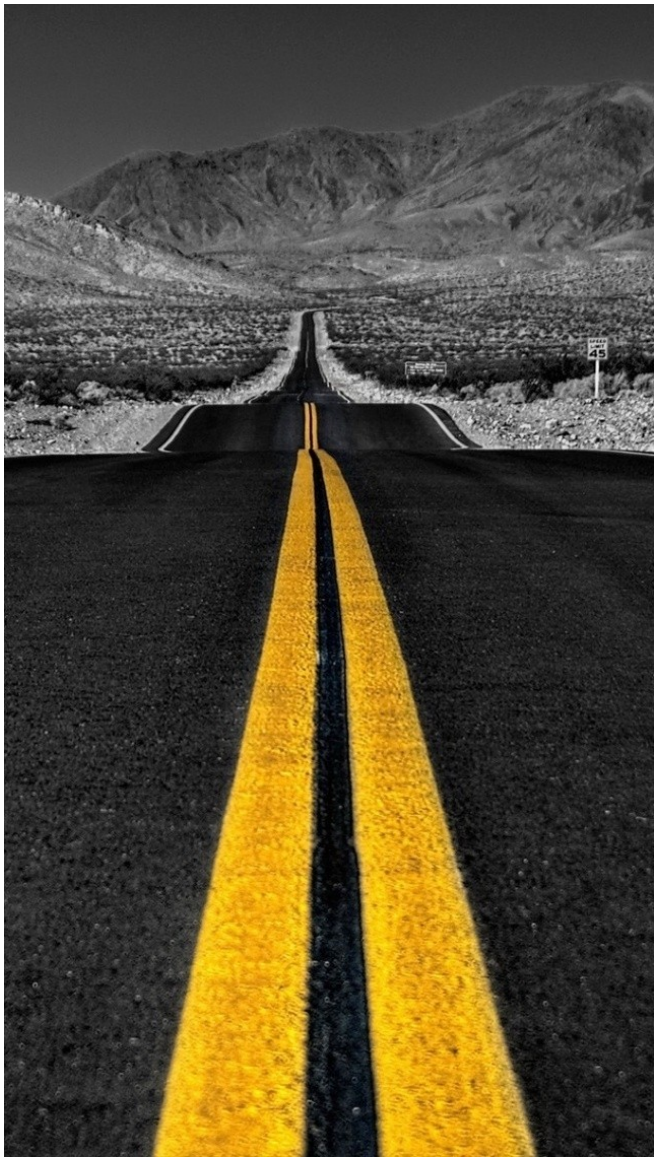


GRAYSCALE ENCODER

ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ

Τεχνολογία Πολυμέσων | 8/1/2019

Μέλη Ομάδας:

Γεωργίου Γεώργιος-Κρίτων (p3150020)

Μέμτσας Αθανάσιος(p3150103)

Πάνος Ευάγγελος (p3150134)

ΕΠΙΛΟΓΗ ΘΕΜΑΤΟΣ

Από το αρχείο Θέματα_Εργασιών_2018-2019.pdf επιλέξαμε το δεύτερο θέμα προς υλοποίηση, δηλαδή την δημιουργία ενός κωδικοποιητή grayscale σύμφωνα με το πρότυπο JPEG που έχει τις παρακάτω λειτουργίες:

- Διάβασμα RAW εικόνων (στις οποίες οι τιμές φωτεινότητας είναι αποθηκευμένες σε binary format, με μέγεθος ένα byte για κάθε pixel)
- Μετασχηματισμός DCT των μπλοκ 8x8 της παραπάνω εικόνας.
- Κβάντιση των συντελεστών DCT με τον default πίνακα κβάντισης του προτύπου JPEG.
- Δυνατότητα επιλογής καλύτερης ή χειρότερης ποιότητας.

Η παραπάνω υλοποίηση πραγματοποιήθηκε στην γλώσσα C++

ΜΕΘΟΔΟΛΟΓΙΑ

Για την δημιουργία του προαναφερόμενου κωδικοποιητή χρησιμοποιήθηκε και η εξωτερική βιβλιοθήκη TinyTIFF για την διαχείριση των .tif raw εικόνων. Στον κωδικά μας υπάρχουν:

- ***Main.cxx*** είναι υπεύθυνη για τους ελέγχους των ορισμάτων που δέχεται το πρόγραμμα κατά την εκκίνηση καθώς και για την κλήση των απαραίτητων μεθόδων σύμφωνα με λειτουργία που επέλεξε ο χρήστης είτε μέσω των ορισμάτων είτε μέσω της κονσόλας.
- ***Parser.h/Parser.cpp*** με την βοήθεια της TinyTIFF πραγματοποίει το «διάβασμα» και το «γράψιμο» των εικόνων μαζί με όλους τους απαραίτητους ελέγχους για την αποφυγή λαθών.
- ***DCT.h/DCT.cpp*** Περιέχουν υλοποίηση για κλάση υπεύθυνη για τον μετασχηματισμό των μπλοκ 8x8 της εικόνας.
- ***Quantization.h/Quantization.cpp*** Περιέχουν υλοποίηση για κλάση που πραγματοποιεί την κβάντιση των συντελεστών DCT σύμφωνα με το πρότυπο JPEG και ορίζει την ποιότητα της εικόνας.
- ***Util.h*** Συντελείται από όλες τις υπόλοιπες απαραίτητες μεθόδους χρειαστήκαμε στο αρχείο Main.cxx όπως την μετατροπή των pixel της εικόνας από 8-bit σε double, την δημιουργία του 8x8 παραθύρου, την πρόσθεση του παραθύρου στην εικόνα και τον υπολογισμό του Peak Signal-to-Noise Ratio (PSNR).

ΧΡΗΣΗ

Οι παράμετροι μπορούν να δοθούν είτε σαν arguments μέσω του command line, είτε διαδραστικά αφότου αν δεν δοθούν σαν arguments. Σε οποιαδήποτε περίπτωση χρειάζονται 3 παράμετροι. Η πρώτη είναι πάντα το mode. Τα υποστηριζόμενα modes είναι 'e' (encoding), 'd' (decoding), 'p' (PSNR Calculation) και 'a'(όλα τα παραπάνω).

Σε περίπτωση encoding /decoding , το δεύτερο argument είναι το Quality Scaling Factor (περισσότερες πληροφορίες για αυτό δίνονται παρακάτω). Το τρίτο argument είναι το path για την εικόνα στην οποία θα γίνει encoding/decoding. Προφανώς στο decoding πρέπει να δωθεί ως input μία εικόνα που έχει βγει ως αποτέλεσμα του encoding.

Στην περίπτωση όπου γίνεται PSNR calculation, το δεύτερο argument είναι το path για την approximation εικόνα (προηγούμενο αποτέλεσμα του αