

2. Συμπίεση εικόνας

2.1 Συμπίεση αναλογικών και ψηφιακών εικόνων

Η συμπίεση εικόνας αναπαριστά μια αρχική εικόνα σε σειρά από bit (bit stream), έτσι ώστε να μπορούν να μεταδωθούν ή να αποθηκευτούν σε ένα ψηφιακό μέσο και ο αριθμός αυτός των bit να είναι μικρότερος από αυτόν που θα χρειαζόταν εάν αποθηκεύαμε ή μεταδίδαμε την αρχική εικόνα. Εάν η εικόνα είναι αναλογική, θα πρέπει αρχικά να την μετατρέψουμε σε ψηφιακή μορφή.

Στην ψηφιακή επεξεργασία εικόνας, κάθε δείγμα της εικόνας, το οποίο λέγεται pixel, κβαντίζεται σε κάποιον επαρκή σταθερό αριθμό από bits και μετά αποθηκεύεται (ή μεταδίδεται) ψηφιακά.

Στην ιδανική περίπτωση θα θέλαμε η κωδικοποιημένη εικόνα να απαιτεί όσο το δυνατόν λιγότερα bits, για να ελαχιστοποιεί το χώρο που χρειάζεται για να αποθηκευτεί ή τον χρόνο για να μεταδοθεί. Σε μερικές περιπτώσεις θα θέλαμε επίσης η αρχική εικόνα να ανακτείται τέλεια από την κωδικοποιημένη μορφή της. Δυστυχώς όμως αυτοί οι στόχοι δεν είναι πάντα εφικτοί. Για παράδειγμα εάν η αρχική εικόνα είναι μια αναλογική φωτογραφία, τότε είναι αδύνατο να την ξαναδημιουργήσουμε τέλεια από την ψηφιακή της αναπαράσταση, κάτι που είναι ανεξάρτητο από τον αριθμό των bits που θα χρησιμοποιήσουμε. Ωστόσο όπως θα περίμενε κάποιος χρησιμοποιώντας αρκετά bits για την ψηφιακή αναπαράσταση θα έπρεπε να παράγει μια κωδικοποιημένη εικόνα, η οποία οπτικά δεν θα είναι δυσδιάκριτη από την αρχική. Όμως “αρκετά bits” ίσως να είναι πάρα πολλά για τα διαθέσιμα αποθηκευτικά ή τα τηλεπικοινωνιακά μέσα.

Μια ψηφιακή μορφή μιας απλής έγχρωμης εικόνας σε ανάλυση τηλεόρασης περιέχει πληροφορία της τάξης του ενός εκατομμυρίου bytes. Η ανάλυση 35mm απαιτεί δέκα φορές το παραπάνω μέγεθος. Η χρήση ψηφιακών εικόνων συχνά δεν είναι εφαρμόσιμη εξαιτίας του υψηλού κόστους αποθήκευσης ή μετάδοσης, ακόμη και αν συσκευές οι οποίες θα τις χειριστούν είναι ανεκτές από πλευρά κόστους.
Επιπλέον, ίσως να λείπει πληροφορία η οποία μπορεί να είναι σημαντική, ακόμη και αν το ανθρώπινο μάτι δεν μπορεί να δει κάτι τέτοιο. Ο ορισμός της ποιότητας είναι συνεπώς στενά εξαρτημένος με την εφαρμογή. Τυπικά, μια συμπιεσμένη εικόνα όταν

αποδικωποιείται για ανακατασκευή της αρχικής συνοδεύεται με κάποια παραμόρφωση. Ένα βολικό μαθηματικό μέτρο της απόδοσης θα μπορούσε να χρησιμοποιηθεί για να βοηθούσε τους αλγορίθμους συμπίεσης να βελτιστοποιήσουν τον παραγόμενο κώδικα. Όπως επίσης και ένας αντιληπτικά παραγόμενος έλεγχος θα μπορούσε να χρησιμοποιηθεί για επιβεβαίωση (validation) του παραγόμενου κώδικα για μια συγκεκριμένη εφαρμογή.

Η απόδοση ενός αλγορίθμου συμπίεσης μετριέται με την ικανότητα συμπίεσης πού έχει, με την παραγόμενη παραμόρφωση, καθώς και με την πολυπλοκότητα υλοποίησης του. Η πολυπλοκότητα των αλγορίθμων συμπίεσης δεδομένων είναι ιδιαίτερα σημαντικό στοιχείο στις hardware υλοποιήσεις.

2.1.1 Πλεονεκτήματα της συμπίεσης εικόνας

Ας υποθέσουμε τον απλούστερο τρόπο για μετάδοση εικόνων μέσο ενός δικτύου ή αποθήκευσής τους. Ο προφανέστερος τρόπος για να σταλεί μια εικόνα είναι να διαβαστεί γραμμή προς γραμμή και να σταλεί μέσο δικτύου. Αλλά το απαιτούμενο εύρος για αυτό το εγχείρημα είναι πολύ μεγάλο. Για παράδειγμα, μια 256*256 εικόνα με 256 επίπεδα gray-scale για κάθε pixel, απαιτεί περίπου 500 kbits. Εφόσον εφαρμογές βίντεο και τηλεδιασκέψεων χρειάζονται μερικές εικόνες το δευτερόλεπτο, βλέπουμε ότι η συμπίεση είναι απαραίτητη. Επιπλέον μια απλή 800*600 true color εικόνα απαιτεί 1.44 MB και ένα ασυμπίεστο βίντεο με 30 frames το δευτερόλεπτο σε ανάλυση 320*200 true color απαιτεί 57.6 MB χώρο στον δίσκο.

Οι τυπικές εικόνες τηλεόρασης έχουν χωρική ανάλυση προσεγγιστικά 512*512 pixels για κάθε frame. Εάν η ένταση (intensity) κάθε pixel έχει resolution 8 bits με 30 frames/sec έχουμε ένα ρυθμό δεδομένων κοντά στα $60x10^6$ bits/sec. Οι ρυθμοί δεδομένων εξαρτώνται κυρίως από την εφαρμογή και κυμαίνονται από 10^5 bits/frame εώς 10^8 bits/frame ή 10^6 bits/sec έως 10^9 bits/sec αντίστοιχα. Οι υψηλές απαιτήσεις μνήμης ή χωρητικότητας καναλιού για μετάδοση ψηφιακών εικόνων και αποθήκευσής τους κάνουν υποχρεωτική την ύπαρξη τεχνικών συμπίεσης δεδομένων.

Ο βασικός σκοπός της συμπίεσης εικόνας είναι η μετατροπή μιας αρχικής με υψηλό ρυθμό δεδομένων εικόνας σε μία συμπιεσμένη εικόνα με δυαδική κωδικοποίηση, έτσι ώστε η παραγόμενη αυτή εικόνα να έχει την καλύτερη δυνατή πιστότητα. Η πιστότητα μπορεί εκτιμηθεί με ποσοτικά κριτήρια, όπως το μέσο τετραγωνικό σφάλμα, με φαινομενικά κριτήρια όπως η εκτίμηση ποιότητας κάποιου παρατηρητή, ή με στατιστικά τα οποία βασίζονται ελέγχους αποδεκτικότητας για συγκεκριμένες εφαρμογές, όπως για παράδειγμα η ακρίβεια διάγνωσης σε μια ιατρική εφαρμογή. Η συμπίεση εικόνας σκοπεύει να πετύχει την καλύτερη δυνατή ποιότητα για το διαθέσιμο αποθηκευτικό ή τηλεπικοινωνιακό μέσο. Υπάρχουν διάφοροι τομείς στους οποίους η συμπίεση μπορεί να φανεί χρήσιμη:

- Η συμπίεση μειώνει τον ρυθμό δεδομένων. Για παράδειγμα, συμπιεσμένες εικόνες μπορούν να "στριμωχτούν" σε ένα μικρότερο bandwidth και να έχουμε έτσι περισσότερα κανάλια βίντεο σε κάποιο τοπικό δίκτυο.
- Καλές τεχνικές συμπίεσης επιτρέπουν προοδευτική μετάδοση, παράγωντας μια καλή ανακατασκευή της εικόνας καθώς φθάνουν τα bits. Αυτό είναι χρήσιμο για telebrowsing, διαλογή σημαντικών κομματιών της εικόνας και "quick-look" αποφάσεις από απομακρυσμένους αισθητήρες.
- Συστήματα συμπίεσης μπορούν να εκτελέσουν διάφορες λειτουργίες στην επεξεργασία εικόνας, όπως edge detection και halftoning.
- Συστήματα συμπίεσης μπορούν να ενσωματωθούν σε συστήματα απόκρυψης (encryption systems) για ασφάλεια δεδομένων.

Αν και το bandwidth συνεχώς γίνεται μεγαλύτερο σε πολλές εφαρμογές, εξαιτίας της χρήσης των οπτικών ινών, και η αποθήκευση όλο και φθηνότερη, χάρη στα οπτικά και μαγνητικά μέσα, η συμπίεση παραμένει επίκαιρη διότι:

- Θα υπάρχουν πάντα μέσα με σχετικά χαμηλή χωρητικότητα, τα οποία οι άνθρωποι θα χρησιμοποιούν, π.χ. τα μέσα δορυφορικών επικοινωνιών τα οποία δεν μπορούν να χειριστούν επιστημονικό υλικό από απομακρυσμένους αισθητήρες, ή χαμηλού κόστούς και χαμηλού ρυθμού μετάδοσης modem και οι ασύρματοι σε πλοιάρια και βάρκες με μικρές κεραίες.
- Ακόμα και η οπτική ίνα μπορεί να "γεμίσει" γρήγορα με ψηφιακό πολυκαναλικό (multichannel) HDTV.
- Τεράστιες βάσεις δεδομένων εικόνων χρειάζονται browsing σε λογικό χρόνο και καλοί αλγόριθμοι συμπίεσης μπορούν να προσφέρουν τις κατάλληλες δομές δεδομένων για αυτό, με μικρό κόστος σε on-line μνήμη.

Η συμπίεση μπορεί να βοηθήσει στην διαδοχική επεξεργασία σήματος. Ένας λόγος που ψηφιοποιούμε τις εικόνες είναι να επιτρέψουμε την ψηφιακή επεξεργασία σήματος (digital signal processing, DSP) πάνω σε αυτές. Συνηθισμένα παραδείγματα περιέχουν enhancement, classification και ανάλυση σκηνών. Ένα σχήμα συμπίεσης εικόνας συχνά εκτελεί καθήκοντα ενός front end. Λιγότερα bits σημαίνουν απλούστερη επεξεργασία, με την προϋπόθεση ότι τα συστήματα συμπίεσης δε χάνουν σημαντική πληροφορία. Ωστόσο μερικές φορές λειτουργίες DSP μπορούν να ενσωματωθούν σε ένα σύστημα συμπίεσης, παράγοντας ένα "έξυπνο" front-end και μειώνοντας την χρήση της προεπεξεργασίας.

2.2 Τεχνικές συμπίεσης εικόνας

Οι τεχνικές συμπίεσης εικόνας μπορούν να διακριθούν σε δύο κατηγορίες, εάν πάρουμε σαν κριτήριο το αν έχουμε ακριβή ανακατασκευή ή όχι. Τις lossles τεχνικές και τις lossy τεχνικές:

- Lossless συμπίεση ή κωδικοποίηση χωρίς θόρυβο (noiseless coding), αντιστρέψιμη κωδικοποίηση (invertible coding), όπου η αρχική εικόνα μπορεί να ανακτηθεί από την ψηφιακή της απεικόνιση. Η lossless συμπίεση μπορεί να εφαρμοστεί μόνο σε είδη ψηφιακές εικόνες και χρησιμοποιεί τεχνικές κωδικοποίησης μεταβλητού μήκους (variable length coding). Η βασική ιδέα τέτοιων κωδικοποιήσεων είναι η χρησιμοποίηση μεγάλων codewords για απίθανες εισόδους, και μικρά για πιθανά. Οι παραγόμενοι κώδικες είναι ρητά σχεδιασμένοι έτσι ώστε ο μέσος αριθμός των bits για κάθε pixel εισόδου να είναι το μικρότερο δυνατό. Οι περισσότερο γνωστές τεχνικές lossless συμπίεσης είναι αυτές των: Huffman, adaptive Huffman, Ziv-Lempel και αριθμητικοί κώδικες .

  Τυπικοί λόγοι συμπίεσης για lossless κώδικες σε ακίνητες εικόνες φτάνουν μέχρι 4:1. Σε ιατρικές εφαρμογές οι συμπιέσεις είναι από 1.7:1 μέχρι 2:1.

  Η άποψη που επικρατεί ευρέως είναι ότι η lossless κωδικοποίηση είναι απαραίτητη στις περισσότερες εφαρμογές και για αυτό το μεγαλύτερο μέρος των ερευνών και υλοποίησης αλγορίθμων σε software και hardware έχει στραφεί προς αυτή την κατεύθυνση. Η αναγκαιότητα του μη χάσιμου πληροφορίας είναι προφανής σε μερικές εφαρμογές όπως προγραμμάτων υπολογιστών και αυθαιρέτων Binary αρχείων αλλά δεν είναι και τόσο προφανής η αναγκαιότητα αυτή σε άλλου είδος εφαρμογές. Η lossles κωδικοποίηση δεν είναι δυνατή αν τα αρχικά δεδομένα είναι αναλογικά μια και η συνήθης τεχνική συμπίεσης τέτοιων εισόδων ή scalar quantization είναι από την φύση της lossy. Εάν θέλουμε να έχουμε την λιγότερη δυνατή απώλεια πληροφορίας σε αυτές τις περιπτώσεις χρησιμοποιούμε διαδοχικά έναν scalar κβαντιστή με ένα noiseless codec.

Τα θεμελιώδη όρια της lossless συμπίεσης βρέθηκαν από την εντροπία της πηγής του Shannon. Θα πρέπει να τονιστεί ότι το χάσιμο πληροφορίας είναι αναπόφευκτο εάν το bit rate είναι μικρότερο από την εντροπία της πηγής.

- Lossy συμπίεση, όπου οι αρχικές εντάσεις των pixel δεν μπορούν να ανακτηθούν τέλεια, έχουμε στην περίπτωση της απλής κβαντοποίησης ή A/D μετατροπής. Ο στόχος είναι να ελαχιστοποιήσει μια μέση παραμόρφωση, όπως για παράδειγμα το μέσο τετραγωνικό σφάλμα, για ένα δοσμένο ρυθμό μετάδοσης. Τυπικές τιμές λόγου συμπίεσης κυμαίνονται από 4:1 με 32:1 για ακίνητες εικόνες χρησιμοποιώντας πρόσφατες τεχνικές.

2.2.1 Lossy συμπίεση

Τα συστήματα συμπίεσης που χρησιμοποιούν lossy τεχνικές μπορούν να κατηγοριοποιηθούν σε μερικούς βασικούς τύπους:

- Scalar Quantization (PCM) με κωδικοποίηση εντροπίας (entropy coding). Τα συστήματα αυτά είναι απλά, και δουλεύουν αρκετά ικανοποιητικά όταν χρησιμοποιούνται σε μεγάλο όγκο δεδομένων και σε πηγές χωρίς μνήμη (memoryless sources).
- Predictive coding με χρήση scalar κβαντιστών. Αυτό είναι ένα βασικό διαφορικό PCM (DPCM) σύστημα με ένα ίσως πολύπλοκο predictor. Ο κβαντιστής ενεργεί σε ένα κλειστό loop παίρνοντας την διαφορά μεταξύ της νέα scalar είσοδο και μιας πρόβλεψης της εισόδου βασισμένη σε ψηφιακή κωδικοποίηση που έγινε πριν. Το σύστημα είναι αποδοτικό όταν ο ρυθμός μετάδοσης είναι μεγάλος και υπάρχει μεγάλη συσχέτιση ανάμεσα στα δείγματα.
- Transform και Subband Coding. Εδώ τα διανύσματα εισόδου μετασχηματίζονται σε μια γραμμική μορφή και κβαντίζονται βαθμωτά χρησιμοποιώντας έξυπνο τρόπο διανομής των bits (smart bit allocation). Ο μετασχηματισμός μπορεί να ολοκληρωθεί με πράξεις πινάκων ή με γραμμικό φιλτράρισμα. Η τεχνική είναι

αρκετά δημοφιλής, προσαρμόζεται εύκολα και είναι καλά κατανοητή μετά από αρκετά χρόνια έρευνας και πρακτικής. Υπάρχουν αρκετές παραλλαγές της προσέγγισης αυτής και μερικές από αυτές έχουν τυποποιηθεί (π.χ. CCITT's H.26, JPEG, MPEG video). Οι περισσότερες από αυτές τις τεχνικές παρέχουν υψηλή απόδοση στο 1 bpp σε μονόχρωμες εικόνες. Ο δημοφιλέστερος μετασχηματισμός είναι ο μετασχηματισμός διακριτού συνημίτονου (discrete cosine transform DCT), αλλά έχουν προταθεί και χρησιμοποιηθεί και άλλοι όπως οι Fast Fourier, Hartley, Walsh, Wavelet. Τα subband φίλτρα, όπως τα QMF (quadrature mirror filters), έχουν επίσης αποδειχθεί χρήσιμα για την παραγωγή του μετασχηματισμένου σήματος. Οι μετασχηματισμοί wavelet είναι σχετικά πρόσφατοι αλλά δίνουν αρκετές υποσχέσεις. Σε αυτού του είδους τις κωδικοποιήσεις ο decoder ή ο decompressor πρέπει να αναστρέψει τον μετασχηματισμό ή να ανασυνδέσουν τα ξεχωριστά μέρη ενός αποσυνθεμένου σήματος.

- Fractals. Η fractal συμπίεση εικόνας συνδυάζει υψηλούς λόγους συμπίεσης με γρήγορους χρόνους αποσυμπιέσεις. Ένα fractal είναι μια απείρως μεγενθύσημη εικόνα η οποία παράγεται με ένα μικρό σύνολο εντολών και δεδομένων. Με ένα fractal, όσο περισσότερο κάνεις zoom σε μια εικόνα, τόσο περισσότερες λεπτομέρειες βλέπεις.
- Κώδικες "δεύτερης γενιάς". Αυτή η κατηγορία συστημάτων συμπίεσης λειτουργεί με την εξαγωγή κάποιων σημαντικών χαρακτηριστικών, όπως για παράδειγμα πληροφορίες για edge και texture του σήματος εισόδου και μετά τα κωδικοποιεί. Μερικές τεχνικές χρησιμοποιούν και wavelet αναπαραστάσεις για τα σήματα. Αυτοί οι αλγόριθμοι συχνά παράγουν ενδεχόμενες ή σαφείς ταξινομήσεις για τμήματα της εικόνας.
- Vector Quantization (VQ). Αυτό είναι ένα απλό Shannon μοντέλο για κωδικοποίηση πηγής. Η ιδέα στηρίζεται στη συμπίεση ενός συνόλου από pixels μαζί αντί για καθένα ξεχωριστά κάθε φορά. Γενικά η μέθοδος αυτή περιέχει μετασχηματισμούς οι οποίοι εφαρμόζονται σε blocks και στην συνέχεια scalar κβάντιση σε συντελεστές του block τους οποίους εξάγει ο μετασχηματισμός. Εάν εφαρμοστεί κατευθείαν σε διανύσματα, οι κβαντιστές μπορούν να σχεδιαστούν έτσι ώστε να ελαχιστοποιούν την μέση παραμόρφωση.

2.3 Πεδία εφαρμογής

Η παροχή υποστήριξης για την χρήση και εκμετάλλευση μέσων όπως φωνή και video φαίνεται να επικρατεί στον τομέα ανάπτυξης υπολογιστικών συστημάτων σήμερα. Η έρευνα που έχει γίνει πάνω στην συμπίεση εικόνας έχει δείξει έναν επαρκή βαθμό επιτυχίας, κάτι που επιβεβαιώνεται από το εύρος των εφαρμογών οι οποίες έχουν αναπτυχθεί σήμερα (κυρίως σε συνδυασμό με βάσεις δεδομένων). Η αρχική επιτυχία της επεξεργασίας εικόνας έχουν οδηγήσει σε μια γενική αισιοδοξία όσον αναφορά τα πιθανά πλεονεκτήματα που θα μπορέσει να προσφέρει η έρευνα στο τομέα αυτό στα υπολογιστικά συστήματα του αύριο. Σ' αυτή την παράγραφο θα αναφέρουμε επιγραφικά μερικές τέτοιες εφαρμογές.

- Σύσκεψη σε πραγματικό χρόνο (Real-time conferencing).

  Η εφαρμογή αυτή επιτρέπει σε πολλούς χρήστες να αλληλεπιδρούν ταυτόχρονα μέσο των υπολογιστών τους με τον ίδιο τρόπο σαν να βρισκόταν στο ίδιο δωμάτιο συσκέψεως. Έχοντας την δυνατότητα να παίρνεις μέρος σε πολλαπλές συσκέψεις με συναδέλφους χωρίς να χρειάζεται να αφήσεις το γραφείο επιφέρει πολλά επιπλέον πλεονεκτήματα σε σχέση με τις παραδοσιακές συσκέψεις. Για παράδειγμα έτσι υπάρχει μικρότερη πιθανότητα να χαθούν κάποια σημαντικά μηνύματα, όπως επίσης και έχεις και λιγότερο χάσιμο χρόνου και χρήματος, αφού δεν χρειάζεται να ταξιδεύει κανείς για να συνδιασκεφτεί με συναδέλφους που είναι απομακρυσμένοι γεωγραφικά.

- Εκπαίδευση μέσο απόστασης (Distance Learning).

  Μια σημαντική υπόθεση για χρόνο και προσπάθεια που ξοδεύεται καθημερινά από καθηγητές και μαθητές για μετακίνηση από και προς το σχολείο. Η δυνατότητα για μια αλληλεπιδρών μάθηση μέσο απόστασης θα μπορούσε να ελαττώσει αυτό το χάσιμο χρόνου και επιπρόσθετα προσφέρει και άλλα σημαντικά πλεονεκτήματα σε σχέση με τις παραδοσιακές τεχνικές μάθησης. Η αλληλεπίδραση είναι ένα κρίσιμο θέμα για να δίνει την δυνατότητα στους μαθητές να παρακολουθούν το αντικείμενο συζήτησης με τον εκπαιδευτή τους, παρά το γεγονός ότι μπορεί να τους χωρίζει μια σημαντική απόσταση.

- Ιατρικές εφαρμογές

  Ένα από τα συνηθισμένα πεδία εφαρμογών της επεξεργασίας εικόνας είναι αυτό της ιατρικής. Η επεξεργασία εικόνας επιτρέπει στους γιατρούς να αποθηκεύουν εικόνες που πάρθηκαν από διάφορα ιατρικά όργανα (π.χ. Ultrasound), μαζί με δεδομένα για τον ασθενή και να τα επεξεργαστούν αργότερα. Μέχρι σήμερα αυτό γινόταν συνήθως σε σταθμούς εργασίας. Όμως αυτοί οι σταθμοί είναι ακριβοί και επιπλέον συχνά πολύπλοκοι στον χειρισμό τους.

- Ηλεκτρονικό ταχυδρομείο με χρήση πολυμέσων (Multimedia electronic mail)

  Το ηλεκτρονικό ταχυδρομείο έχει πια ευρέως αναγνωριστεί σαν ένας αποτελεσματικός τρόπος ασύγχρονης επικοινωνίας μεταξύ χρηστών υπολογιστών, γεφυρώνοντας το χάσμα μεταξύ της παραδοσιακής επιστολής και των βασισμένων στο τηλέφωνο τρόπων επικοινωνίας. Εκτός των πλεονεκτημάτων που απορρέουν από την διαλειτουργικότητα που ηλεκτρονικού ταχυδρομείου, θα μπορούσε να γίνει και μια πρόοδος όσον αφορά την εμφάνιση του μηνύματος που θέλουμε να στείλουμε. Συγκεκριμένα χρησιμοποιώντας στοιχεία των πολυμέσων θα μπορούσαμε να συνθέσουμε και να στείλουμε σε άλλους χρήστες του δικτύου μηνύματα που να περιέχουν εικόνες φωνή και video.

- Έλεγχος παραγωγής (production control)

  Ένα από τα πλεονεκτήματα των συστημάτων αυτομάτου έλεγχου παραγωγής με χρήση επεξεργασίας εικόνας, είναι το ότι η ποιότητα των παραγόμενων αντικειμένων είναι περισσότερο συνεπή. Επιπλέον αυτή η αυτοματοποίηση απαλλάσσει τους ανθρώπους από επαναλαμβανόμενων, βαρετών και πολλές φορές επικίνδυνων διαδικασιών. Ένα τέτοιο σύστημα, που μπορεί να χρησιμοποιηθεί για επίβλεψη γραμμών παραγωγής, όπου κάθε διαδικασία επαναλαμβάνεται, το οποίο είναι αρκετά απλό και μπορεί να επιθεωρηθεί οπτικά, είναι μια ενδεχόμενη εφαρμογή αυτού του πεδίου. Ξύλο, δέρμα, ,μέταλλο και πλαστικό μπορούν να επιθεωρηθούν πριν γίνει η παραπέρα επεξεργασία. Αντίστοιχα μια άλλη εφαρμογή είναι η αναγνώριση προτύπων στην οποία απλές μορφές και χρώματα μπορούν να σορταριστούν από ένα ρομπότ.

- Interactive συστήματα εκμάθησης

  Η εκπαίδευση είναι ένα πεδίο όπου η αποτελεσματική απεικόνιση της πληροφορίας είναι κρίσιμη. Η έκφραση “μια εικόνα ισοδυναμεί με χίλιες λέξεις” ισχύει εδώ, όπως και σε κάθε άλλο πεδίο της επεξεργασίας εικόνας. Μάθηση με εικόνες μπορεί να είναι ευχάριστη για κάθε ηλικία. Μια άλλη σημαντική άποψη είναι ότι η αποκτηθείς με μάθηση πληροφορία απομνημονεύεται καλήτερα, εάν έχει παρθεί με αλληλεπίδραση. Η πληροφορία που έχει παρθεί οπτικά είναι πιο ακριβείς και πιο κοντά στην παραγματικότητα. Τέτοιες εφαρμογές συνδιάζουν τις δυνατότητες των βιβλίων και των υπολογιστών σε ένα ολοκληρωμένο περιββάλον και ανοίγει νέους δρόμους και στρατηγικές εκπαίδευσης. Ένα interactive σύστημα εκπαίδευσης μπορεί να “χτιστεί” με την αποθήκευση εικόνών σε συμπιεσμένη μορφή σε έναν υπολογιστή. Τέτοια συστήματα μπορούν να χρησιμοποιηθούν όχι μόνο στην εκπαίδευση, αλλά και στην υποστήριξη τεχνικών στην δουλειά τους. Το σύστημα μπορεί να οδηγήσει έναν τεχνικό να βρει και να επισκευάσει σφάλματα. Τέλος, παρόμοιοι σταθμοί εκμάθησης μπορούν να βοηθήσουν επιστήμονες να βρούν λύσεις σε προβλήματα, π.χ. βοήθεια σε γιατρούς στην διάγνωση και θεραπεία των ασθενών. Εάν έχει προταθεί μια νέα μέθοδος θεραπείας, τα interactive συστήματα μπορούν να βοηθήσουν στην εκμάθησή της.

- Βάσεις δεδομένων με εικόνες

  Με την χρήση της συμπίεσης εικόνας μπορούμε να αποθηκεύσουμε εικόνες με επαγγελματική ποιότητα και να δημιουργήσουμε βάσεις δεδομένων με εικόνες, οι οποίες μπορούν να αναχθούν οποιαδήποτε στιγμή. Αυτός ο τρόπος είναι πιο ευέλικτος από τις συνηθισμένες βάσεις δεδομένων. Η απεικόνιση των εικόνων στον υπολογιστή σε true color ποιότητα μαζί με κείμενο και γραφικά εκτείνει τις δυνατότητες των βάσεων αυτών. Τέτοιες βάσεις μπορούν να χρησιμοποιηθούν για τουριστικές πληροφορίες από ταξιδιωτικά γραφεία για ενημέρωση των πελατών τους, αν συνδυαζόταν οι εικόνες με χάρτες και άλλα στοιχεία. Άλλες εφαρμογές είναι η εικονική ξενάγηση μουσείων και βάσεις με στοιχεία ανθρώπων που ανήκουν σε κάποια οργάνωση.

- Επικοινωνίες μέσο τηλεφωνικών γραμμών

  Η ψηφιοποίηση των εικόνων μεταξύ άλλων πραγμάτων έχει σαν αποτέλεσμα την μετάδοσή τους μέσο των γραμμών τηλεφώνων. Αυτό σημαίνει ότι η εικόνα μπορεί να σταλεί με ένα modem μεσώ μιας γραμμής τηλεφώνου όπου εμείς θέλουμε. Σε αυτού του είδος των εφαρμογών συμπεριλαμβάνονται τα βιντεοτηλέφωνα, όπου κάποιος μπορεί να βλέπει παράλληλα εικόνα. Αυτά τα βιντεοτηλέφωνα έχουν το πλεονέκτημα ότι οι χρήστες μπορούν να βλέπουν ταυτόχρονα την εικόνα του συνομιλητή τους.

3. Νευρωνικά δίκτυα

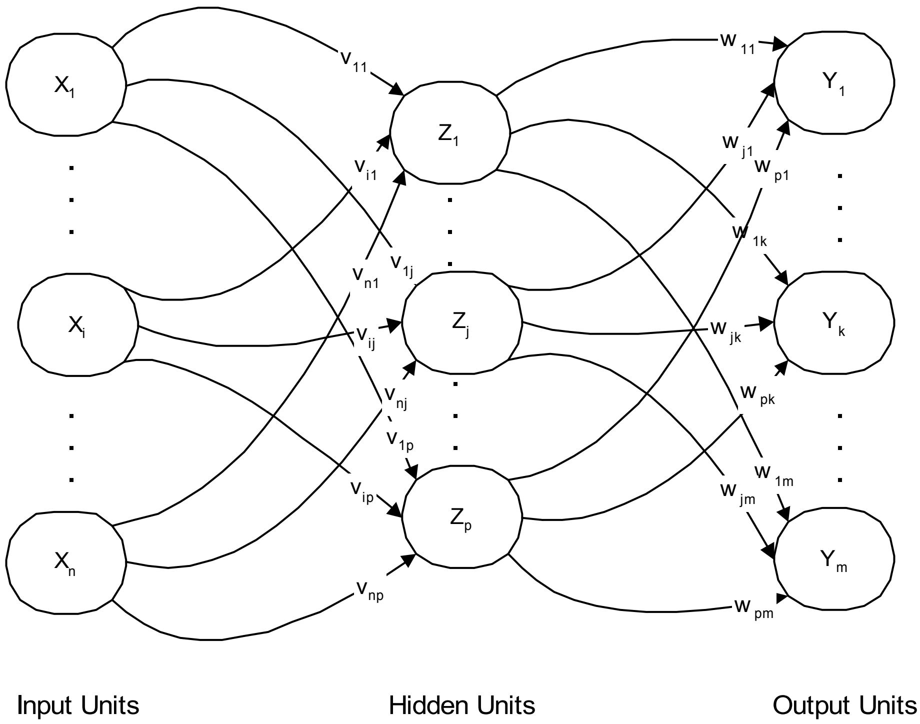
3.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται μια αναφορά κάποιων στοιχείων από τη θεωρία των νευρωνικών δικτύων, όπως επίσης και μερικές εφαρμογές τους στην συμπίεση εικόνας. Συγκεκριμένα, θα δοθεί μια συνοπτική περιγραφή της δομής και των βασικών επιμέρους στοιχείων που αποτελούνται, καθώς και τα γενικά είδη τοπολογίας και αλγορίθμων εκμάθησής τους. Με τον τρόπο αυτό προσδοκούμε να γίνει πληρέστερη η κατανόηση της χρησιμοποίησης των νευρωνικών δικτύων σε προβλήματα όπως αυτό της συμπίεσης εικόνας.

3.2 Ορισμός

Τα νευρωνικά δίκτυα είναι συστήματα επεξεργασίας πληροφορίας. Στην πολύ γενική τους μορφή μπορεί να αναπαρασταθούν σαν συσκευές "black box" οι οποίες δέχονται είσοδο και παράγουν κάποια έξοδο. Στο σχήμα 3.1 φαίνεται ένα τυπικό νευρωνικό δίκτυο. Όπως φαίνεται, αποτελείται από τα στρώματα (layers) Fx, Fy, Fz, καθένα από τα οποία περιέχει μια συλλογή από επεξεργαστικά στοιχεία (proccesing elements) που ονομάζονται νευρώνια. Κάθε ένα από τα νευρώνια δέχεται εισόδους, εκτελεί μια προκαθορισμένη μαθηματική λειτουργία και παράγει μία και μοναδική έξοδο. Το κάθε νευρώνιο λειτουργεί ανεξάρτητα από τα υπόλοιπα. Τα νευρώνια συνδέονται μεταξύ τους με συνδέσεις οι οποίες πολλαπλασιάζονται με κάποιο βάρος. Για λόγους απλότητας οι συνδέσεις αναφέρονται στη βιβλιογραφία ως βάρη. Έτσι, όπως φαίνεται στο σχήμα, το νευρώνιο x_1 συνδέεται με το νευρώνιο z_2 με βάρος v_{21} κ.τ.λ. Οι τιμές που παίρνουν τα βάρη, καθορίζονται από τη διαδικασία εκμάθησης του νευρωνικού δικτύου και η ρύθμισή τους συνεπάγεται την ικανότητα του νευρωνικού δικτύου να μαθαίνει. Άρα μπορεί να ειπωθεί ότι τα βάρη αποτελούν τα στοιχεία μνήμης μιας διάταξης νευρωνίων που μαζί με την είσοδο συντελούν στην παραγωγή της εξόδου.

Συνεπώς οι βασικές λειτουργίες ενός νευρωνικού δικτύου είναι αφ’ ενός η *διαδικασία εκμάθησης* (ρύθμιση των βαρών) και αφ’ ετέρου η *διαδικασία ανάκτησης πληροφορίας* με τη χρησιμοποίηση των ήδη υπαρχόντων βαρών.



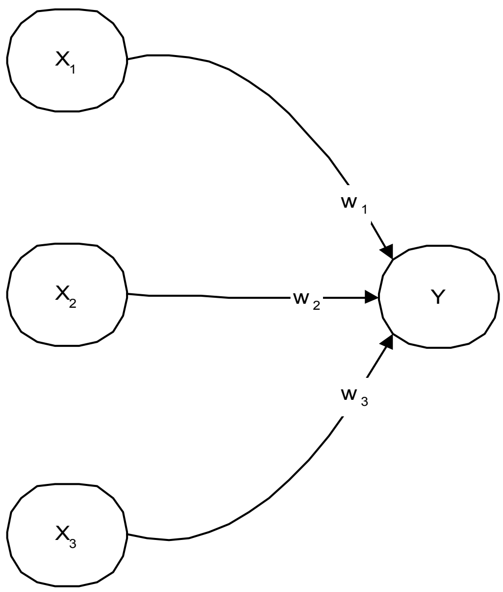
Σχήμα 3.1 Τυπικό Νευρωνικό Δίκτυο

3.3 Συνδέσεις – βάρη

Ένα νευρωνικό δίκτυο είναι ισοδύναμο με ένα κατευθυνόμενο γράφο. Ένας κατευθυνόμενος γράφος έχει ακμές μεταξύ κορυφών οι οποίες επιτρέπουν ροή πληροφορίας μόνο σε μία κατεύθυνση (αυτή που καθορίζεται από το βέλος). Η πληροφορία ρέει μέσω των ακμών και συλλέγεται στις κορυφές. Με την αναπαράσταση αυτή οι ακμές εξυπηρετούν ένα και μόνο σκοπό: δείχνουν τη ροή της πληροφορίας σε ένα γράφο. Τα νευρωνικά δίκτυα επεκτείνουν την αναπαράσταση αυτή με την εισαγωγή ενός βάρους σε κάθε ακμή. Έτσι μία σύνδεση καθορίζει από τη μια τη ροή και από την άλλη μεταβάλει το ποσό πληροφορίας που ρέει μεταξύ των νευρωνίων. Τα βάρη μπορεί να πάρουν θετικές ή και αρνητικές τιμές. Είναι προφανές ότι βάρος με τιμή 0 σημαίνει ανυπαρξία σύνδεσης.

3.4 Νευρώνια(Επεξεργαστικά στοιχεία)

Όπως έχουμε δει, τα νευρώνια αποτελούν τα κομμάτια εκείνα του δικτύου στα οποία γίνεται η επεξεργασία της πληροφορίας. Στο σχήμα 3.2 φαίνεται ο κυριότερος τύπος νευρωνίου. Το νευρώνιο αυτό έχει δύο σημαντικές ιδιότητες:



Σχήμα 3.2 Ένα απλό νευρώνιο

1. Η λειτουργία του απαιτεί μόνο τοπική πληροφορία, δηλαδή όλη η πληροφορία που χρειάζεται το κάθε νευρώνιο την αποκτά από τις εισόδους του και μόνο.
2. Παράγει μία και μόνο μία τιμή εξόδου. Η τιμή αυτή μεταδίδεται διαμέσου των συνδέσεων στα υπόλοιπα νευρώνια ή αποτελεί μέρος της εξόδου του συστήματος

Αυτά τα δύο στοιχεία επιτρέπουν στα νευρώνια να λειτουργούν παράλληλα. Υπάρχουν διάφοροι μηχανισμοί για την εξαγωγή της εξόδου ενός νευρωνικού δικτύου. Όπως δείχνει το σχήμα 3.2 η έξοδος y_j του νευρωνίου j αποτελεί συνάρτηση των εισόδων **X** = (x_1, x_2, x_3) και των βαρών, $\mathbf{W}_j$= (w_1, w_2,w_3). Ισχύει δηλαδή y_j=F(**X**, $\mathbf{W}_j$). Υπάρχουν δύο κύρια είδη συσχετίσεων μεταξύ **X** και **W**:

1. *Γραμμικός συνδυασμός*: αποτελεί το κυριότερο είδος συσχέτισης της εξόδου και των **X** και **W**. Στη βιβλιογραφία αναφέρεται συχνά με το όνομα *NET* και δίνεται από τη σχέση:

$$y_i = f\left(\sum_{i=1}^{n} x_i * w_{ij}\right) \qquad (3.1)$$

2. *Mean - variance*: σε κάποιες περιπτώσεις ένα νευρώνιο μπορει να συνδέεται με δύο συνδέσεις με τα υπόλοιπα νευρώνια αντί για μία.. Μια πιθανή χρήση της δυαδικής αυτής μορφής είναι το νευρώνιο που προτάθηκε από τους Lee και Kil (1988), στο οποίο το ένα βάρος αναπαριστά το μέσο όρο W_j και το άλλο τη μεταβλητότητα V_j ενός συνόλου βαρών. Με το σχήμα αυτό η έξοδος y_j υπολογίζεται από τη σχέση:

3.5 Συναρτήσεις νευρωνίων

Οι *συναρτήσεις νευρωνίων,* που αναφέρονται συχνά στη βιβλιογραφία ως *συναρτήσεις ενεργοποίησης* ή *συναρτήσεις εισόδου-εξόδου*, δέχονται ως είσοδο κάποια από τα παραπάνω είδη συσχέτισης (συνήθως γραμμικός συνδυασμός), και παράγουν έξοδο η οποία ανήκει σε ένα προκαθορισμένο πεδίο τιμών. Οι κυριότερες από αυτές είναι:

1. *Γραμμική:* η γραμμική συνάρτηση παράγει έξοδο που δίνεται από τη σχέση:

$$f(x) = ax \qquad (3.4)$$

2. *Βηματική:* η βηματική συνάρτηση παράγει μόνο δύο τιμές, β και δ. Αν η είσοδος x στη συνάρτηση ισούται ή ξεπερνά μια προκαθορισμένη τιμή θ, τότε η βηματική συνάρτηση παράγει την τιμή β, αλλιώς παράγει την τιμή -δ, όπου β και δ είναι θετικά βαθμωτά μεγέθη. Μαθηματικά η συνάρτηση αυτή περιγράφεται από τη σχέση:

$$f(x) = \begin{cases} \beta & x \geq \theta \\ -\delta & x < \theta \end{cases} \quad (3.5)$$

Για τις τιμές β=1, δ=0 και θ=0, η βηματική συνάρτηση γίνεται:

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (3.6)$$

η οποία χρησιμοποιείται αρκετά στα νευρωνικά δίκτυα, όπως αυτό του Hopfield.

3. *Συνάρτηση ράμπας:* η συνάρτηση ράμπας είναι συνδυασμός γραμμικής και βηματικής συνάρτησης. Η μαθηματική περιγραφή της είναι:

$$f(x) = \begin{cases} \gamma, & x \geq \gamma \\ x, & |x| < \gamma \\ -\gamma, & x \leq -\gamma \end{cases} \quad (3.7)$$

4. *Σιγμοειδής*: η συνάρτηση αυτή είναι στην ουσία η συνεχής έκδοση της συνάρτησης ράμπας. Η πιο κοινή της μορφή είναι:

$$f(x) = \frac{1}{1+e^{-ax}} \quad (3.8)$$

όπου *a*>0 (συνήθως *a*=1). Η συνάρτηση αυτή χρησιμοποιείται ευρέως στη στατιστική. Όταν το *a* γινεται άπειρο, προκύπτει η βηματική συνάρτηση. Δύο άλλες εναλλακτικές μορφές της είναι:

(α) Η συνάρτηση της υπερβολικής εφαπτομένης:

$$f(x) = \tanh(x) \quad (3.9)$$

(β) Η ακόλουθη συνάρτηση:

$$f(x) = \begin{cases} \frac{x^2}{1+x^2}, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (3.10)$$

Η τιμή της (3.9) κυμαίνεται μεταξύ -1 και 1, ενώ της (3.10) μεταξύ 0 και 1.

3.6 Ταξινόμηση νευρωνικών δικτύων

Ανάλογα με τα παρακάτω κριτήρια, τα νευρωνικά δίκτυα διακρίνονται:

a. *Στρώματα (layers):* Τα νευρωνικά δίκτυα οργανώνονται σε στρώματα. Σε ένα layer τα νευρώνια είναι παρόμοια κατά δύο αποψεις. Αφ' ενός οι συνδέσεις οι οποίες τροφοδοτούν τα νευρώνια του στρώματος προέρχονται από την ίδια πηγή. Αφ' ετέρου τα νευρώνια του ίδιου layer χρησιμοποιούν τον ίδιο τύπο συνάρτησης. Τα νευρωνικά δίκτυα υποδιαιρούνται σε μονοστρωματικά ή πολυστρωματικά, ανάλογα με τον αριθμό των στρωμάτων του δικτύου.

b. *Intralayer versus interlayer συνδέσεις:* Υπάρχουν δύο είδη συνδέσεων σε ένα νευρωνικό δίκτυο: intralayer και interlayer. Οι συνδέσεις intralayer είναι συνδέσεις μεταξύ νευρωνίων του ιδίου στρώματος, ενώ οι συνδέσεις interlayer είναι συνδέσεις μεταξύ νευρωνίων που ανήκουν σε διαφορετικά στρώματα. Είναι πιθανό να έχουμε νευρωνικά δίκτυα που περιέχουν συνδέσεις του ενός ή και των δυο ειδών.

c. *Feedforward versus feedback (recurrent) δίκτυα:* Όταν ένα νευρωνικό δίκτυο έχει συνδέσεις στις οποίες η πληροφορία ρέει μόνο προς μία κατεύθυνση, χωρίς ανάδραση, τότε το νευρωνικό δίκτυο λέγεται feedforward. Ορίζουμε ως feedback μονοπάτι κάθε μονοπάτι που επιτρέπει στο ίδιο νευρώνιο να είναι επισκέψιμο δυο φορές. Αν υπάρχουν συνδέσεις feedback, τότε το δίκτυο λέγεται αναδρομικό (feedback ή recurrent).

3.7 Εκμάθηση νευρωνικών δικτύων

Όπως έχουμε δει, μια από τις πιο σημαντικές ιδιότητες των νευρωνικών δικτύων είναι η ικανότητα της εκμάθησης. Πιο συγκεκριμένα, στην ορολογία των νευρωνικών δικτύων η εκμάθηση ορίζεται ως η αλλαγή στις τιμές βαρών, η οποία αποσκοπεί στη βαθμιαία σύλληψη της πληροφορίας, η οποία αργότερα είναι διαθέσιμη προς ανάκτηση. Αρκετοί αλγόριθμοι εκμάθησης έχουν αναπτυχθεί. Αυτοί ομαδοποιούνται στις εξής κατηγορίες:

a) *Supervised versus unsupervised learning:* Ένα νευρωνικό δίκτυο λέμε ότι έχει εκπαιδευτεί με αλγόριθμο supervised όταν για την εύρεση των βαρών απαιτείται η ύπαρξη τόσο του διανύσματος εισόδου όσο και του διανύσματος της επιθυμητής εξόδου. Αυτά τα δύο διανύσματα αναφέρονται στη βιβλιογραφία ως *training pair.* Συνήθως ένα νευρωνικό δίκτυο εκπαιδεύεται χρησιμοποιώντας πολλά τέτοια ζεύγη διανυσμάτων. Κατά τη διαδικασία αυτή, εφαρμόζεται ένα διάνυσμα εισόδου, υπολογίζεται η έξοδος του δικτύου, συγκρίνεται με το διάνυσμα εξόδου, υπολογίζεται η διαφορά (σφάλμα) και αλλάζουν τα βάρη βάσει κάποιου αλγορίθμου ο οποίος τείνει να ελαχιστοποιήσει το σφάλμα αυτό. Η όλη διαδικασία συνεχίζεται έως ότου η έξοδος του δικτύου ικανοποιεί τις προδιαγραφές που θέλουμε. Σε αντίθεση με τα παραπάνω, η εκπαίδευση ενός νευρωνικού δικτύου γίνεται με αλγόριθμο unsupervised όταν δεν απαιτείται το διάνυσμα εξόδου για τη ρύθμιση των βαρών. Στην περίπτωση αυτή ο αλγόριθμος αλλαγής βαρών ρυθμίζει τα βάρη ούτως ώστε οι έξοδοι που παράγονται από διανύσματα εισόδου που δε διαφέρουν σημαντικά να είναι ίδιες. Με τον τρόπο αυτό παρόμοιες είσοδοι που παράγουν την ίδια έξοδο, ομαδοποιούνται σε κατηγορίες (κλάσεις). Οι αλγόριθμοι unsupervised χρησιμοποιούνται γενικά για τη λύση προβλημάτων στα οποία δε γνωρίζουμε εκ των προτέρων τη λύση τους (π.χ. προβλήματα πρόβλεψης).

b) *Structural versus temporal learning:* Ένας αλγόριθμος εκμάθησης είναι structural όταν η απόκριση του νευρωνικού δικτύου δεν εξαρτάται από τις προηγούμενες εισόδους και εξόδους. Στην αντίθετη περίπτωση λέγεται temporal. Οι περισσότεροι αλγόριθμοι που χρησιμοποιούνται είναι structural.

c) *Off-line versus on-line learning:* Οι περισσότερες τεχνικές εκμάθησης γίνονται off-line, δηλαδή το δίκτυο εκπαιδεύεται να δίνει λύση προτού χρησιμοποιηθεί για το σκοπό αυτό, Παράδειγμα τέτοιας τεχνικής είναι ο γνωστός αλγόριθμος backpropagation. Από τη στιγμή που γίνει αυτό, τα βάρη για κάθε έξοδο διατηρούνται, και το δίκτυο είναι έτοιμο προς χρήση. Η εκμάθηση off-line learning μειονεκτεί στο γεγονός ότι το δίκτυο παύει να μαθαίνει από τη στιγμή που χρησιμοποιείται προς ανάκτηση πληροφοριών, Για τη λύση του παραπάνω προβλήματος έχουν αναπτυχθεί αλγόριθμοι που λειτουργούν on-line κι έτσι ανά πάσα στιγμή το δίκτυο προσφέρεται για εκμάθηση καινούργιων καταστάσεων. Παρ' όλα αυτά οι αλγόριθμοι αυτοί μειονεκτούν στο γεγονός ότι δε δίνουν τόσο ικανοποιητικές λύσεις σε δύσκολα προβλήματα. Για το λόγο αυτό χρησιμοποιούνται προς το παρόν αλγόριθμοι off-line.

Οι κυριότεροι αλγόριθμοι εκμάθησης που έχουν αναπτυχθεί ως τώρα και χρησιμοποιούνται στην πράξη δίνονται στον πίνακα 3.1, μαζί με διάφορα χαρακτηριστικά όπως:

διάφορα χαρακτηριστικά όπως:

Learning Algorithm	Training Time	On-Line/ Off-Line	Supervised/ Unsupervised	Linear/ Nonlinear	Structural/ Temporal	Storage Capacity
Hebbian learning	Fast	On-line	Unsupervised	Linear	Structural	Poor
Principal component learning	Slow	Off-line	Unsupervised	Linear	Structural	Good
Differential Hebbian learning	Fast	On-line	Unsupervised	Linear	Temporal	Undetermined
Competitive learning	Slow	On-line	Unsupervised	Linear	Structural	Good
Min-max learning	Fast	On-line	Unsupervised	Nonlinear	Structural	Good
Two-layer error correction learning	Slow	Off-line	Supervised	Linear	Both	Good
Multilayer error correction learning	Very slow	Off-line	Supervised	Nonlinear	Both	Very good
Reinforcement learning	Extremely slow	Off-line	Supervised	Nonlinear	Both	Good
Stochastic learning	Extremely slow	Off-line	Supervised	Nonlinear	Structural	Good
Hardwired systems	Fast	Off-line	Supervised	Nonlinear	Structural	Good

Πίνακας 3.1

- Χρόνος εκμάθησης: πόσο περίπου χρόνο χρειάζεται για να γίνει το δίκτυο ικανό να συλλάβει την απαιτούμενη πληροφορία (γρήγορα, αργά, πολύ αργά, εξαιρετικά αργά);
- On-line / off-line: ο αλγόριθμος είναι on-line ή off-line;
- Supervised / unsupervised: ο αλγόριθμος είναι supervised ή unsupervised;
- Γραμμικός / μη γραμμικός: ο αλγόριθμος είναι ικανός για τη σύλληψη μη γραμμικών αντιστοιχίσεων;
- Structural / temporal: ο αλγόριθμος είναι structural ή temporal;
- Χωρητικότητα αποθήκευσης: κατανέμεται αποθηκευτικά η πληροφορία σε σχέση με τον αριθμό των συνδέσεων στο δίκτυο;

Ένας επίσης σηναντικός πίνακας είναι αυτός του πίνακα 3.2 όπου φαίνονται τα διάφορα ως τώρα νευρωνικά δίκτυα ταξινομημένα ανάλογα με το αν είναι supervised / unsupervised και feedforward / feedback, ανάλογα με τον τρόπο που ανακτάται η πληροφορία.

	RECALL INFORMATION FLOW	
Learning	**Feedback**	**Feedforward**
Unsupervised	Hopfield networks (Amari, 1972; Hopfield, 1982) ART1 & ART2 (Carpenter and Grossberg, 1987a, b) Bidirectional associative memory (Kosko, 1988; Simpson, 1990b) Principal component networks (Oja, 1982; Sanger, 1989)	Linear associative memory (Anderson, 1970; Kohonen, 1972) Associative reward-penalty (Barto, 1985) Adaptive heuristic critic (Barto, Sutton and Anderson, 1983) Drive-reinforcement learning (Klopf, 1986) Learning vector quantization (Kohonen, 1984) Fuzzy min-max classifier (Simpson 1991a) Learnmatrix (Steinbuch and Piske, 1963; Willshaw, 1980)
Supervised	Brain-state-in-a-box (Anderson et al., 1977) Neural optimization (Hopfield and Tank, 1985)	Boltzmann machine (Ackley, Hinton and Sejnowski, 1985) Neocognitron (Fukushima, 1988) Avalanche matched filter (Grossberg, 1969; Hecht-Nielsen, 1990) Sparse distributed memory (Kanerva, 1988) Gaussian potential function network (Lee and Kil, 1989) Backpropagation (Werbos, 1974; Parker, 1982, Rumelhart et al., 1986) Perceptron (Rosenblatt, 1962) Probabilistic neural network (Specht, 1990) Cauchy machine (Szu, 1986) ADALINE (Widrow and Hoff, 1960)

Πίνακας 3.2

Βιβλιογραφία

- Laurence, Fausett (1994). *Fundamentals of Neural Networks,* Prentice Hall

- Haykin, Simon (1998). *A comprehensive Foundation,* Prentice Hall

4. Ο Block Truncation Coding αλγόριθμος

4.1 Εισαγωγή

Από την αρχή της χρήσης ψηφιακών μεθόδων επεξεργασίας εικόνας, υπήρχε η επιδίωξη να βρεθούν ικανοποιητικοί τρόποι κωδικοποίησης της. Καθώς ο αριθμός των bits που χρειάζονται για να διατηρηθεί η δυναμική περιοχή και η ανάλυση μιας τυπικής PCM εικόνας κυμαίνεται από 10^6 ως 10^8, έγινε εφικτό να μεταδοθεί μια τέτοια εικόνα μέσο ενός κοινού κανάλι επικοινωνίας.

Μια εκτεταμένη μελέτη μιας μεθόδου κβαντισμού, η οποία χρησιμοποιεί την ιδέα της διατήρησης απόκλισης στατιστικών δειγμάτων (preserving moments) σαν ένα κριτήριο σχεδιασμού του κβαντιστή(quantizer), έγινε από τον Delp και εφαρμόστηκε σε ακίνητες εικόνες (still images) από τους Delp και Mitchell [1], και σε κινούμενες εικόνες από τους Healy και Mitchell [2]. Μια σημαντική εφαρμογή της moment-preserving κβαντοποίησης, η οποία είναι γνωστή ως Block Truncation Coding ή BTC, αναπτύχθηκε από το Purdue University το 1979. Μια πολυεπίπεδη έκδοση του BTC με κριτήριο το minimax σφάλμα αναπτύχθηκε από τους Wu και Coll με ικανοποιητικά αποτελέσματα [3].

Αυτή η τεχνική κωδικοποίησης εικόνων δουλεύει στο χωρικό πεδίο (spatial domain) και χρησιμοποιεί έναν 1-bit κβαντιστή, ο οποίος προσαρμόζεται στις τοπικές περιοχές (local regions) της εικόνας. Σε γενικές γραμμές, ο κβαντιστής είναι τέτοιος, ώστε να διατηρεί τα τοπικά στατιστικά στοιχεία της εικόνας. Επιπλέον η απόδοσή του, η απλότητα του, η μικρή απαίτηση χώρου αποθήκευσης και η σταθερή συμπεριφορά του σε παρουσία θορύβου στο κανάλι, κάνουν αυτή την μέθοδο κωδικοποίησης πολύ ελκυστική για μετάδοση εικόνων. Ο βασικός αλγόριθμος για αυτή την κωδικοποίηση πετυχαίνει ικανοποιητική ανοικοδόμηση (reconstruction) της εικόνας σε ρυθμό δεδομένων 2 bpp, χρησιμοποιώντας μια απλή κλειστή μορφή προσαρμοζόμενου κβαντιστή, ο οποίος λειτουργεί σε δυσδιάστατα 4x4 κομμάτια εικονοστοιχείων (pixel blocks).

Σ‘ αυτό το κεφάλαιο, ο BTC αλγόριθμος θα παρουσιαστεί και θα συγκριθεί με τις τεχνικές συμπίεσης εικόνας του προηγουμένου κεφαλαίου. Επίσης εξετάζεται η απόδοση του BTC αλγορίθμου από πλευράς θορύβου, όπως επίσης παραλλαγές του βασικού BTC (standard BTC) που μειώνουν τον ρυθμό δεδομένων (bit rate) και βελτιώνουν την ποιότητα της εικόνας (image quality). Σε γενικές γραμμές, στην παράγραφο 4.2 παρουσιάζεται ο βασικός BTC αλγόριθμος, ενώ στην 4.3 εξετάζονται άλλα σχήματα με μη-παραμετρικούς κβαντιστές (non-parametric quantizer schemes), που σε μερικές περιπτώσεις παράγουν καλύτερη απόδοση. Στην παράγραφο 4.4 παρουσιάζεται μια απλή μέθοδο, η οποία μειώνει το bit rate του βασικού αλγορίθμου χρησιμοποιώντας τον πλεονασμό του bit plane. Εκτιμήσεις της απόδοσης του standard BTC και σύγκριση με άλλες τεχνικές φαίνονται στην παράγραφο 4.5.

4.2 Ο βασικός BTC αλγόριθμος

Ο BTC αλγόριθμος χρησιμοποιεί έναν 1-bit μη-παραμετρικό κβαντιστή, ο οποίος προσαρμόζεται στις τοπικές ιδιότητες της εικόνας. Ο κβαντιστής είναι τέτοιος έτσι ώστε να διατηρεί τα τοπικά sample moments. Αυτός ο κβαντιστής παράγει εικόνες καλής ποιότητας σε ρυθμούς δεδομένων 2 bpp, ενώ δεν απαιτεί μεγάλη αποθήκευση δεδομένων και η υπολογισμοί που πρέπει να γίνουν είναι σχετικά λίγοι.

Σε γενικές γραμμές, ο βασικός αλγόριθμος παίρνει έναν πεπερασμένο αριθμό από samples και προσπαθεί να διατηρήσει τα πρώτα δύο samples moments, με το να προσδιορίσει ένα δύο επιπέδων κβαντιστή (two-level quantizer), ο οποίος σχεδιάζεται για αυτό το σκοπό. Οι εξισώσεις του κβαντιστή είναι σχετικά απλές και ο αλγόριθμος δείχνει σταθερή συμπεριφορά σε περιπτώσεις θορύβου στο κανάλι. Επίσης δίνει καλές ανακατασκευασμένες εικόνες, αφού διατηρεί τα τοπικά χαρακτηριστικά των χωρικών κομματιών της εικόνας, κάτι σημαντικό για τον ανθρώπινο παρατηρητή.

Η μέθοδος διαιρεί μια εικόνα σε nxn κομμάτια εικονοστοιχείων (pixel blocks) και για την μελέτη που γίνεται σ' αυτήν την εργασία η εικόνα θα διαιρείται σε 4x4 pixel blocks. Εάν κάποιος χρησιμοποιήσει για τον δύο επιπέδων κβαντιστή τον κλασσικό σχεδιασμό κβαντισμού του Max, ο οποίος ελαχιστοποιεί το μέσο σφάλμα (mean square error), πρέπει a priori να ξέρει την συνάρτηση συχνότητας πιθανότητας (probability density function) των pixelsσε κάθε block. Η ίδια γνώση απαιτείται επίσης για το απόλυτο σφάλμα πιστότητας κριτήριο (absolute error fidelity criteria) του Kassam. Εφόσον στην γενική περίπτωση δεν είναι δυνατό να βρούμε ικανοποιητικά μοντέλα συνάρτησης πυκνότητας, χρησιμοποιήσαμε μη-παραμετρικούς κβαντιστές για τα σχήματα κωδικοποίησης.

Ας δούμε τώρα πιο λεπτομερώς τον BTC αλγόριθμο: αφού διαιρέσουμε την εικόνα σε μη επικαλυπτόμενα 4x4 pixel blocks, κάθε block κωδικοποιείται ανεξάρτητα σε ένα δύο επιπέδων σήμα (1 bpp). Τα επίπεδα επιλέγονται έτσι ώστε να διατηρούν τα δύο πρώτα moments του block, υπολογίζοντας για καθένα από αυτά τα blocks την μέση τιμή (sample mean) και την τυπική απόκλιση (sample standard deviation).

Ας ορίσουμε m=n^2 και x_1, x_2, …, x_m οι αρχικές τιμές σε κάποιο δοσμένο block της αρχικής εικόνας. Τότε το πρώτο και δεύτερο sample moments και η τυπική απόκλιση είναι αντίστοιχα

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

$$\overline{x^2} = \frac{1}{m}\sum_{i=1}^{m} x_i^2$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} x_i^2 - \bar{x}^2 = \overline{x^2} - \bar{x}^2 \qquad (4.1)$$

Όπως με κάθε σχεδιασμό οποιουδήποτε 1-bit κβαντιστή, βρίσκουμε ένα όριο x_{th} και δύο επίπεδα εξόδου, a και b, έτσι ώστε

$$output = \left\{ \begin{matrix} \mathrm{a\ if\ x_i} < \mathrm{x_{th}} \\ \mathrm{b\ if\ x_i} < \mathrm{x_{th}} \end{matrix} \right\} \qquad \mathrm{for\ i = 1,2,...m} \qquad (4.2)$$

Για τον πρώτο κβαντιστή θέτουμε $x_{th} = \bar{x}$. Επιθυμώντας την διατήρηση των $\bar{x}$ και $\overline{x^2}$ τα επίπεδα εξόδου a και b βρίσκονται λύνοντας τις εξισώσεις (4.3) όπου q είναι ο αριθμός των x_i’s μεγαλύτερα από το x_{th}:

$$m\bar{x} = (m - x)a + qb$$

και

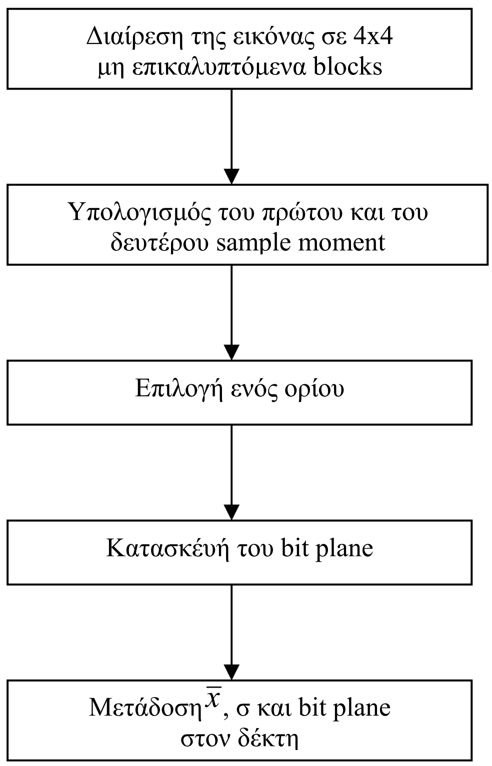
$$m\overline{x^2} = (m - q)a^2 + qb^2 \qquad (4.3)$$

Λύνοντας a για και b:

$$a = \bar{x} - \sigma\sqrt{\left[\frac{q}{m-q}\right]}$$

$$b = \bar{x} - \sigma\sqrt{\left[\frac{m-q}{q}\right]} \qquad (4.4)$$

Κάθε block έτσι περιγράφεται από τις τιμές των $\bar{x}$, σ και ένα nxn bit plane το οποίο περιέχει άσσους στα σημεία οπού $x_i \geq \bar{x}$ και μηδενικά στις άλλες περιπτώσεις. Έτσι, για κάθε block μεταδίδονται το $\bar{x}$, το σ και ένα bit plane (1bit per pixel με το οποίο επιλέγεται το επίπεδο a ή b του κβαντιστή). Χρησιμοποιώντας 4x4 pixel blocks και οχτώ bits για κάθε $\bar{x}$ και σ, πετυχαίνουμε ρυθμό δεδομένων 2 bpp. Στον δέκτη το block ανακατασκευάζεται, υπολογίζοντας τα a και b από την εξίσωση (4.4) με την πληροφορία που έχει για την μέση τιμή και την τυπική απόκλιση, τα οποία μεταδοθήκανε, και αναθέτοντας αυτές τις τιμές,a και b, στα pixels των blocks σε συμφωνία με το bit plane. Μια περιληπτική σύνοψη του βασικού BTC αλγορίθμου φαίνεται σε block diagram στο σχήμα 4.1.



Σχήμα 4.1 Ο βασικός BTC αλγόριθμος

Για παράδειγμα, υποθέτουμε ότι δίνεται το παρακάτω 4x4 block:

$$x_{ij} = \begin{bmatrix} 121 & 114 & 56 & 47 \\ 37 & 200 & 247 & 255 \\ 16 & 0 & 12 & 169 \\ 43 & 5 & 7 & 251 \end{bmatrix}$$

έτσι

$$\bar{x} = 98.75$$
$$\bar{\sigma} = 92.95$$
$$q = 7$$

και

$$a = 16.7 \cong 17$$
$$b = 204.2 \cong 204.$$

Το bit plane είναι τότε

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Το ανακατασκευασμένο block τότε γίνεται

$$\begin{bmatrix} 204 & 204 & 17 & 17 \\ 17 & 204 & 204 & 204 \\ 17 & 17 & 17 & 204 \\ 17 & 17 & 17 & 204 \end{bmatrix}$$

και παρατηρούμε ότι έχουμε διατηρήσει την μέση τιμή και την απόκλιση.

Παραδείγματα αυτού του συστήματος κωδικοποίησης δείχνονται στο σχήμα 4.2. Τα όρια των blocks δεν είναι εμφανή όπως επίσης και ο θόρυβος του κβαντιστή στα μέρη της εικόνας όπου υπάρχει μικρή διακύμανση στην φωτεινότητα (luminance). Εδώ τα επίπεδα του κβαντιστή είναι αρκετά κοντά. Αυτό συμβαίνει μόνο στις περιπτώσεις όπου υπάρχει μεγάλη διαφορά στην φωτεινότητα. Αλλά είναι γνωστό ότι μεγάλες αλλαγές στην φωτεινότητα κρύβει (mask) θόρυβο. Συνεπώς η BTC κωδικοποίηση χρησιμοποιεί αυτή την masking ιδιότητα στην ανθρώπινη όραση. Στα επόμενα κεφάλαια θα δούμε μεθόδους για βελτίωση της ποιότητας της εικόνας και μείωσης του bit rate.

Σχήμα 4.2 Αποτελέσματα του βασικού BTC αλγορίθμου.(Οι αρχικές εικόνες (πάνω) έχουν ανάλυση 256x256 pixels. Οι κωδικοποιημένες (κάτω) έχουν data rate 2 bits/pixel)

4.3 Άλλα μη-παραμετρικά σχήματα κβαντιστή

Υπάρχουν πολλές παραλλαγές του βασικού BTC αλγορίθμου, οι οποίες σε μερικές περιπτώσεις έχουν καλύτερη απόδοση. Μερικές από αυτές τις παραλλαγές διατηρούν τρία moments αντί για δύο, ενώ άλλες διατηρούν την ν-ιοστή και την 2η sample moments, παράγοντας καλύτερα αποτελέσματα. Άλλες πάλι μέθοδοι διατηρούν απόλυτα moments αντί για standard [4], έχοντας απλούστερους υπολογισμούς και μικρότερο μέσο σφάλμα (mean square error). Αναφορά για αυτές τις μεθόδους που διαφοροποιούνται από τον βασικό BTC αλγόριθμο θα γίνει σ’ αυτή την παράγραφο, όπως επίσης και στην χρήση των κριτηρίων πιστότητας του ελάχιστου τετραγωνικού σφάλματος (minimum Mean Square Error, MSE) και του ελάχιστου απόλυτου σφάλματος (minimum Mean Absolute Error, MAE) στην σχεδίαση δύο επιπέδων μη-παραμετρικών κβαντιστών.

Για να χρησιμοποιήσει κάποιος το MSE κριτήριο πιστότητας, πρέπει να ταξινομήσει σε αύξουσα σειρά τα $\bar{x}_i$'s και να κατασκευάσει το ιστόγραμμα των $\bar{x}_i$'s. Ας ορίσουμε $\mathrm{Y}_1, \mathrm{Y}_2, ..., \mathrm{Y}_\mathrm{m}$ τα ταξινομημένα x_i's ($Y_1 \le Y_2 ... \le \mathrm{Y}_\mathrm{m}$). Ας ορίσουμε επίσης q = ο αριθμός των $\bar{x}_i$'s που είναι μεγαλύτερα από το x_{th}, όπως και στην προηγούμενη παράγραφο. Τότε τα a και b βρίσκονται ελαχιστοποιόντας την εξίσωση

$$D_{MSE} = \sum_{i=1}^{m-q-1}(Y_i - a)^2 + \sum_{i=m-q}^{m}(Y_i - b)^2 \tag{4.5}$$

όπου

$$a = \frac{1}{m-q}\sum_{i=1}^{m-q-1} Y_i$$
$$b = \frac{1}{q}\sum_{i=m-q}^{m} Y_i \tag{4.6}$$

Για να λυθεί αυτό το πρόβλημα, ο αλγόριθμος πρέπει να προσπαθήσει με κάθε πιθανό όριο (υπάρχουν το πολύ m-1 όρια) και να διαλέξει αυτό με το μικρότερο D_{MSE}. Υποθέτοντας πως τα a και b είναι οχτώ bit το καθένα όπως και προηγουμένως έχουμε έναν ρυθμό δεδομένων 2 bpp.

Η χρήση του κριτηρίου πιστότητας MAE είναι παρόμοια με αυτή του MSE. Μόνο πού σε αυτή την περίπτωση για να βρεθούν τα a και b ελαχιστοποιείται η παρακάτω εξίσωση

$$D_{MAE} = \sum_{i=1}^{m-q-1}|Y_i - a| + \sum_{i=m-q}^{m}|Y_i - b| \tag{4.7}$$

όπου

$$\mathrm{a} = \text{mean of}\,(\mathrm{Y}_1, \mathrm{Y}_2, ..., \mathrm{Y}_{\mathrm{m\text{-}q\text{-}1}})$$
$$\mathrm{b} = \text{mean of}\,(\mathrm{Y}_{\mathrm{m-q}}, ..., \mathrm{Y}_\mathrm{m}) \tag{4.8}$$

Όπως έχει προβλεφθεί, ο MSE κβαντιστής έχει το μικρότερο υπολογιζόμενο μέσο τετραγωνικό σφάλμα και ο MAE κβαντιστής το μικρότερο μέσο απόλυτο σφάλμα. Η απόδοση του BTC συγκρινόμενη με αυτά τα κριτήρια πιστότητας είναι αρκετά καλή. Το πλεονέκτημα της χρήσης ενός μη-παραμετρικού κβαντιστή είναι ότι η φόρμουλα του κβαντιστή είναι σε κλειστή μορφή. Αυτό απλοποιεί σημαντικά το υπολογιστικό φορτίο του.

4.4 Ο AMBTC αλγόριθμος (Absolute Moment BTC)

Από την αρχική ιδέα που χρησιμοποιείται στον BTC, η οποία είναι η επιτυχία συμπίεσης με την διατήρηση sample moments, υπάρχουν αρκετές άλλες παραλλαγές, σαν την μέθοδο που θα παρουσιαστεί στην παράγραφο αυτή, η οποία διατηρεί απόλυτα moments αντί για standard moments και η οποία καλείται Absolute Moment Block Truncation Coding ή AMBTC.

Η μέθοδος αυτή διαιρεί την εικόνα σε 4x4 μη-επικαλυπτόμενα pixel blocks όπως και ο κανονικός BTC. Κάθε block κβαντίζεται σε δύο επίπεδα (1 bpp). Τα επίπεδα διαλέγονται έτσι ώστε να διατηρούν το sample mean και το κεντρικό sample first absolute moment του αρχικού block. Ας ορίσουμε πάλι $m=n^2$ τον συνολικό αριθμό των pixels στο block, και x_i την τιμή κάθε pixel. Συνεπώς είναι απαραίτητο να υπολογιστούν το sample mean

$$\overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{4.9}$$

και το sample first absolute central moment

$$\alpha = \frac{1}{m}\sum_{i=1}^{m} \left|x_i - \overline{x}\right| \tag{4.10}$$

Η μέση τιμή περιέχει πληροφορία σχετικά με την κεντρική κλίση (tendency). Είναι η ίδια πληροφορία που χρησιμοποιήθηκε από τους Delp και Mitchell στον αρχικό BTC [1]. Από την άλλη μεριά το sample first absolute central moment περιέχει πληροφορία σχετικά με την διασπορά της μέσης τιμής. Η αντίστοιχη τιμή στον standard BTC είναι η τυπική απόκλιση. Άρα δύο από τα πιο σημαντικά τοπικά χαρακτηριστικά των χωρικών gray level blocks διατηρούνται: η κεντρική διασπορά και η απόκλιση από το κέντρο.

Για τον AMBTC αλγόριθμο, οι τιμές των a και b δίνονται αντίστοιχα από [4]:

$$\begin{aligned} a &= \overline{x} - \frac{\gamma}{m-q} \\ b &= \overline{x} + \frac{\gamma}{q} \end{aligned} \tag{4.11}$$

όπου

$$\gamma = \frac{m\alpha}{2} = \sum_{\text{for } x \geq \overline{x}} x_i - \overline{x}q \tag{4.12}$$

Εξετάζοντας την (4.11) βλέπουμε ότι για την περίπτωση όπου όλα τα pixels έχουν την ίδια gray τιμή, η πληροφορία απόκλισης είναι ίση με το μηδέν και τα a και b παίρνουν την τιμή της μέσης τιμής, η οποία είναι η σωστή τιμή για κάθε pixel. Σε άλλη περίπτωση που τα pixels είναι διασκορπισμένα από την μέση τιμή, η τιμή που παίρνει το b είναι η μέση τιμή συν μια στατιστική απόκλιση, η οποία εξαρτάται απευθείας από την διασκόρπιση και αντίστροφα από τον αριθμό των pixels που έχουν τιμή πάνω από την μέση τιμή.

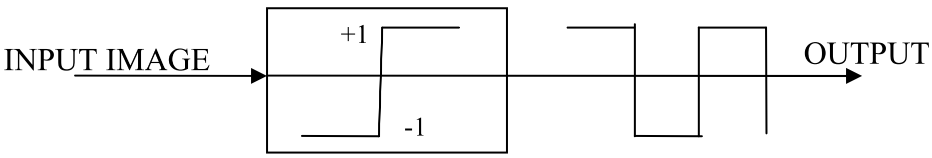
O AMBTC και ο BTC εφαρμόστηκαν σε πραγματικές εικόνες στο [26]. Όλες οι εικόνες κβαντίστηκαν χρησιμοποιώντας 4x4 blocks, 6 bits για κάθε μέση τιμή, 4 bits για κάθε first absolute central moment πετυχαίνοντας bit rate 1.625 bpp. Το μέσο τετραγωνικό σφάλμα στον AMBTC είναι μικρότερο αλλά η κυριότερη παρατήρηση είναι η εμφάνιση σε μερικές περιπτώσεις ακανόνιστων ακμών όπως επίσης και μη-ικανοποιητική απεικόνιση ορισμένων μεσαίας απόστασης τιμών που οφείλεται στην ανάθεσή τους σε μεγάλη ή μικρή τιμή.

O AMBTC είναι γρηγορότερος του BTC, επειδή στον BTC είναι απαραίτητος ο υπολογισμός ενός αθροίσματος από m τιμών(βλέπε (4.1)), η καθεμιά υψωμένη στο τετράγωνο, ενώ στην περίπτωση του AMBTC είναι μόνο απαραίτητος ο υπολογισμός του αθροίσματος από τιμές που δίνονται στην (4.10). Εφόσον ο χρόνος που απαιτείται για κάποιον πολλαπλασιασμό στους περισσότερους επεξεργαστές είναι αρκετές φορές μεγαλύτερος από αυτόν της πρόσθεσης, χρησιμοποιώντας τον AMBTC ο συνολικός χρόνος στον πομπό είναι σημαντικά μικρότερος. Κάτι παρόμοιο συμβαίνει και στον δέκτη. Για τον standard BTC είναι απαραίτητο να εκτιμήσουμε δύο πηλίκα και δύο τετραγωνικές ρίζες (βλέπε (4.4)), ενώ στον AMBTC μόνο δύο πηλίκα (βλέπε (4.11)). Συνεπώς και εδώ έχουμε μείωση του χρόνου επεξεργασίας σε πομπό και δέκτη. Τα υπόλοιπα χαρακτηριστικά του AMBTC παραμένουν τα ίδια με αυτά του BTC καθώς η φιλοσοφία και των δύο μεθόδων είναι ουσιαστικά η ίδια. Μερικά παραδείγματα αυτής της παραλλαγής του BTC βλέπουμε στο σχήμα 4.3.

Σχήμα 4.3 Αποτελέσματα του AMBTC αλγορίθμου.(Οι αρχικές εικόνες (πάνω) έχουν ανάλυση 256x256 pixels. Οι κωδικοποιημένες (κάτω) έχουν data rate 1.625 bits/pixel)

4.5 Κωδικοποίηση του image bit plane

Πολλές εφαρμογές όπως για παράδειγμα μετάδοση video σε πραγματικό χρόνο, απαιτούν ρυθμούς μετάδοσης της τάξης του 1bpp. Δυστυχώς ο απλός BTC αλγόριθμος με ρυθμό μετάδοσης περίπου 1.63 bpp, δεν πετυχαίνει κάτι τέτοιο με λογική παραμόρφωση. Προσθετικές μειώσεις θα μπορούσαν να γίνουν με αποτελεσματική κωδικοποίηση του bit plane. Όμως δεν μπορούμε να χρησιμοποιήσουμε run length coding ή κάποιο άλλο standard σχήμα εφόσον κάθε binary block πρέπει να σταλεί ξεχωριστά. Σ' αυτή την παράγραφο θα αναλύσουμε μια απλή fixed length μέθοδο κωδικοποίησης, η οποία χρησιμοποιεί τον πλεονασμό του bit plane για να μειώσει τον ρυθμό μετάδοσης. Όπως δείχνεται στο 4.4 η truncation διαδικασία είναι ένα αυστηρός περιοριστικής.



Σχήμα 4.4 Αυστηρός περιοριστής

Γενικά, εάν Δx και Δy είναι οι αποστάσεις μεταξύ των pixels στην x και y διεύθυνση, τα σήματα εικόνων μοντελοποιούνται καθώς έχουν εκθετική συσχέτιση στην μορφή

$$R(\Delta\chi, \Delta y) = e^{-\alpha|\Delta x| - \beta|\Delta y|}$$

Εφόσον εδώ συζητάμε για μονοδιάστατο φιλτράρισμα, το σήμα εισόδου στον αυστηρό περιοριστή έχει μια αυτοσυσχέτιση της μορφής

$$R_{xx}(\Delta\chi) = e^{-\alpha|\Delta x|} \tag{4.13}$$

όπου α κυμαίνεται μεταξύ 0.05 και 0.3 εξαρτώμενο από τα χαρακτηριστικά της εικόνας.

Εάν θεωρήσουμε την εικόνα εισόδου σαν μια συνεχή Gaussian διαδικασία με συσχέτιση όπως δείχνεται στην (4.13), η συσχέτιση του truncated σήματος μπορεί να δειχθεί ότι είναι

$$R_{yy}((\Delta\chi) = \frac{2}{\pi}\sin^{-1}(e^{-\alpha|\Delta x|}) \tag{4.14}$$

Βλέπουμε ότι το bit plane διατηρεί μερική συσχέτιση. Από την (4.14) συμπεραίνουμε ότι

$$e^{-\alpha|\Delta x|} = \frac{\pi}{2}R_{yy} - \frac{\left[\frac{\pi}{2}R_{yy}(\Delta x)\right]}{3!} + \frac{\left[\frac{\pi}{2}R_{yy}(\Delta x)\right]}{5!} - \ldots.$$

Εφόσον η αυτοσυσχέτιση R_{yy} είναι άνω φραγμένη στο 1, μπορούμε να προσεγγίσουμε την σειρά με

$$e^{-\alpha|\Delta x|} \cong \frac{\pi}{2}R_{yy}(\Delta x) - \frac{\left[\frac{\pi}{2}R_{yy}(\Delta x)\right]^3}{3!} \tag{4.15}$$

Για Δx μεγάλο βλέπουμε ότι η αυτοσυσχέτιση της εξόδου είναι ακριβώς η αυτοσυσχέτιση της εισόδου πολλαπλασιασμένη με 2/π. Από (4.13) και (4.14) βρίσκουμε

$$\frac{R'_{yy}(\Delta y)}{R'_{xx}(\Delta x)} = \frac{2}{\pi\sqrt{1-e^{-2\alpha|\Delta x|}}}, \quad \text{for}\,|\Delta \mathrm{x}| > 0 \tag{4.16}$$

Για μικρά Δx η τιμή της αυτοσυσχέτισης εξόδου πέφτει πολύ γρηγορότερα από αυτή της εισόδου. Αυτό σημαίνει ότι ο περισσότερος πλεονασμός του bit plane είναι μεταξύ πολύ κοντινών pixels. Αυτό είναι που απαλείφει η κωδικοποίηση του bit plane με την χρήση median φίλτρων ,όπως θα παρουσιαστεί στην παρακάτω παράγραφο.

4.6 Η μεθοδολογία της κωδικοποίησης με την χρήση median φίλτρων

Σ' αυτή την παράγραφο θα παρουσιάσουμε μια απλή fixed-length μέθοδο κωδικοποίησης, η οποία χρησιμοποιεί τις στατιστικές ιδιότητες των παραγόμενων από τον BTC εικόνων για να μειώσει το bit rate και περικλείει μια απλή μη γραμμική λειτουργία, η οποία καλείται median φιλτράρισμα.

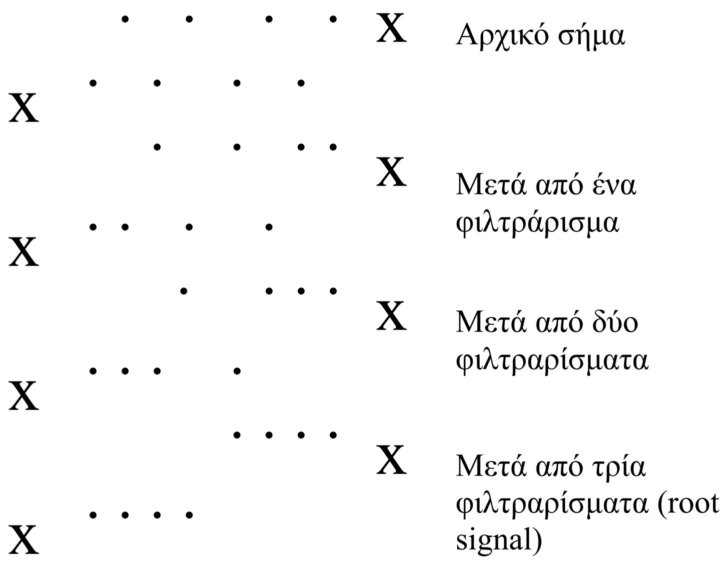
Αρχικά ένα δείγμα (sample) προσθέτεται στην αρχή και το τέλος του σήματος. Οι τιμές των δειγμάτων αυτών είναι ίσες με την πρώτη και την τελευταία τιμή του σήματος αντίστοιχα. Στην συνέχεια μετακινούμε κατά μήκος ολόκληρου του σήματος ένα παράθυρο, το οποίο εκτείνεται σε k σημεία δειγμάτων. Σε κάθε σημείο υπολογίζουμε το μέσο (median) των k γειτονικών δειγμάτων που βρίσκονται μέσα στο παράθυρο και αντικαθιστούμε το κεντρικό δείγμα με το υπολογιζόμενο μέσο. Αυτή η μέθοδος έχει την εξής ιδιότητα: εάν ένα σήμα φιλτραριστεί διαδοχικά, η έξοδος μετά από κάθε πέρασμα του φίλτρου συγκλίνω σε ένα σήμα αμετάβλητο σε επιπλέον median φιλτραρίσματα. Αυτό το αμετάβλητο σε άλλα φιλτραρίσματα σήμα καλείται "root signal set" και είναι κατά πολύ μικρότερο από το δυαδικό χώρο. Ο αριθμός των root σημάτων R_n για ένα φίλτρο με μέγεθος παραθύρου k=3 και μήκος σήματος n σχετίζεται με την σειρά Fibonacci (f_n) ως εξής

$$R_n = 2f_{n+1} \qquad \mathrm{n} \geq 1,$$

όπου

$$f_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

Στην περίπτωσή μας έχουμε μια σειρά από 4x4el block, n=4 και R_4=10 αντί για 16. Ένα παράδειγμα το οποίο δείχνει την μονοδιάστατη διαδικασία του median filtering δείχνεται στο σχήμα 4.3.



Σχήμα 4.5 Παράδειγμα της median filter κωδικοποίησης, όπου X τα επιπρόσθετα δείγματα στην αρχή και το τέλος του σήματος

Η μέθοδος αυτή μείωσης του bit rate του binary plane, φιλτράρει κάθε σειρά του bit plane που παράγεται από τον BTC μέχρι η σειρά αυτή να γίνει root signal. Έτσι αντί να σταλεί στον δέκτη ολόκληρη η σειρά στέλνεται ένας κωδικός που προσδιορίζει την σειρά αυτή. Στην επόμενη παράγραφο θα δούμε ότι σήματα του block έχουν μεγάλη πιθανότητα να είναι κάποιο root σήμα. Εφόσον το σύνολο των root σημάτων περιέχει 10 καταστάσεις αντί για 16 που έχει το δυαδικό πεδίο, μεταδίδουμε λιγότερα bits που αναπαριστούν ένα block και έτσι πετυχαίνουμε συμπίεση. Το σχήμα 4.4 δείχνει ένα παράδειγμα εφαρμογής της κωδικοποίησης με χρήση median φίλτρου σε ένα 4x4 block εικόνας:

10 10 11 12	1 1 1 1	1 1 1 1	12 12 12 12
10 9 12 15	1 0 1 1	1 1 1 1	12 12 12 12
3 4 12 15	0 0 1 1	0 0 1 1	3 3 12 12
2 4 10 15	0 0 1 1	0 0 1 1	3 3 12 12
Αρχικό block	BTC block	Φιλτραρισμένο block	Ανακατασκευασμένο block

Σχήμα 4.6 Median φιλτράρισμα: $\bar{x}$=9.6,σ=4.2, a=3 και b=12

Χρησιμοποιώντας το πλεονέκτημα που μας δίνει η στατιστική φύση του bit plane και αφαιρώντας τον πλεονασμό με median filtering, μπορούμε να δούμε ότι απαιτούνται μόνο 12 bits (3 bits για να καθορίσουμε τα οχτώ πιθανά roots για κάθε γραμμή όπως θα δούμε αναλυτικά παρακάτω) για να καθορίσουμε το bit plane. Έτσι μειώνουμε τον ρυθμό μετάδοσης από 1.63 σε 1.31 bpp. Μπορεί ακόμη να δειχθεί ότι ένα πέρασμα του φίλτρου είναι αρκετό για να φτάσουμε από το σήμα εισόδου σε κάποιό root σήμα. Μερικά παραδείγματα κωδικοποίησης με χρήση median filtering με data rates 1.63 και 1.31 bpp δίνονται στα σχήματα 4.7 και 4.8 αντίστοιχα.

Σχήμα 4.7 Αποτελέσματα εικόνων με χρήση media filtering (data rate 1.625 bits/pixel).

Σχήμα 4.8 Αποτελέσματα εικόνων με χρήση media filtering (data rate 1.31 bits/pixel).

4.7 Μαθηματική παρουσίαση της Truncation διαδικασίας

Όπως δείχνεται και στο σχήμα 4.2, τα σήματα του bit plane μοντελοποιούνται ως γκαουσιανά σήματα αυστηρού περιοριστή. Και σαν τέτοια σήματα προσεγγίζονται με ευρείας έννοιας μαρκοβιανές (wide sense Markovian). Παρατηρούμε ότι η κατανομή πιθανότητας για κάθε ένα από τα σήματα του bit plane είναι στενά συσχετισμένη με την κατανομή της διάρκειας των excursions των γκαουσιανών διακυμάνσεων. Ο Rice εξάγει μια προσεγγιστική πυκνότητα πιθανότητας, έχοντας ένα πάνω excursion μήκους τ, δοσμένου ότι η excursion ξεκίνησε από την αρχή. Για μια τέτοια κατανομή, οι μηδενοδιελεύσεις (zero crossing) που περιμένουμε σε ένα πεπερασμένο διάστημα είναι ανάλογες με

$$\frac{1}{\pi}\sqrt{\left[-\frac{R''(\tau)}{R(\tau)}\right]}$$

Εφόσον

$$R''(\tau) = \frac{\partial^2 e^{-\alpha|\tau|}}{\partial \tau^2} \rightarrow \infty \qquad \text{as } \tau \rightarrow 0,$$

ο αναμενόμενος αριθμός των crossings δεν φράζεται. Για αυτό τον λόγο μπορούμε να θεωρήσουμε την εικόνα σαν ένα τυχαία διαδικασία αντί για ένα συνεχές σήμα.

Για τους σκοπούς της ανάλυσής μας θα δηλώνουμε τα δείγματα από τις γραμμές εισόδου του 4x4 pixel block με $\{f_n, f_{n+1}, f_{n+2}, f_{n+3}\}$ και οι αντίστοιχες δυαδικές τους μορφές με $\{s_n, s_{n+1}, s_{n+2}, s_{n+3}\}$. Εάν τα τέσσερα δείγματα της εικόνας αφήνονται να είναι από κοινού κανονικά, παίρνουμε την πιθανότητα ύπαρξης για το root σήμα ολοκληρώνοντας την 4-μεταβλητή κανονικής κατανομής μέσα στα καθοριζόμενα από την σειρά root όρια. Ένα απλό μοντέλο βρίσκεται προσεγγίζοντας το δυαδικό bit plane με μία πρώτης τάξεως διαδικασία Markov, λαμβάνοντας υπόψη μόνο την συσχέτιση μεταξύ πολύ κοντινών γειτόνων όπως δείχνουν οι σχέσεις (4.15)-(4.16) της προηγουμένης παραγράφου. Για το Markov μοντέλο, δεδομένης μιας σειράς “0011” του bit plane και της πιθανότητας ύπαρξής της, μπορούμε α γράψουμε

$$\Pr(s_{n+3}=0, s_{n+2}=0, s_{n+1}=1, s_n=1) =$$
$$= \Pr(s_{n+3}=0 \mid s_{n+2}=0)\Pr(s_{n+2}=0 \mid s_{n+1}=1)\Pr(s_{n+1}=1 \mid s_n=1)\Pr(s_n=1)$$

Έτσι για να πάρουμε την πιθανότητα ύπαρξης κάποιου σήματος του bit plane, χρειαζόμαστε μόνο τις πιθανότητες μετάβασης. Η πιθανότητα αλλαγής μιας bit τιμής μεταξύ δύο συνεχόμενων pixels είναι

$$P_1 = \Pr(s_{n+1} = 0 \mid s_n = 1) = \Pr(s_{n+1} = 1 \mid s_n = 0)$$

και η πιθανότητα να έχουν την ίδια τιμή δύο συνεχόμενα pixels

$$P_0 = 1 - P_1$$

Αν οι τυχαίες μεταβλητές f_n και f_{n+1} είναι συνδεδεμένες κανονικά με συντελεστή συσχέτισης ρ τότε:

$$P_1 = \frac{\beta}{\pi}, \quad \kappa\alpha\iota \quad \mathrm{P}_0 = 1 - \frac{\beta}{\pi}$$

όπου $\beta=\cos^{-1}\rho$ και $\rho=e^{-\alpha}$ ο συντελεστής συσχέτισης.

Για α=0.15, η πιθανότητα μια σειρά να είναι ένα root σήμα είναι Pr (row$\in$root)=0.94 και Pr(row$\notin$root)=0.052. Οι άλλες έξι καταστάσεις του δυαδικού χώρου συμβαίνουν με τις πιθανότητες που δείχνονται στον πίνακα 4.1, ενώ τα root σήματα στον πίνακα 4.2.

Συγκεκριμένα ο πίνακα 4.2 δείχνει κάθε root σειρά με την πιθανότητα ύπαρξής της με αναλυτικά αποτελέσματα (για α=0.2) και με πειραματικά αποτελέσματα από [5].

4.8 Ανάλυση παραμόρφωσης

Θέλοντας να αποτιμήσουμε μια μέθοδο κωδικοποίησης, πρέπει να δούμε αναλυτικά και πειραματικά την παραμόρφωση που εισέρχεται από όλη την διαδικασία .

Ας θεωρήσουμε seq_j,j=1,…,16 μια σειρά εισόδου στο φίλτρο. Η παραμόρφωση που εισέρχεται με το φιλτράρισμα είναι

$$D_f = \sum_{seq_j} \left[\sum_{x_i \in seq_j} (x_i - x_f)^2 \right] \Pr(seq_j) \qquad (4.17)$$

row	$\Pr(root \mid \alpha)$	$\Pr(root \mid \alpha = 0.2)$
0010	$1/2P_1^2P_0$	0.015
0100	$1/2P_1^2P_0$	0.015
0101	$1/2P_1^3$	0.003
1010	$1/2P_1^3$	0.003
1011	$1/2P_1^2P_0$	0.015
1101	$1/2P_1^2P_0$	0.015

Πίνακας 1 Κατανομή πιθανότητας για τις έξι μη-root σειρές

όπου το εσωτερικό μέλος είναι η μέτρηση της παραμόρφωση για την j σειρά και το τελευταίο μέλος δείχνει την πιθανότητα εμφάνισης της j σειράς. Θυμίζουμε ότι με x δηλώνουμε μια pixel τιμή στο 4x4 pixel block και το οποίο παίρνει τιμές 0 και 1. Από τους πίνακες 4.1 και 4.2 μπορεί κάποιος να παρατηρήσει ότι τα μη-root σήματα διαφέρουν από τα rootμόνο κατά ένα bit και εφόσον τα root σήματα δεν εισάγουν παραμόρφωση, η (4.17) μπορεί να γραφεί ως D_f =P(row$\notin$root).

Διαλέγοντας διαφορετικά υποσύνολα μέσα στο σύνολο των root σημάτων, μπορούμε να πετύχουμε μεγαλύτερη συμπίεση δεδομένων με κόστος μεγαλύτερη πιθανότητα παραμόρφωσης. Για παράδειγμα βλέποντας τον πίνακα 2 τα root 4 και 7 είναι λιγότερο πιθανά να εμφανιστούν από ότι τα άλλα. Έτσι εάν διαγράψουμε αυτά τα δύο roots, έχοντας σαν αποτέλεσμα ένα σύνολο από οχτώ σήματα, χρειαζόμαστε μόνο 4x3=12 bits για κωδικοποίηση του bit plane, αντί για 13.4 bits εάν χρησιμοποιούσαμε ολόκληρο το σύνολο των roots.

$root\,\#$	$sequence$	$\Pr(root/\alpha)$	$\Pr(root/\alpha=0.2)$	$\Pr(root/\pi\varepsilon\iota\rho\alpha\mu\alpha\tau\iota\kappa\acute{\alpha})$
1	0000	$1/2P_0^3$	0.261	0.20
2	0001	$1/2P_0^2P_1$	0.063	0.08
3	0011	$1/2P_0^2P_1$	0.063	0.11
4	0110	$1/2P_0P_1^2$	0.015	0.02
5	0111	$1/2P_0^2P_1$	0.063	0.07
6	1000	$1/2P_0^2P_1$	0.063	0.09
7	1001	$1/2P_0P_1^2$	0.015	0.02
8	1100	$1/2P_0^2P_1$	0.063	0.12
9	1110	$1/2P_0^2P_1$	0.063	0.09
10	1111	$1/2P_0^3$	0.261	0.20

Πίνακας 2 Κατανομή πιθανότητας για τα root σήματα

4.9 Συμπεράσματα

Ο BTC, όπως αρχικά προτάθηκε από τους Delp και Mitchell στο [1], παρουσιάζει ορισμένα πλεονεκτήματα, τα οποία κάνουν την μέθοδο κωδικοποίησης αυτή πολύ ελκυστική για μετάδοση εικόνων σε πραγματικό χρόνο.

Τα σημαντικότερα πλεονεκτήματα είναι η εύκολη υλοποίηση του, οι υπολογισμοί είναι σχετικά απλοί, οι απαιτήσεις δεδομένων μικρές, αφού χρειάζεται μόνο τέσσερις γραμμές της εικόνας στην μνήμη, και παράγει ένα σταθερού μήκος (fixed length)κώδικα για κάθε block. Άρα ο αλγόριθμος χρειάζεται μικρότερο υπολογιστικό φορτίο και πολύ γρηγορότερη μνήμη από τις μεθόδους κωδικοποίησης μετασχηματισμών (Transform Coding Algorithms). Για παράδειγμα η μέθοδος των Chen και Smith στο [6] απαιτεί ένα δυσδιάστατο μετασχηματισμό συνημίτονου (cosine transform) πάνω σε ένα 16x16 block, όπως επίσης και πολλαπλά περάσματα του μετασχηματισμού πάνω στα δεδομένα για να συλλέξει διάφορα στατιστικά σχετικά με τους συντελεστές του μετασχηματισμού. Ο QPC (Quantum Pattern Coding) επίσης απαιτεί πολλές αριθμητικές πράξεις, το λιγότερο τις διπλάσιες από ότι ο βασικός BTC αλγόριθμος, ενώ ο SBC (Subband Coding) παρουσιάζει υπολογιστικές απαιτήσεις συγκρίσιμες με αυτές των κωδικοποιήσεων μετασχηματισμού.

Θα πρέπει να αναφερθεί ότι διάφορες μελέτες έχουν δείξει ότι τα κριτήρια πιστότητας, μέσο τετραγωνικό σφάλμα και μέσο απόλυτο σφάλμα, πολλές φορές δεν έχουν σχέση με τις εκτιμήσεις κάποιων ειδικών σε θέματα εικόνας. Δηλαδή σε μερικές περιπτώσεις, εικόνες με μεγάλο μέσο τετραγωνικό σφάλμα εκτιμούνται καλύτερα από άλλες με μικρότερο. Ωστόσο, αν και το MSE δεν σχετίζεται και τόσο καλά με την ποιότητα των εικόνων που παίρνουμε, θα δώσουμε τις τιμές του, όπως και αυτό του MAE των εικόνων που χρησιμοποιήσαμε στο κεφάλαιο αυτό για λόγους αριθμητικών συγκρίσεων (Πίνακας 3).

	MSE	MAE
Σχήμα 4.2α	69.028	4.686
Σχήμα 4.2β	108.002	7.520
Σχήμα 4.3α	64.019	4.473
Σχήμα 4.3β	20.624	2.089
Σχήμα 4.7α	74.048	4.871
Σχήμα 4.7β	84.147	6.005
Σχήμα 4.8α	79.129	4.961
Σχήμα 4.8β	90.285	6.152

Πίνακας 3 MSE και MAE των αποτελεσμάτων

Βιβλιογραφία

[1] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding", IEEE Trans. Commun., COM-27, pp1335-1342, Sept.1979

[2] D. J. Healy and O. R.Mitchell, "Digital video bandwidth compression using block truncation coding", IEEE Trans. Commun. Vol. COM-29, pp1809-1817, Dec 1981

[3] Y.Wu and D. C. Coll, "Multilevel block truncation coding using a minimax error criterion for high-fidelity compression of digital images", IEEE Trans. Commun., vol. Com-41, pp 1179-1191, Aug. 1993

[4] M. D Lema and O. R. Mitchell, "Absolute moment block truncation coding and application to color image", IEEE Trans. Commun., COM-32, pp1148-1157, Oct. 1984

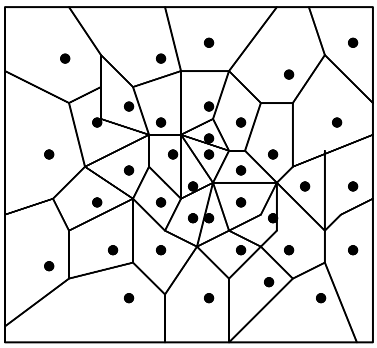
[5] G. R. Arce and N. C. Gallagher Jr., "BTC image using median filter roots", IEEE Trans. Commun. Vol. COM-31, June 1983

[6] W. H. Chen and C.H. smith, "Adaptive Coding of monochrome and color images", IEEE Trans. Commun., vol. COM-25, pp1285-1292, Nov. 1977

5. Vector Quantization

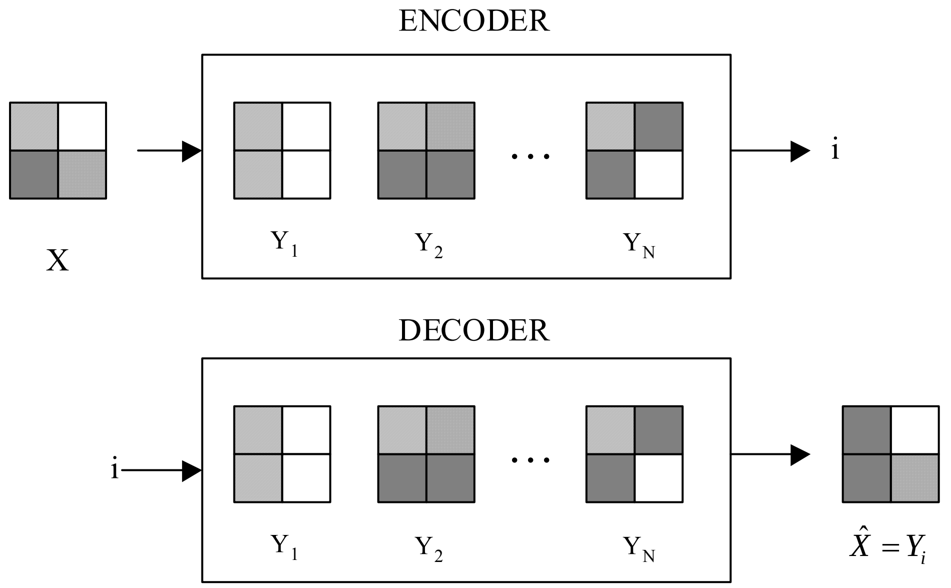
5.1 Εισαγωγή

Ο διανυσματικός κβαντισμός (vector quantization) είναι ένας αλγόριθμος συμπίεσης, ο οποίος εφαρμόζεται σε διανύσματα αντί σε βαθμωτά και μπορεί εύκολα να κατανοηθεί μέσω του βαθμοτού κβαντισμού (scalar quantization). Το scalar quantization απεικονίζει ένα μεγάλο σύνολο από αριθμούς σε ένα μικρότερο και περιέχει λειτουργίες όπως “στρογγυλοποίηση στον κοντινότερο ακέραιο αριθμό” αν και στην γενική περίπτωση τα επίπεδα κβαντισμού δεν πρέπει να είναι ακέραιοι. Το vector quantization στρογγυλοποιεί (ή κβαντίζει) σύνολα από αριθμούς μαζί, αντί για ένα την φορά. Αυτά τα σύνολα των αριθμών ονομάζονται διανύσματα εισόδου(input vectors) και τα επίπεδα κβαντισμού αναπαραγόμενα διανύσματα(reproduction vectors). Για να οριστεί ένας vector quantizer χρειάζεται ένα σύνολο από πιθανά reproduction vectors και ένα κανόνα απεικόνισης των input vectors στα reproduction vectors. Ένα δυσδιάστατο παράδειγμα του VQ φαίνεται στο σχήμα 5.1. Οι τελείες απεικονίζουν τα reproduction vectors και ο κανόνας απεικόνισης ενεικνύεται από τις γραμμές, οι οποίες περιγράφουν τα όρια μεταξύ των περιοχών. Κάθε input vector, το οποίο βρίσκεται σε μια δοσμένη περιοχή, απεικονίζεται με το reproduction vector στην ίδια περιοχή.



5.1 Ένα 2D παράδειγμ του VQ

Ένας άλλος τρόπος για να περιγράψουμε το σύστημα είναι αυτός του σχήματος 5.2, ο οποίος δείχνει ένα VQ ο ποίος λειτουργεί απευθείας σε κομμάτια εικόνας (pixel blocks). Η εικόνα εισόδου αναλύεται σε μια σειρά από σύνολα από pixel, στο παράδειγμά μας 2 x 2. Ο κωδικοποιητής "βλέπει" ένα input vector και εφαρμόζει τον κανόνα απεικόνισης για να διαλέξει ένα από τα N πιθανά reproduction inputs από το codebook. Το reproduction vector που διαλέχτηκε Y_i καλείται και κωδικολέξη (codeword) και είναι (συνήθως) ένα gray scale pixel block με τις διαστάσεις του input block. Ο δείκτης i του Y_i είναι ένα δυαδικό διάνυσμα. Να σημειώσουμε ότι το X και το $\hat{X}$ είναι μερικώς διαφοροποιημένα στο σχήμα 3.2, για να δείξουμε ότι το VQ είναι μια lossy τεχνική συμπίεσης.



Σχήμα 5.2 Vector Quantizer

Εάν ο κώδικας έχει ένα σταθερό ρυθμό από b bits για κάθε διάνυσμα εισόδου, τότε το i έχει μέγεθος b. Με μεταβλητό ρυθμό κώδικα, οι δείκτες i έχουν μεταβλητό μέγεθος και b είναι το μέσο μέγεθος τους. Η συμπιεσμένη εικόνα απεικονίζεται με αυτούς τους δείκτες i και η συμπιεσμένη απεικόνιση απαιτεί λιγότερα bits. Για παράδειγμα για μια εικόνα με 8-bits per pixel (bpp), για το block εισόδου απαιτούνται 4 x 8 = 32 bits. Για σταθερό μήκος κώδικα με 256 codewords στο codebook απαιτείται 8-bit δείκτης. Έτσι ο λόγος συμπίεσης είναι 32:8 ή 4:1. Ο αποκωδικοποιητής έχει επίσης ένα αντίγραφο του codebook, το οποίο λειτουργεί σαν ένα απλό lookup table. Αφού λάβει ένα δείκτη i βγάζει σαν είσοδο το αποθηκευμένο codeword Yi.

Η λειτουργία του αποκωδικοποιητή είναι περιγράφεται επομενώς τελείως από την στιγμή που καθορίσαμε το codebook. Η λειτουργία του κωδικοποιητή απαιτεί μια επιλογή για τον κανόνα απεικόνισης. Το βασικό μοντέλο του Shannon για κωδικοποίηση πηγής παρέχει έναν κωδικοποιητή βέλτιστο για ένα δοσμένο codebook, εάν ο στόχος είναι να ελαχιστοποιήσουμε την μέση παραμόρφωση. Εάν υποθέσουμε το $d(X,\hat{X}) \geq 0$ δείχνει την παραμόρφωση ή το κόστος της αναπαραγωγής του διανύσματος X σε $\hat{X}$ και εάν επιπλέον υποθέσουμε ότι η συνολική παραμόρφωση του συστήματος μετριέται με μια μέση παραμόρφωση, ο βέλτιστος κωδικοποιηής διαλέγει το διάνυσμα Y_i όπου:

$$d(X,Y_i) \leq d(X,Y_j), \quad \text{for all j} \quad (5.1)$$

Με άλλα λόγια, ο αποκωδικοποιητής λειτουργεί σε ένα κοντινό γείτονα ή σε μικρότερο μέτρο παραμόρφωσης.

Η επιλογή του τρόπου μέτρησης της παραμόρφωσης μας επιτρέπει να ποσοτικοποιήσουμε την απόδοση ενός VQ σε μία έκφραση η οποία μπορεί να και να χρησιμοποιηθεί στην ανάλυση και στην βελτιστοποίηση σχεδιασμού. Ο συνηθέστερος τρόπος μέτρησης παραμόρφωσης για συμπίεση εικόνας είναι το μέσο τετραγωνικό σφάλμα. Αν και υπάρχούν πολλές πολλές προσεγγίσεις για τον σχεδιασμό του κώδικα, οι πιο συνηθισμένοι είναι αλγόριθμοι του Lloyd (Forgey, Isodata, k-mean) και ο LBG (Linde, Buzo, Gray). Εμείς στην εργασία θα δούμε μια άλλη προσέγγιση για τον σχεδιασμό του codebook με χρήση νευρωνικόύ δικτύου όπως την έχει μορφοποιήσει ο Kohonen και συγκεκριμένα τον SOF (Self–organizing maps) αλγόριθμο.

5.2 Self-organizing map (SOM)

Ο αλγόριθμος self-organizing map του Kohonen έχει δώσει μορφή σε μια βάση στην έρευνα για την εφαρμογή δικτυακών μοντέλων στο πρόβλημα του σχεδιασμού του codebook για το vector quantization. Ο Kohonen έχει εισαγάγει την έννοια των κλάσεων (class), οι οποίες είναι διατεταγμένες σε έναν “τοπολογικό χάρτη”. Σε πολλούς clustering αλγορίθμους, όπως για παράδειγμα αυτός των k-means, κάθε διάνυσμα εισόδου x κατηγοριοποιείται και μόνο η “winning” κλάση τροποποιείται σε κάθε επανάληψη. Στον SOM αλγόριθμο, το διάνυσμα x χρησιμοποιείται για την ενημέρωση όχι μόνο της «winning» κλάσεως, αλλά και των γειτονικών κλάσεων σύμφωνα με τον ακόλουθο κανόνα:

Για κάθε διάνυσμα x από το training set:

1) Ταξινόμησε το x σύμφωνα με

$$x \in C_i if \|x - w_i\| = \min_j \|x - w_j\| \quad (5.2)$$

2) Ενημέρωσε τα w_j σύμφωνα με

$$w_j(t+1) = \begin{cases} w_j + a(t)[x - w_j(t)], & C_j \in N(C_i, t) \\ w_j(t), & C_j \notin N(C_i, t) \end{cases} \quad (5.3)$$

όπου w είναι το μελλοντικό διάνυσμα, a μια παράμετρος μάθησης με τιμές από 0<a<1 και $N(C_i, t)$ είναι το σύνολο των κλάσεων που είναι στην γειτονιά της "winning" κλάσεως C_i την χρονική στιγμή t. Η κλάση w_i συγκλίνει στον μέσο όρο της κλάσης. Η γειτονιά μια κλάσης ορίζεται σε σχέση με μερικές μετρήσεις απόστασης σε μια τοπολογική διάταξη των κλάσεων. Για παράδειγμα, εάν οι κλάσεις έχουν διαταχθεί σε ένα δυσδιάστατο τετραγωνικό πλέγμα, η γειτονιά της κλάσης μπορεί να οριστεί σαν ένα σύνολο από κλάσεις των οποίων οι Ευκλείδειες αποστάσεις από την κλάση αυτή είναι μικρότερη από ένα ορισμένο όριο. Αρχικά η γειτονιά μπορεί να είναι αρκετά μεγάλη κατά την διάρκεια του training, για παράδειγμα ο μισός αριθμός των κλάσεων. Καθώς το training προχωρεί το μέγεθος της γειτονιάς μειώνεται, ως ότου τελικά να περιέχει μόνο την κλάση αυτή. Κατά την διάρκεια επίσης του training η παράμετρος εκμάθησης a μειώνεται σε μια μικρή τιμή (π.χ 0.001) για σωστότερη σύγκλιση του αλγόριθμου. Στην επόμενη παράγραφο θα δούμε αναλυτικότερα τον αλγόριθμο και την αρχιτεκτονική του νευρωνικού δικτύου, το οποίο δημιουργεί το codebook τουVQ.

5.3 Αλγόριθμος και αρχιτεκτονική του δικτύου

Το κίνητρο του αλγορίθμου για το νευρωνικό δίκτυο αυτό είναι να βρει την μονάδα εξόδου, η οποία είναι η κοντινότερη στο διάνυσμα εισόδου. Για τον σκοπό αυτό, εάν το x και το wανήκουν στην ίδια κλάση τότε μετακινούμε τα βάρη τα βάρη προς τις τιμές του διανύσματος εισόδου. Εάν x και w ανήκουν σε διαφορετικές κλάσεις τότε μετακινούμε τα βάρη μακριά από το διάνυσμα εισόδου.

Η ονοματολογία των παραμέτρων που χρησιμοποιούμε στον αλγόριθμο είναι:

X training vector $(x_1,\ldots,x_i,\ldots,x_n)$

T σωστή κλάση ή κλάση για το training vector

w_j weight vector($w_{nj},\ldots,w_{ij},\ldots,w_{nj}$)

C_j η κλάση που αντιπροσωπεύει η jth μονάδα εξόδου

$\|x - w_j\|$ η Ευκλείδεια απόσταση ανάμεσα στο διάνυσμα εισόδου και το weight vector για την jth μονάδα εισόδου.

Ο αλγόριθμος μπορεί να περιγραφεί ως εξής:

Βήμα 0. Αρχικοποίηση των weight vectors (μερικές στρατηγικές αρχικοποίησης θα αναφερθούν παρακάτω), αρχικοποίηση της παραμέτρου εκμάθησης, a(0).

Βήμα 1. Εφόσον οι συνθήκες τερματισμού δεν ισχύουν, κάνε τα βήματα 2-6.

Βήμα 2. Για κάθε training vector x, κάνε τα βήματα 3-4.

Βήμα 3. Βρες το J, έτσι ώστε το $\|x - w_j\|$ να είναι ελάχιστο.

Βήμα 4. Ενημέρωσε το w_j με τον εξής τρόπο:

Εάν T = C_j, τότε

$$w_j(new) = w_j(old) - a\lfloor x - w_j(old) \rfloor$$

Εάν $T \neq C_j$, τότε

$$w_j(new) = w_j(old) - a\lfloor x - w_j(old) \rfloor$$

Βήμα 5. Μείωσε την παράμετρο εκμάθησης

Βήμα 6. Έλεγξε τις συνθήκες τερματισμού:

Οι συνθήκες μπορούν να ορίζουν ένα συγκεκριμένο αριθμό επαναλήψεων ή η παράμετρος εκμάθησης φτάσει μια ικανοποιητικά μικρή τιμή.

Η απλούστερη μέθοδος για την αρχικοποίηση των weight vectors είναι να πάρουμε ως αρχικά διανύσματα τα πρώτα m training vector για weight vectors. Τα υπόλοιπα διανύσματα θα χρησιμοποιηθούν για το training του νευρωνικού. Μια άλλη απλή μέθοδος είναι να πάρουμε m τυχαία training vectors για weight vectors. Στην συνέχεια θα δώσουμε ένα απλό παράδειγμα για καλύτερη κατανόηση του αλγορίθμου.

Έστω ότι έχουμε να κατηγοριοποιήσουμε 5 διανύσματα:

Vector	**Class**
(1, 1, 0, 0)	1
(0, 0, 0, 1)	2
(0, 0, 1, 1)	2

(1, 0, 0, 0)   1

(0, 1, 1, 0)   2

Τα πρώτα δύο διανύσματα θα χρησιμοποιηθούν για την αρχικοποίηση των δύο weight vectors. Έτσι η πρώτη μονάδα εξόδου θα απεικονίζει την κλάση 1και το δεύτερο την κλάση 2 (συμβολικά C_1=1 και C_2=1). Έτσι μας μένουν τρία training vectors.

Βήμα 0. Αρχικοποίηση βαρών:

$w_1 = (1, 1, 0, 0)$

$w_2 = (0, 0, 0, 1)$.

Βήμα 1. Έναρξη υπολογισμών.

Βήμα 2. Για το input vector x = (0, 0, 1,1) με T = 2 κάνε τα βήματα 3-4

Βήμα 3. J = 2, εφόσον το x είναι πιο κοντά στο w_2 από ότι στο w_1.

Βήμα 4 Εφόσον T = 2 και C_2 = 2, ενημέρωσε το w_2 ως εξής:

$w_2 = (0,0,0,1) + 0.1[(0,0,1,1) – (0,0,0,1)]$

$= (0,0,0.1,0.1)$

Βήμα 2. Για το input vector x = (1, 0, 0, 0) με T = 1 κάνε τα βήματα 3-4

Βήμα 3. J = 1, εφόσον το x είναι πιο κοντά στο w_1 από ότι στο w_2.

Βήμα 4 Εφόσον T = 1 και C_1 = 1, ενημέρωσε το w_1 ως εξής:

$w_1 = (1,1,0,0) + 0.1[(1,0,0,0) – (1,1,0,0)]$

$= (1,0.9,0,0)$.

Βήμα 2. Για το input vector x = (0, 1, 1, 0) με T = 2 κάνε τα βήματα 3-4

Βήμα 3. J = 1, εφόσον το x είναι πιο κοντά στο w_1 από ότι στο w_2.

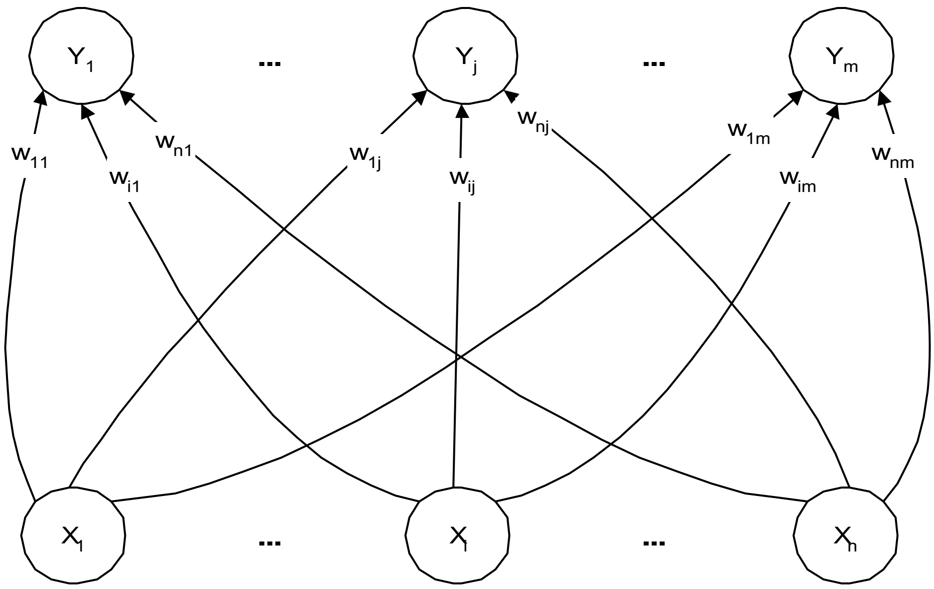
Βήμα 4 Εφόσον T = 2 και C_1 = 1, ενημέρωσε το w_1 ως εξής:

w_1 = (1,0.9,0,0) - 0.1[(0,1,1,0) – (1,0,9,0,0)] = (1.1,0.89,-0.1,0).

Βήμα 5. Εδώ ολοκληρώθηκε μια εποχή του training. Μείωσε την παράμετρο εκμάθησης.

Βήμα 6. Έλεγξε τις συνθήκες τερματισμού.

Στο σχήμα 5.2 φαίνεται η αρχιτεκτονική του νευρωνικού δικτύου.



Σχήμα 5.3 Το νευρωνικό Vector quantization

5.4 Ιδιότητες του SOM και σύγκρισή του με τον LBG αλγόριθμο

Ο αλγόριθμος του SOM έχει μερικές σημαντικές ιδιότητες που τον κάνουν ελκυστικό για την χρήση στην δημιουργία του codebook για το vector quantization.

1) Το σύνολο των μελλοντικών διανυσμάτων είναι μια πολύ καλή προσέγγιση του αρχικού διαστήματος εισόδου.
2) Τα μελλοντικά διανύσματα είναι τοπολογικά διατεταγμένα στον χάρτη (map), έτσι ώστε η συσχέτιση μεταξύ των μελλοντικών διανυσμάτων να αυξάνει καθώς η απόσταση τους μειώνεται.

3) Η ένταση του μελλοντικού χάρτη αντιστοιχεί στην ένταση της κατανομής της εισόδου, έτσι ώστε πεδία με μεγαλύτερη κατανομή πιθανότητας να έχουν καλύτερη ανάλυση (resolution) από άλλες με μικρότερη ένταση.

Ο LBG αλγόριθμος για τον σχεδιασμό codebook και ο SOM αλγόριθμος είναι στενά συσχετισμένοι. Ο LBG αλγόριθμος είναι ισοδύναμος του SOM για μέγεθος γειτονιάς ίσο με ένα, $N(C_i) = \{C_i\}$. Ο LBG αλγόριθμος ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα (MSE) μέσα σε μια κλάση

$$D_{MSE}(x, w_i) = E\left[\|x - w_i\|^2\right], x \in C_i \quad (5.4)$$

θέτοντας το διάνυσμα w_i στην μέση τιμή της κλάσης. Οι μέσες τιμές υπολογίζονται για παράδειγμα μετά από n δείγματα. Για την ενημέρωση του διανύσματος w_i μετά από την επεξεργασία κάθε training vector, μπορεί να χρησιμοποιηθεί μια gradient descent προσέγγιση βασισμένη στην ελαχιστοποίηση. Τότε ο κανόνας ενημέρωσης γίνεται

$$\begin{aligned} &w_i(t+1) = w_i(t) - \frac{1}{2} a \nabla D_{MSE}(x, w_i(t)) = \\ &w_i(t) + a(t)[x - w_i(t)] \quad (5.5) \end{aligned}$$

η οποία είναι ισοδύναμη με την σχέση (5.3) για $N(C_i) = \{C_i\}$. Άρα και οι δύο περιπτώσεις υπολογίζουν τον μέσο όρο της κλάσης, έχοντας σαν αποτέλεσμα ένα codebook με ένα ελάχιστο MSE.

Ένας αριθμός ερευνών έχουν χρησιμοποιήσει με επιτυχία τον SOM αλγόριθμο για την δημιουργία codebooks για vector quantization και έχουν αποδείξει αρκετά πλεονεκτήματα της χρήσης του SOM αλγορίθμου σε σχέση με τον κλασικό LBG αλγόριθμο. Μερικά από αυτά είναι: μικρότερη ευαισθησία στην αρχικοποίηση του codebook, καλύτερος βαθμός στην απόδοση παραμόρφωσης και γρηγορότερη σύγκλιση του αλγορίθμου. Έχει επίσης δειχθεί ότι τα codewords, ή τα βάρη, συγκλίνουν στην μέση τιμή της κλάσης. Τα παραγόμενα codewords είναι βέλτιστα από την άποψη ότι η μέση παραμόρφωση ελαχιστοποιείται.

Βιβλιογραφία

- Pamela C. Cosman, Karen L. Oehler, Eve A. Riskin and Robert M. Gray, *Using Vector Quantization for Image Processing,* Proceedings of the IEEE, Vol. 81, No. 9, September 1993
- Robert D. Dony and Simon Haykin, *Neural Network Approaches to Image Compression,* Proceedings of the IEEE, Vol. 83, No. 2, February 1995

6. Ο BTC-VQ αλγόριθμος

6.1 Εισαγωγή

Στα προηγούμενα δύο κεφάλαια αναλύσαμε σε θεωρητικό και πρακτικό επίπεδο δύο βασικούς lossy αλγορίθμους συμπίεσης εικόνας: τον Block Truncation Coding και τον Vector Quantization. Στο κεφάλαιο αυτό θα δούμε έναν κωδικοποιητή, ο οποίος είναι ένας συνδυασμός των δύο παραπάνω αλγορίθμων. Σκοπός της εργασίας αυτής είναι να δούμε κατά πόσο ο κωδικοποιητής αυτός θα είχε την δυνατότητα να συνδυάσει τα πλεονεκτήματα των αλγορίθμων αυτών, έτσι ώστε να γίνει ελκυστικός για μεταδόσεις σε πραγματικό χρόνο. Δηλαδή να έχει την δυνατότητα υψηλών λόγων συμπίεσης, χωρίς επιπλέον να αυξάνει τον χρόνο κωδικοποίησης. Στις επόμενες παραγράφους θα δούμε αναλυτικά το προτεινόμενο σχήμα κωδικοποιητή όπως φυσικά και τα αποτελέσματα που πήραμε, τόσο από οπτικής πλευράς όπως επίσης εξετάζοντας και τα αποτελέσματα από ποσοτική πλευρά.

6.2 Ο Vector Quantizer (VQ)

Όπως είπαμε και στο προηγούμενο κεφάλαιο το vector quantization είναι μια ικανοποιητική τεχνική συμπίεσης δεδομένων, πετυχαίνοντας σχετικά υψηλούς λόγους συμπίεσης με αποδεκτή πιστότητα. Είναι μια διαδικασία όπου τα δεδομένα εισόδου "σπάζονται" σε block, ή διανύσματα, τα οποία αντιπροσωπεύουν την πληροφορία προς κωδικοποίηση. Ένα σύνολο από διανύσματα (codewords), τα οποία αντιπροσωπεύουν "καλά" την είσοδο, ομαδοποιούνται σε ένα codebook. Κατά την διάρκεια της διαδικασίας κωδικοποίησης κάθε διάνυσμα εισόδου αντικαθίσταται με το "κοντινότερο" codeword που περιέχεται στο codebook.

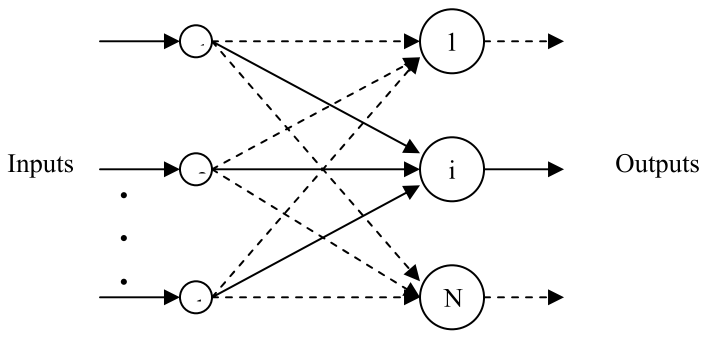
Ο κλασσικός αλγόριθμος που χρησιμοποιείται για το σχεδιασμό του VQ είναι ο LBG [1]. Ο σχεδιασμός του codebook με τον αλγόριθμο αυτό ολοκληρώνεται όταν φτάσει ένα αποδεκτό όριο παραμόρφωσης μέσα στο training set, μετρημένο σε μέσο τετραγωνικό σφάλμα (mean square error, MSE). Ο LBG αλγόριθμος συγκλίνει σε ένα τοπικό ελάχιστο του MSE και είναι πολύ ευαίσθητος από την αρχικοποίηση του codebook. Αυτό τον κάνει ακατάλληλο για περιπτώσεις όπου χρειαζόμαστε έναν γρήγορο VQ που να προσαρμόζεται στα στατιστικά δεδομένα της πηγής.

Αντί της χρήσης του LBG αλγορίθμου, ένας αρκετά μεγάλος αριθμός από νευρωνικές προσεγγίσεις έχουν επιτυχημένα εφαρμοστεί για τον σχεδιασμό του VQ codebook [2, 3]. Η κατηγορία των unsupervised competitive learning (CL) νευρωνικών δικτύων είναι πολύ για τον σχεδιασμό του codebook λόγω της απλότητας υλοποίησης του και τις ιδιότητες γρήγορης σύγκλισης. Η εμπειρία από τις εφαρμογές των νευρωνικών δικτύων που ανήκουν στην οικογένεια των CL σε δύσκολες πολυδιάστατες διαδικασίες clustering αποδεικνύει τα προαναφερθεί πλεονεκτήματα, όπως επίσης και τις πολύ καλές δυνατότητες γενίκευσης που έχουν [4, 5].Ο αλγόριθμος competitive learning με αρχικοποίηση από το training set έχει επιλεχθεί για τον του codebook των προτεινόμενων από εμάς BTC-VQ σχημάτων των παρακάτω παραγράφων λόγω των μικρών χρόνων εκπαίδευσης του νευρωνικού όπως επίσης και για την ικανοποιητική χρησιμοποίηση των μονάδων του δικτύου που πετυχαίνει.

Στους CL αλγορίθμους ένα διάνυσμα w_i συνδέεται με κάθε κόμβο (node) του δικτύου. Οι διαστάσεις του διανύσματος είναι ίσες με αυτές του διανύσματος εισόδου και οι συνιστώσες του καλούνται βάρη του κόμβου (node weights). Στο τέλος της διαδικασίας παραγωγής του codebook, το w_i αναπαριστά το $i-th$ codeword (ο πίνακας βαρών του δικτύου $W = [w_1,...,w_N]$ αναπαριστά το codebook μεγέθους N). Στην αρχή οι συνιστώσες των w_i αρχικοποιούνται με μικρές τυχαίες τιμές (τιμές που κυμαίνονται από 0 ως 1). Μετά ο αλγόριθμος επαναλαμβάνει συνεχώς με διανύσματα εισόδου από το training data έναν ορισμένο αριθμό με σκοπό να πετύχει μια ικανοποιητική αναπαράσταση για τα codewords. Το CL σχήμα μπορεί να περιγραφεί ως εξής: για κάθε training vector x και για όλα τα νευρώνια εξόδου υπολογίζεται η απόσταση $D = d(x, w_i)$ και επιλέγεται η μονάδα εξόδου w_i^* με την μικρότερη απόσταση. Τότε το weight vector της ενημερώνονται σύμφωνα με τον κανόνα:

$$\mathrm{w}_i^*(t+1) = \mathrm{w}_i^*(t) + e(t, N)[x(t) - \mathrm{w}_i^*(t)] \quad (6.1)$$

όπου t είναι ο χρόνος του training και e(t, N) μια παράμετρος εκμάθησης, η οποία μειώνεται κατά την διάρκεια της διαδικασίας εκπαίδευσης (0<e(t, N)>1). Όπως επίσης ενημερώνονται και όλες μονάδες οι οποίες βρίσκονται στην γειτονιά του w_i^*, ή αλλιώς των οποίων η απόσταση N από το w_i^* είναι μικρότερη από N_c. Περιοδικά το μήκος της γειτονίας μειώνεται με το να μειώνουμε το N_c ως το 0. Η μορφή ενός CL νευρωνικού δικτύου φαίνεται στο σχήμα 6.1.



Σχήμα 6.1 Η δομή ενός CL νευρωνικού δικτύου

Ένα βασικό πρόβλημα με τους CL αλγορίθμους είναι ένα σημαντικό ποσοστό των μονάδων τους (πάνω από 30%) δεν ενημερώνεται στην διαδικασία εκπαίδευσης. Σαν συνέπεια έχουμε το ότι αυτό το ποσοστό των μονάδων δεν χρησιμοποιούνται ούτε και στην διαδικασία απεικόνισης. Μια πιθανή λύση του προβλήματος θα μπορούσε να είναι μια διαφορετική αρχικοποίηση των βαρών του δικτύου στην φάση της εκπαίδευσης. Αντί της χρήσης μικρών τυχαίων τιμών για αρχικά βάρη δικτύου, μπορούμε να αρχικοποιήσουμε τα βάρη από το training set. Έτσι όλες οι μονάδες έχουν την ίδια πιθανότητα να νικήσουν, όπως επίσης αποφεύγουμε και την στατιστική απόκλιση που υπάρχει δίνοντας τυχαίες τιμές.

6.3 Μια πρώτη προσέγγιση

Όπως έχουμε δει ο BTC αλγόριθμος έχει την ιδιότητα να διατηρεί σημαντικά χαρακτηριστικά block εικόνων, όπως την μέση τιμή των εντάσεων $\bar{x}$ και την τυπική απόκλιση σ των pixels που βρίσκονται μέσα στο block αυτό. Επιπλέον συμπίεση του συνολικού bit stream του BTC πετυχαίνεται κβαντίζοντας επίσης και τα δύο παραπάνω χαρακτηριστικά. Σ' αυτή την περίπτωση εμπειρικές μετρήσεις δείχνουν ότι σ' έναν 8-bit κβαντιστή δεν χρησιμοποιούνται τιμές μεγαλύτερες από το 63 για την τυπική απόκλιση σ. Έτσι μπορούμε να προτείνουμε έναν δυσδιάστατο κβαντιστή για την ταυτόχρονη κωδικοποίηση της μέσης τιμής και την τυπικής απόκλισης, χρησιμοποιώντας 10 bits κάθε pixel block. Τότε το κριτήριο που χρησιμοποιείται για τον σχεδιασμό του δυσδιάστατο κβαντιστή είναι ότι το σφάλμα κβάντισης είναι περισσότερο αισθητό σε περιοχές όπου η απόκλιση είναι μικρή , όπου οι $\bar{x}$-σ συνδυασμοί διαλέγονται έτσι ώστε το παραγόμενο σφάλμα κβάντισης να μειώνεται με το σ. Ένας τέτοιος κβαντιστής παρουσιάζεται στον πίνακα 6.1, όπου δείχνονται οι κβαντισμένες τιμές εξόδου του σ και οι αντίστοιχοι αριθμοί επιπέδων που επιτρέπονται για το $\bar{x}$,ενώ για κάθε δοσμένο σ τα επίπεδα του $\bar{x}$ είναι ομοιόμορφα κατανεμημένα.

σ	Αριθμός των ομοιόμορφων κατανεμημένων επιπέδων για $\bar{x}$
0	256
1, 2, 4	128
6, 8, 13, 19, 28	64
42, 56	32

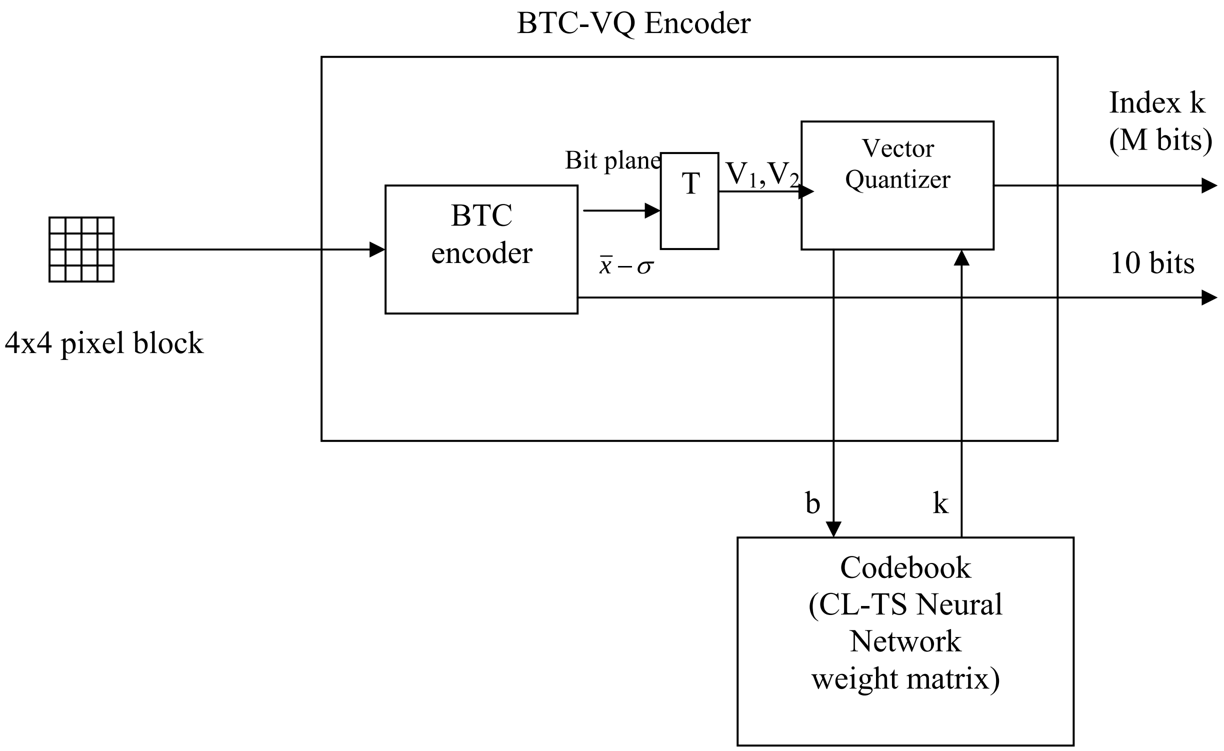
Πίνακας 6.1 Ταυτόχρονη κβάντιση $\bar{x}$ - σ χρησιμοποιώντας 10 bits

Έχοντας αυτό το δεδομένο σκεφτήκαμε ότι θα μπορούσαμε να πετύχουμε συμπιέσεις της τάξης 1,1875 - 1.0625 bpp με το να συνδυάσουμε τον BTC αλγόριθμο με ταυτόχρονο κβαντισμό των $\bar{x}$ - σ και έναν vector quantizer, ο οποίος θα κωδικοποιεί το bit-plane.

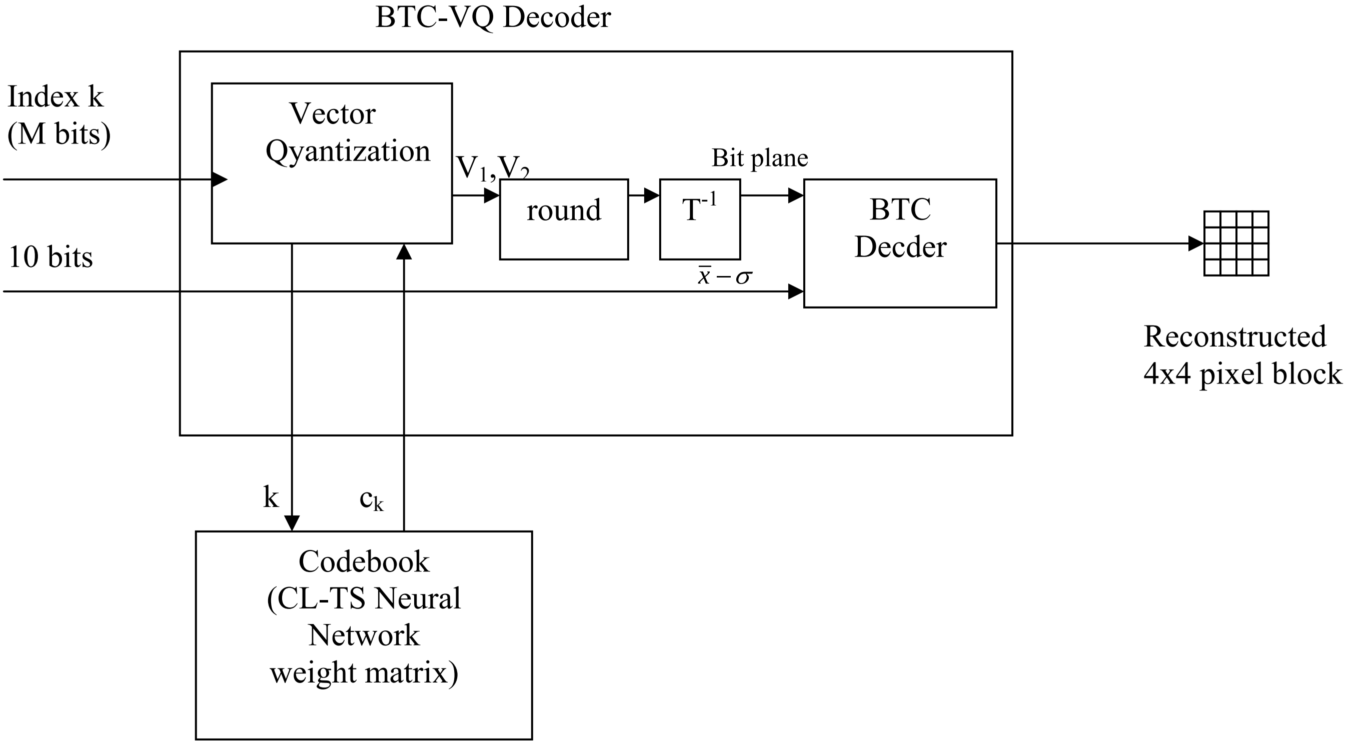
Με σκοπό να αξιοποιήσουμε τις δυνατότητες συμπίεσης ενός VQ, μετασχηματίζουμε το bit plane που παράγεται από τον BTC αλγόριθμο σε ένα διάνυσμα b = $[v_1, v_2]$, όπου τα v_1, v_2 είναι ακέραιες τιμές που παράγονται με το σπάσιμο και μετασχηματισμό του bit plane σε δύο 8-bit σειρές. Για παράδειγμα ας πάρουμε μια περίπτωση, όπου το παραγόμενο από τον BTC bit plane είναι το

$$\begin{matrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Αυτό το bit plane αναπαριστάται με τις δύο 8-bit σειρές v = [1 0 1 1 0 1 1 1 , 1 0 0 0 0 0 0 1] και στην συνέχεια μετασχηματίζονται σε v_1 = 183 και v_2 = 129. Με αυτές τις υποθέσεις, το vector quantization μπορεί να εφαρμοστεί στα δυσδιάστατα διανύσματα b, το οποίο καθένα απεικονίζει το bit plane του BTC για ένα 4x4 pixel block. Στην περίπτωση ενός VQ με μέγεθος codebook $N = 2^M$, το διάνυσμα b αναπαρίσταται από το k-th codeword c_k, k = 1, …, N. Στον αποκωδικοποιητή μεταδίδεται ο δείκτης k σαν μια M-bit σειρά και φυσικά ο συνδυασμός $\bar{x}$ - σ, πετυχαίνοντας έτσι ρυθμό μετάδοσης $\frac{M+10}{16}$ bpp. Εκεί η έξοδος του vector dequantizer είναι το codeword $c_k = [v_{1k}, v_{2k}]$ που στην γενική περίπτωση είναι δεκαδικά στοιχεία. Εφόσον όμως τα v_{1k}, v_{2k} αναπαριστούν το bit plane πρέπει να πάρουν ακέραιες τιμές. Έτσι στρογγυλεύονται στον κοντινότερο ακέραιο πριν πάρουν την δυαδική τους μορφή. Στην συνέχεια το παραγόμενο bit plane μαζί με τις τιμές των $\bar{x}$ - σ χρησιμοποιούνται από τον BTC αποκωδικοποιητή για την ανακατασκευή του μεταδιδομένου block της εικόνας. Τα σχήματα του κωδικοποιητή και αποκωδικοποιητή που περιγράφηκαν φαίνονται στα σχήματα 6.2 και 6.3 αντίστοιχα.



Σχήμα 6.2 Ο VQ-BTC κωδικοποιήτής



Σχήμα 6.3 Ο VQ-BTC αποκωδικοποιητής

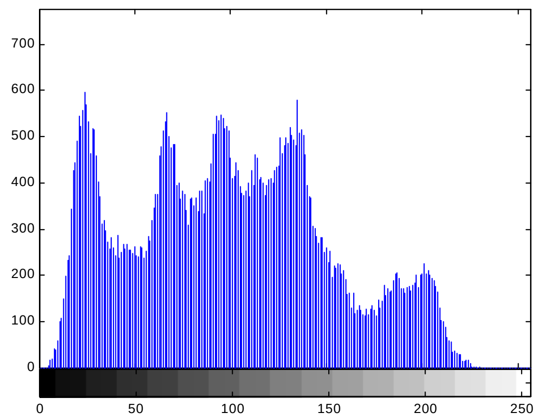
Όπως ειπώθηκε προηγουμένως, το σύνολο των δυσδιάστατων διανυσμάτων b παριστάνουν το παραγόμενο από τον BTC bit plane για 4x4 pixel blocks και απαρτίζουν το input space του VQ. Τα κωδικοποιημένα κατά BTC block προέρχονται από αρκετές εικόνες με διαφορετικά χαρακτηριστικά για να αποτελέσουν την βάση της φάσης εκπαίδευσης του CL-TS αλγορίθμου για έναν πετυχημένο σχεδιασμό του codebook. Χρησιμοποιήσαμε codebooks με μέγεθος 128, 256 και 512, δηλαδή είχαμε ρυθμούς μετάδοσης από 1,1875 - 1.0625 bpp. Ο αριθμός των διανυσμάτων που χρησιμοποιήθηκαν στην φάση εκπαίδευσης ξεπέρασε το 1,000,000, δηλαδή κάτι παραπάνω από 250 εικόνες με διαστάσεις 256x256. Η παράμετρος εκμάθησης ξεκινούσε με την τιμή 0.6 και σαν κριτήριο απόστασης είχαμε την Ευκλείδεια απόσταση. Η παράμετρος N_c πειραματικά τέθηκε στην τιμή 200, έτσι ώστε να έχουμε στην αρχή της φάσης εκπαίδευσης γειτονιές των 6 codewords για codebook μεγέθους 128 και περίπου στο 70% τις διαδικασίας έπαιρνε την τιμή 0, κάτι που σημαίνει ότι κάθε γειτονιά αποτελείται από ένα codeword.

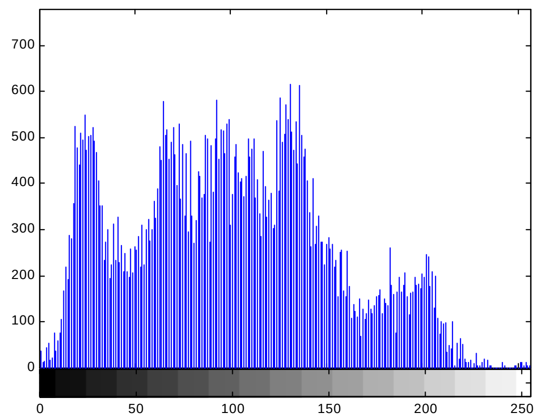
Σχήμα 6.4 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook μεγέθους 128 (1.0625 bpp)

Σχήμα 6.5 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook μεγέθους 256 (1.125 bpp)

Σχήμα 6.6 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook μεγέθους 512 (1.1875 bpp)

Στο σχήμα 6.7 μπορούμε να δούμε τα ιστογράμματα της αρχικής εικόνας LENA όπως επίσης και της ανακατασκευασμένης εικόνας. Βλέποντάς τα παρατηρούμε την μεγάλη ομοιότητα τους. Με αυτό το δεδομένο όπως επίσης και τις μετρήσεις του μέσου απόλυτου σφάλματος που φαίνονται στον πίνακα 6.2 συμπεραίνουμε ότι ο αλγόριθμος αυτός παράγει εικόνες οι οποίες είναι κοντά στις αρχικές εικόνες. Όμως ο βαθμός συμπίεσης που πετυχαίνουμε με την τεχνική αυτή κυμαίνεται σε επίπεδα πάνω του 1 bpp. Στην επόμενη παράγραφο θα δούμε έναν αλγόριθμο ο οποίος θα πετυχαίνει ρυθμό δεδομένων γύρω στο 0.5 bpp. Αυτό το πετυχαίνουμε με την ταυτόχρονη εφαρμογή του vector quantization σε bit plane και των δύο άλλων χαρακτηριστικών που παράγει ο BTC.





Σχήμα 6.7 Τα ιστογράμματα της αρχικής και ανακατασκευασμένης LENA με codebook μεγέθους 256.

	Lena	Bird
128	10.09	4.84
256	10.02	4.78
512	9.68	4.65

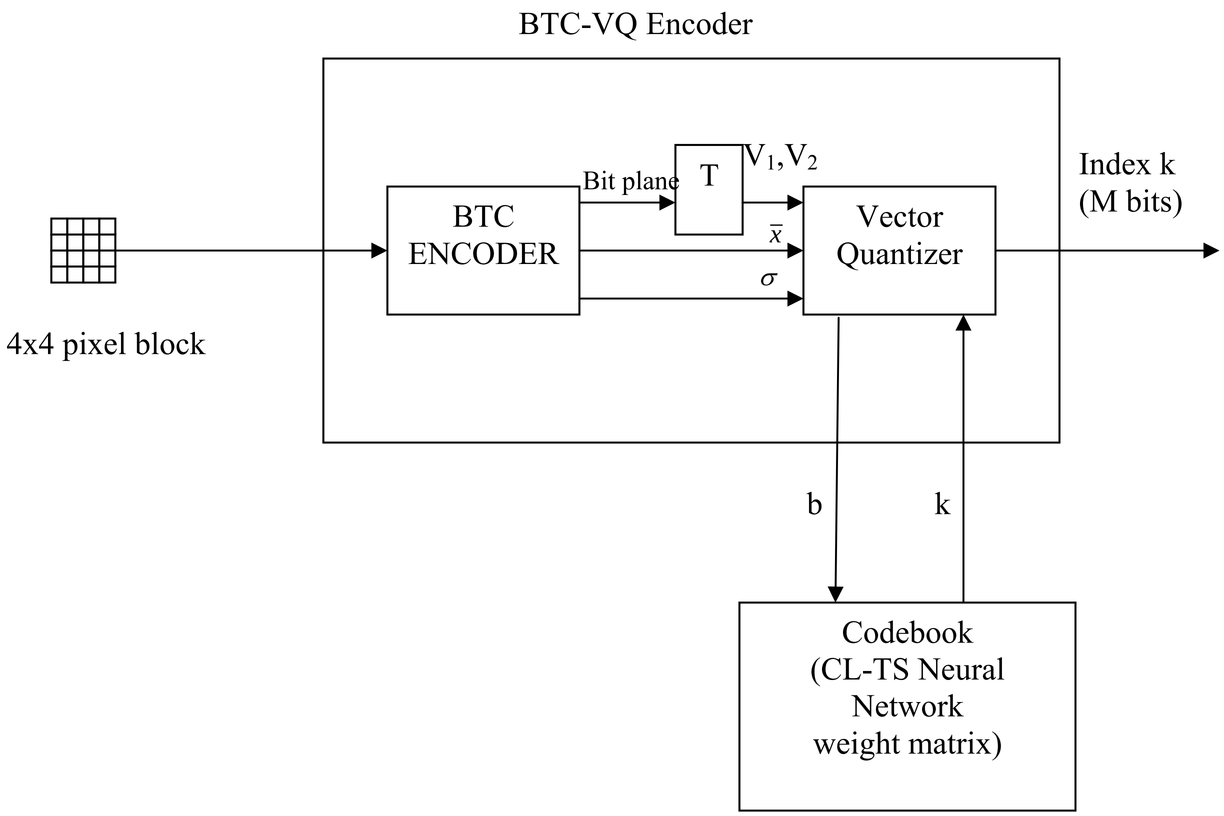
Πίνακας 6.2 Το MAE των εικόνων των δοκιμών

6.3 Ο βασικός VQ-BTC κωδικοποιητής

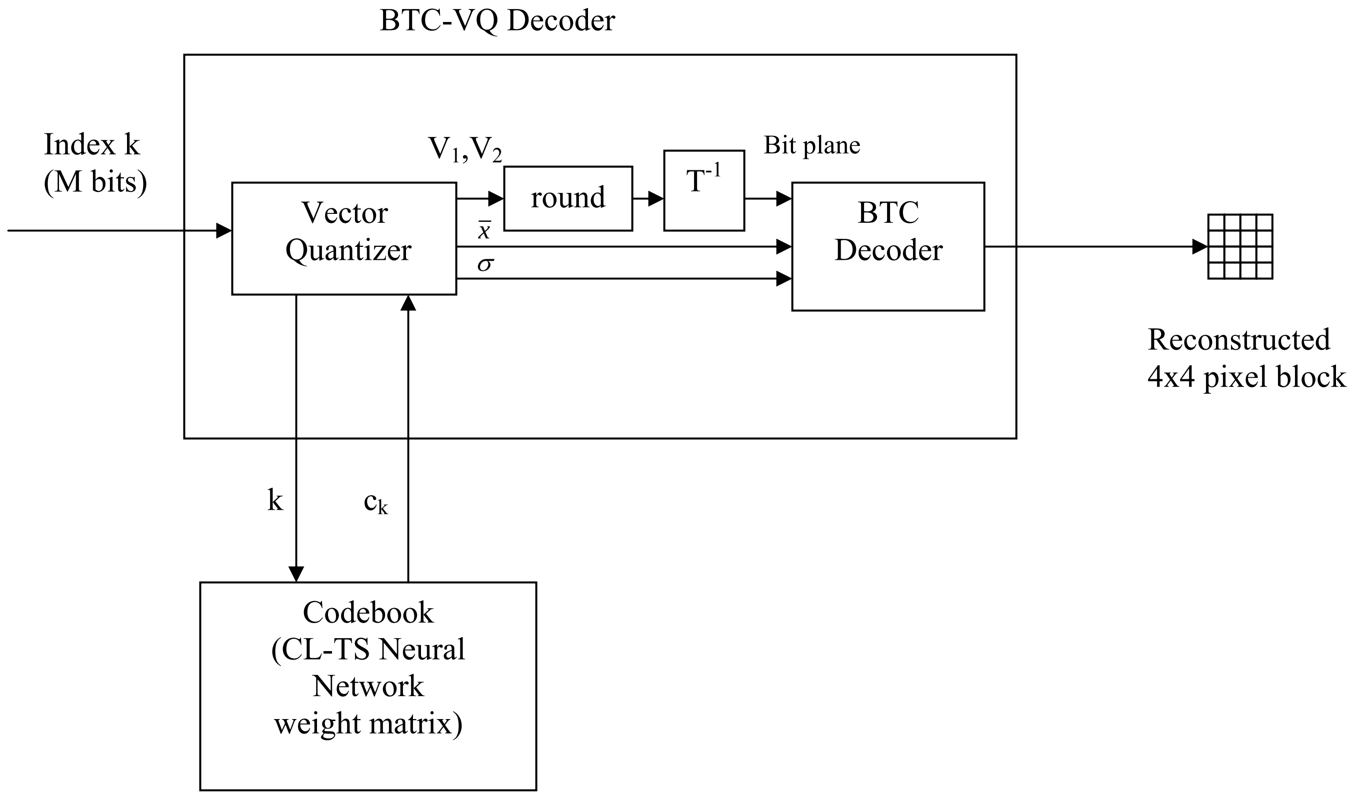
Στην προηγούμενη παράγραφο είδαμε μια πρώτη προσέγγιση του VQ-BTC κωδικοποιητή. Οι συμπιέσεις που πετύχαμε ήταν αρκετά ικανοποιητικές (1,1875 - 1.0625 bpp) και τα αποτελέσματα από ικανοποιητικά ως πολύ καλά. Σκοπός μας όμως ήταν να προσπαθήσουμε να πετύχουμε συμπιέσεις που να είναι μικρότερες του 1 bpp. Έτσι σκεφτήκαμε ότι θα μπορούσαμε να εκμεταλλευτούμε τις μεγάλες δυνατότητες του vector quantization με το να του δίνουμε είσοδο διανύσματα, τα οποία να περιέχουν όλα τα χαρακτηριστικά που μας δίνει ο BTC. Με αυτόν τον τρόπο εξασφαλίζουμε συμπιέσεις 0.4375, 0.5 και 0.5625 για codebook μεγέθους 128, 256 και 512 αντίστοιχα.

Σ' αυτήν την περίπτωση υποθέτουμε ως διάνυσμα εισόδου του κβαντιστή το $b = [v_1, v_2, \bar{x}, \sigma]$, όπου v_1, v_2 είναι δύο ακέραιοι που απεικονίζουν το bit plane του block, $\bar{x}$ η μέση τιμή των εντάσεων των pixel του block και σ η τυπική τους απόκλιση. Στα σχήματα 6.7 και 6.8 φαίνονται ο κωδικοποιητής και ο αποκωδικοποιητής που περιγράφονται στην παράγραφο αυτή. Με αυτές τις υποθέσεις το vector quantization εφαρμόζεται στα 4-διάστατα διανύσματα b, τα οποία στην περίπτωση αυτή αναπαριστούν ολοκληρωμένα την έξοδο του BTC και όχι μόνο το bit plane όπως στην προηγούμενη περίπτωση. Στην περίπτωση ενός VQ με μέγεθος codebook N = 2^M, το διάνυσμα b αναπαρίσταται από το k-th codeword c_k, k = 1, …, N. Στον αποκωδικοποιητή μεταδίδεται ο δείκτης k σαν μια M-bit σειρά πετυχαίνοντας έτσι ρυθμό μετάδοσης $\frac{M}{16}$ bpp.

Για την φάση της δημιουργίας του codebook ισχύει ότι και για το προηγούμενο σχήμα σε ότι αφορά το μέγεθος του training set και την τιμή της παραμέτρου εκμάθησης. Η παράμετρος N_c τέθηκε στην τιμή 400, εφόσον εδώ έχουμε διανύσματα τα οποία οι διαστάσεις τους είναι διπλάσιες από αυτές του προηγούμενου σχήματος. Στα σχήματα 6.10 – 6.12 βλέπουμε τα αποτελέσματα που μας έδωσε το σχήμα αυτό, ενώ στα 6.12 – 6.14 τα αποτελέσματα με εικόνες στων οποίων το bit plane εφαρμόσαμε median filtering. Τέλος δοκιμάσαμε το σχήμα έχοντας σαν BTC τον AMBTC που περιγράψαμε στο κεφάλαιο 4. Δηλαδή στην περίπτωση αυτή στο διάνυσμα b έχουμε την τιμή γ αντί για την τυπική απόκλιση σ. Τα αποτελέσματα φαίνονται στα σχήματα 6.15 – 6.17.



Σχήμα 6.8 Ο VQ-BTC κωδικοποιήτής

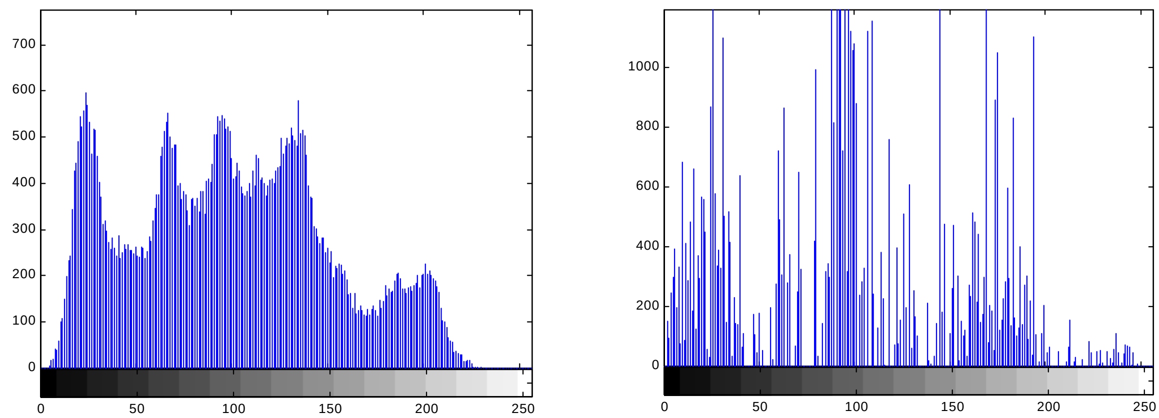


Σχήμα 6.9 Ο VQ-BTC αποκωδικοποιητής

Σχήμα 6.10 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook μεγέθους 128 (0.4375 bpp)

Σχήμα 6.11 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook μεγέθους 256 (0.5 bpp)

Σχήμα 6.12 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook
μεγέθους 512 (0.6255 bpp)



Σχήμα 6.13 Τα ιστογράμματα της αρχικής και ανακατασκευασμένης LENA με codebook μεγέθους 256 του VQ-BTC σχήματος.

Στο σχήμα 6.13 βλέπουμε το ιστόγραμμα της ανακατασκευασμένης Lena για μέγεθος codebook 256. Όπως παρατηρούμε οι τιμές των εντάσεων των pixel της εικόνας είναι αρκετά λιγότερες από αυτές της αρχικής εικόνας. Αυτό είναι αναμενόμενο, διότι τώρα έχουμε ταυτόχρονη κβάντιση όλων των χωρικών χαρακτηριστικών της εικόνας. Αυτό σημαίνει ότι το εύρος των τιμών των εντάσεων των pixel ορίζεται από το μέγεθος του codebook και φυσικά από τις τιμές των διανυσμάτων του. Στην προηγούμενη περίπτωση η κβάντιση του bit plane γινόταν ξεχωριστά από τις άλλες δύο παραμέτρους. Έτσι η κατανομή των ιστογράμματων των παραγόμενων εικόνων μπορούσε να είναι ομοιόμορφη και περισσότερο κοντά. Κάτι που δεν ισχύει στην περίπτωση αυτή.

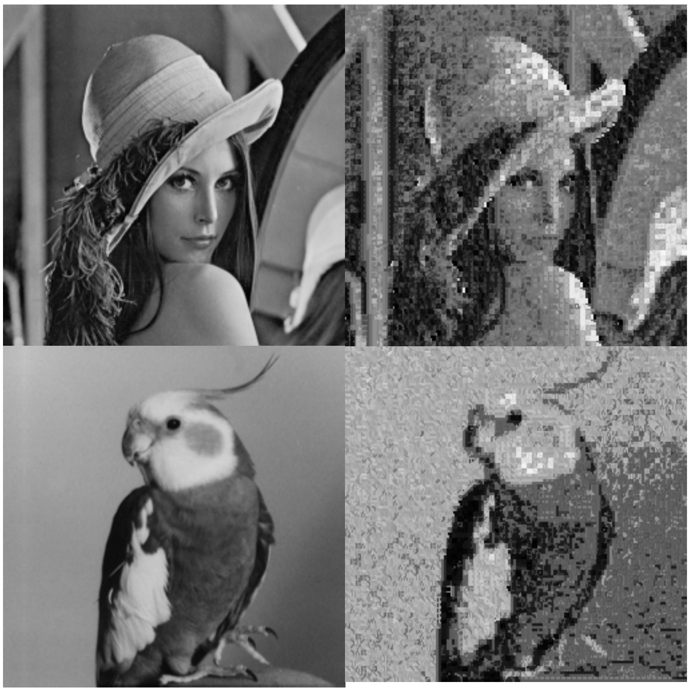
Σ' όλα τα σχήματα χρησιμοποιήσαμε εξαντλητικό ψάξιμο (exhaustive search) για βέλτιστα αποτελέσματα. Άλλωστε η εύρεση του κατάλληλου διανύσματος γινόταν αρκετά γρήγορα, αφού και τα codebook είναι σχετικά μικρά, όπως επίσης και οι διαστάσεις των διανυσμάτων. Στον πίνακα 6.3 βλέπουμε τους χρόνους αυτούς για τις εικόνες των δοκιμών.

	Lena	Bird
128	5.1 sec	4.9 sec
256	9.3 sec	9.6 sec
512	18.9 sec	18.6 sec

Πίνακας 6.3 Οι χρόνοι αποκωδικοποίησης των εικόνων των δοκιμών

Για το σχήμα AMBTC-VQ δοκιμάσαμε και κάποιο είδος τοπικού ψαξίματος (local search). Αυτό που κάναμε είναι να βρούμε κάποιο όριο για την ευκλείδια απόσταση ανάμεσα στο input vector και στο winner vector του codebook. Έτσι τώρα δεν ψάχνουμε όλο το codebook για το διάνυσμα κοντινότερο, στο διάνυσμα εσόδου, διάνυσμα στο, αλλά το πρώτο διάνυσμα του codebook που έχει απόσταση μικρότερη από το dianysma εισόδου. Οι χρόνοι που πήραμε φαίνονται στον πίνακα 6.6, ενώ το μέσο απόλυτο σφάλμα των εικόνων αυτών στον 6.7. Να σημειωθεί ότι οι χρόνοι αυτοί πάρθηκαν από τις προσομοιώσεις σε υπολογιστή pentium 200 MHz με μνήμη 32 MB.

Σχήμα 6.13 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook μεγέθους 128 (0.4375 bpp) και χρήση median filtering



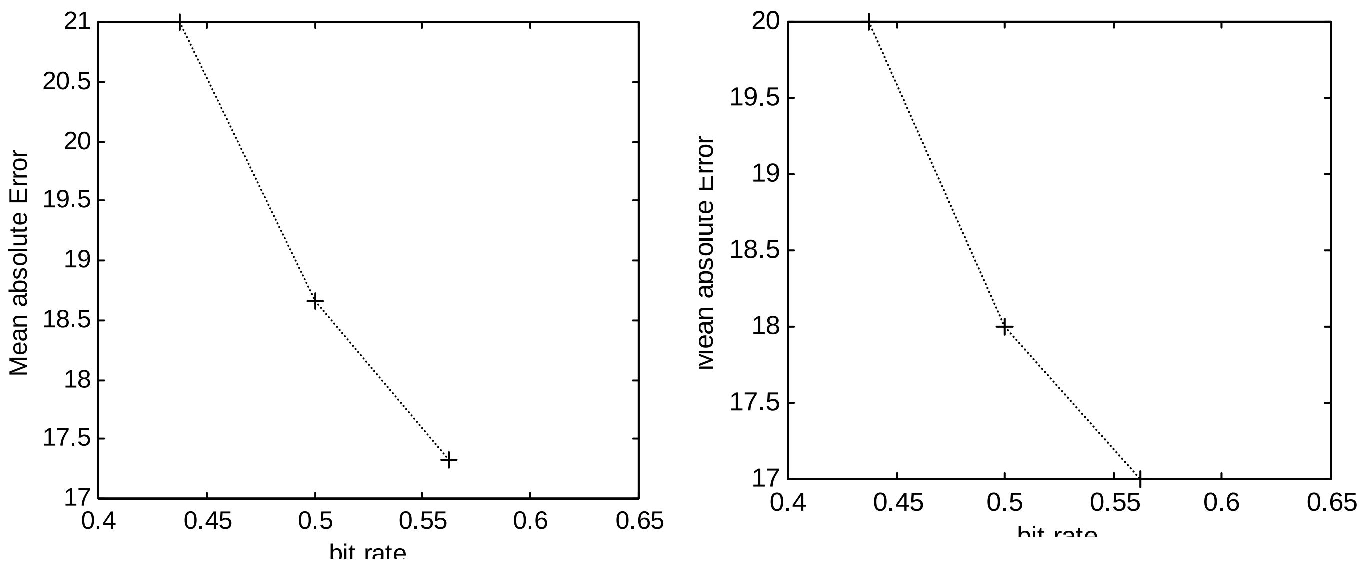
Σχήμα 6.14 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook μεγέθους 256 (0.5 bpp) και χρήση median filtering

Σχήμα 6.15 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook
μεγέθους 512 (0.5625 bpp) και χρήση median filtering

Όπως εύκολα μπορεί κάποιος να διαπιστώσει κοιτώντας τις εικόνες, η ποιότητα τους βελτιώνεται καθώς ο ρυθμός μετάδοσης αυξάνει, κάτι πολύ φυσιολογικό. Όπως βελτιώνεται και με την χρήση του median filtering, εφόσον πλέον οι δυνατοί συνδυασμοί για το bit plane είναι πολύ λιγότεροι από 2^{16} που είναι πριν το φιλτράρισμα. Στον πίνακα 6.3 βλέπουμε το μέσο απόλυτο σφάλμα για τις εικόνες των δοκιμών, ενώ στο σχήμα 6.15 τα διαγράμματα ρυθμού μετάδοσης σε σχέση με το μέσο απόλυτο σφάλμα.

	Lena	Lena median	Bird	Bird median
128	23	20	20	20
256	20	18	18	18
512	18	17	17	17

Πίνακας 6.4 Το MAE των εικόνων των δοκιμών



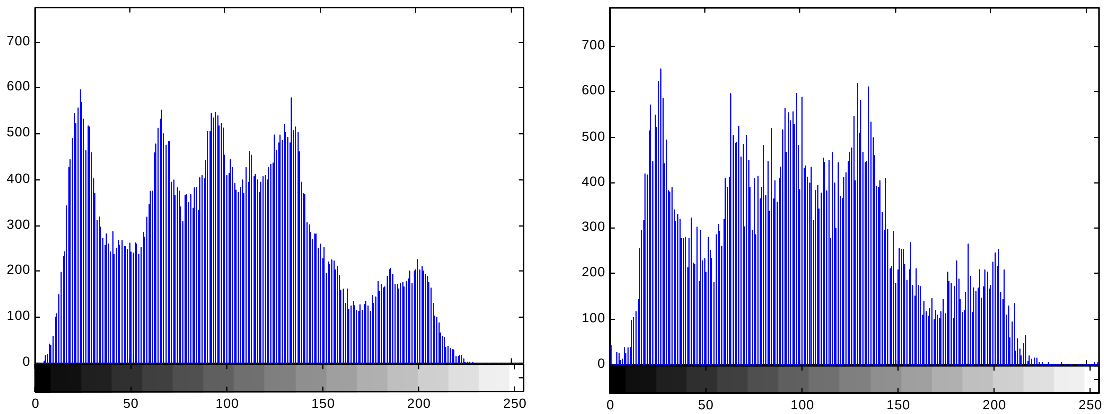
Σχήμα 6.15 Διαγράμματα μέσου απόλυτου σφάλματος σε σχέση με τον ρυθμό μετάδοσης

Στα επόμενα σχήματα φαίνονται τα αποτελέσματα που πήραμε με την χρησιμοποίηση του AMBTC αλγορίθμου στην θέση του βασικού BTC. Οι εικόνες που περιμένουμε να πάρουμε θα πρέπει να έχουν μικρότερο σφάλμα, διότι ο αλγόριθμος αυτός υπολογίζει παραμέτρους για αυτόν ακριβώς τον σκοπό.

Σχήμα 6.17 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook
μεγέθους 128 (0.4375 bpp) και χρήση AMBTC

Σχήμα 6.18 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook
μεγέθους 256 (0.5 bpp) και χρήση AMBTC

Σχήμα 6.18 Αριστερά οι αρχικές εικόνες των δοκιμών.
Δεξιά οι εικόνες μετά την κωδικοποίηση με codebook
μεγέθους 512 (0.5625 bpp) και χρήση AMBTC



Σχήμα 6.19 Τα ιστογράμματα της αρχικής και ανακατασκευασμένης
LENA με codebook μεγέθους 256 του AMBTC-VQ σχήματος.

Τα αποτελέσματα του τελευταίου κωδικοποιητή είναι κάτι παραπάνω από ικανοποιητικά. Οι διαφορές της κωδικοποιημένης εικόνας από την αρχική που μπορεί να παρατηρήσει κάποιος είναι ελάχιστες και εστιάζονται κυρίως στις ακμές των εικόνων. Ο απλός VQ-BTC σε πολλές περιπτώσεις είχε το πρόβλημα ότι σε ομοιόμορφες περιοχές της εικόνας μπορούσε κανείς να ξεχωρίσει τα κωδικοποιημένα block. Στον VQ-AMBTC κάτι τέτοιο πλέον δεν υφίσταται, ενώ όπως φαίνεται και στον πίνακα 6.5 το απόλυτο μέσο σφάλμα είναι πολύ μικρότερο, κάτι το οποίο το είχαμε προβλέψει.

	Lena	Bird
128	8	4
256	8	3.5
512	7.5	3.4

Πίνακας 6.5 Το MAE για τον VQ-AMBTC

	Lena	Bird
128	4.3 sec	4.1 sec
256	6.8 sec	5.3 sec
512	9.4 sec	7.0 sec

Πίνακας 6.6 Οι χρόνοι κωδικοποίησης με local search.

	Lena	Bird
128	8.5	3.8
256	8.1	3.6
512	7.9	3.5

Πίνακας 6.7 Το MAE για τον AMBTC-VQ με την χρήση local search

Όπως μπορούμε να παρατηρήσουμε η χρήση του τοπικού ψαξίματος του codebook δεν επιφέρει μεγάλη επιπλέον παραμόρφωση, ενώ μειώνει αρκετά τον χρόνο κωδικοποίησης (για μέγεθος codebook 512 η μείωση είναι πάνω από 50%). Άρα η χρήση του στο σχήμα αυτό θεωρείται επιβεβλημένη.

6.4 Συμπεράσματα

Στο κεφάλαιο αυτό παρουσιάσαμε τρία σχήματα του VQ-BTC κωδικοποιητή. Συγκρίνοντας τα με έναν απλό VQ βλέπουμε ότι υπερτερούν στην ταχύτητα κωδικοποίησης λόγω του μικρού μεγέθους των διανυσμάτων εισόδου. Από την άλλη στην σύγκριση με τον απλό BTC έχουμε συμπιέσεις πολύ υψηλότερες. Ο συνδιασμός όμως των δύο αυτών αλγορίθμων έχει αποτελέσματα, τα οποία αφήνουν αρκετές υποσχέσεις για τον σχεδιασμό κωδικοποιητή, ο οποίος να ανταπεξέρχεται στις απαιτήσεις συστημάτων μετάδοσης εικόνων σε πραγματικό χρόνο.

Στις προσωμοιώσεις των τριών προτεινόμενων σχημάτων μπορούμε να κάνουμε τις εξής παρατηρήσεις:

a. Στο πρώτο σχήμα η ποιότητα των ανακατασκευασμένων εικόνων είναι αρκετα ικανοποιητική αλλά μειωνεκτεί στους μικρούς ρυθμούς μετάδοσης, οι οποίοι κυμαίνονται πάνω από το 1 bpp.
b. Το δεύτερο σχήμα πετυχαίνει τους επιθημητούς ρυθμούς μετάδοσης, ωστόσο η ποιότητα των εικόνων κυμαίνεται σε μέτρια επίπεδα.
c. Τα πλεονεκτήματα των δυο προηγουμένων σχημάτων φαίνεται να συγκεντρωνει το τρίτο σχήμα. Δηλαδή υψηλούς ρυθμούς μετάδοσης με ικανοποιητική ποιότητα των κωδικοποιημένων εικόνων.

Αναφορές

[1] Y. Linde, M. Buzo and R. M. Gray, “An algorithm for Vector Quantizer Design”, IEEE Trans. on Communications, vol COM-28, No 1, pp. 84-95, Jan. 1980.

[2] R. Dony and S. Haykin, “Neural Network Approaches to Image Compression”, Proceedings of the IEEE, pp. 208 –303, Feb. 1995.

[3] R. Lancini, S. Tubaro, “Adaptive Vector Quantization for Picture Coding using Neural Networks”, IEEE Trans. on Communications, vol. 43, No 2/3/4, pp. 534-544, Feb./Mar./Apr. 1995.

Μερικές από τις εικόνες του training set

7 Σχολιασμός και επεκτάσεις

7.1 Γενικά συμπεράσματα

Γενικά οι τεχνικές συμπίεσης εκμεταλλεύονται δύο παράγοντες: τον στατιστικό πλεονασμό ανάμεσα στα pixel και τους ψυχοφυσικούς περιορισμούς των ανθρώπινων παρατηρητών. Αυτό προκύπτει από την υψηλή χωρική συσχέτιση χαρακτηριστικών φωτογραφιών με φυσικές σκηνές, ενώ το δεύτερο τονίζει το ότι πληροφορία η οποία δεν μπορεί να γίνει αντιληπτή από τον χρήστη δεν χρειάζεται να μεταδοθεί. Οι κωδικοποιητές εικόνων ανήκουν σε μια από τις τρεις παρακάτω κατηγορίες: τους κωδικοποιητές χωρικού πεδίου, τους κωδικοποιητές στο πεδίο της συχνότητας και τους υβριδικούς κωδικοποιητές. Γενικά οι κωδικοποιητές στο πεδίο της συχνότητας πετυχαίνουν μεγαλύτερους λόγους συμπίεσης, με κόστος το μεγάλο υπολογιστικό φορτίο και μερικές φορές χρειάζεται το πέρασμα δύο ή τριών φορών της αρχικής εικόνας για την ολοκλήρωση της κωδικοποίησης. Από την άλλη πλευρά, οι κωδικοποιητές που δουλεύουν στο χωρικό πεδίο συνήθως προσφέρουν περιορισμένη συμπίεση αλλά έχουν ένα σαφώς μικρότερο υπολογιστικό φορτίο και συνήθως απαιτούν ένα πέρασμα των αρχικών δεδομένων. Με βάση της παραπάνω σύγκριση, οι υβριδικοί κωδικοποιητές βρίσκουν μια μέση βάση ανάμεσα στις χωρικές και συχνότητας μεθόδους, όσον αφορά την ποιότητα και το υπολογιστικό φορτίο.

Η μέθοδος που αναπτύχθηκε στην εργασία αυτή οδηγεί σε έναν κωδικοποιητή που εργάζεται στο χωρικό πεδίο και ο οποίος είναι αρκετά ελκυστικός μετάδοση εικόνων σε πραγματικό χρόνο εξαιτίας της απλότητάς και της απόδοσής του. Η μέθοδος διατηρεί σημαντικά χαρακτηριστικά της εικόνας όπως απότομες ακμές και ξαφνικές gray level αλλαγές, ενώ βάζοντας διαδοχικά από τον BTC coder έναν vector quantizer πετυχαίνει υψηλούς λόγους συμπίεσης, χωρίς να αυξάνει σημαντικά ο χρόνος κωδικοποίησης, εξαιτίας των μικρών διαστάσεων των διανυσμάτων εισόδου. Στην γενική περίπτωση τα κωδικοποιημένα από τον BTC χαρακτηριστικά της εικόνας διατηρούνται ακόμα και όταν τα δεδομένα εξόδου του BTC τροφοδοτούνται στον vector quantizer παράγοντας έναν μειωμένο ρυθμό κωδικοποίησης. Επιπλέον πλεονεκτήματα είναι το εξαιρετικά μικρό υπολογιστικό φορτίο, η απλότητα και οι μικρές απαιτήσεις μνήμης και αποθήκευσης.

Έγιναν κάποιες τροποποιήσεις στον βασικό BTC αλγόριθμο για βελτίωση της απόδοσής του, ενώ με την συνύρπαξη με τον vector quantizer πετυχαίνονται λόγοι συμπίεσης γύρω στο 0.5 bpp.

7.2 Περαιτέρω εργασίες και επεκτάσεις του συστήματος

Αυτή η παράγραφος περιέχει περιλιπτική αναφορά για περαιτέρω εργασίες, οι οποίες δεν περιέχονται στην εργασία αυτή, αλλά τα οποία είναι άξια προσοχής. Σε γενικές γραμμές θα ασχοληθούμε με την κωδικοποίηση έγχρωμων εικόνων όπως επίσης και με την συμπίεση ακολουθία εικόνων (video).

Η συμπίεση 24 bpp εγχρώμων εικόνων μπορεί προσεγγιστικά να θεωρηθεί σαν συμπίεση πολλαπλών grayscale εικόνων, οι οποίες είτε συμπιέζονται συνολικά μια την φορά ή συμπιέζονται σε διαδοχικά 4x4 block. Μια άλλη προσέγγιση είναι η χρήση των διαστημάτων χρώματος (color space) όπως YUV, CIELUV, τα οποία απεικονίζουν την χρωματική πληροφορία σε δύο συνιστώσες και την grayscale πληροφορία σε μία τρίτη και είναι περισσότερο αποτελεσματικό για συμπίεση από ότι διαστήματα όπως RGB. Σε αυτή την περίπτωση οι έγχρωμες εικόνες μετασχηματίζονται από την RGB απεικόνιση στην YIQ luminance και chrominance απεικόνιση. Στην συνέχεια το luminance συμπιέζεται με την χρήση του BTC-VQ σχήματος. Εναλλακτικά κάποιος θα μπορούσε να κωδικοποιήσει κάθε από τα R, G, B σήματα και μετά να κωδικοποιήσει τα άλλα δύο στοιχεία (όπου είναι δυνατό), χρησιμοποιώντας μετασχηματισμούς παρεμφερής με αυτούς που χρησιμοποιούνται για την κωδικοποίηση του πρώτου στοιχείου.

Για κωδικοποίηση video, επιπλέον μείωση του bit rate θα μπορούσε να επιτευχθεί με την εκμετάλλευση του interframe πλεονασμού, εφόσον η συσχέτιση μεταξύ συνεχόμενων frame σε μια σειρά εικόνων σε μια σκηνή είναι πολύ υψηλή. Η γενική τάση είναι, ότι κάποιος μπορεί να εκμεταλλευτεί την υψηλή συσχέτιση μεταξύ διαδοχικών frame με την χρήση interframe predictor για να απαλείψει τον inteframe πλεονασμό.Τότε μπορεί να εφαρμοστεί BTC-VQ intraframe κωδικοποίηση στα παραγόμενα παραμένων frame, πετυχαίνοντας μια ικανοποιητική συνολική κωδικοποίηση, η οποία εξαρτάται από την συνδυασμένη απόδοση του interframe predictor και του intraframe αποκωδικοποιητή.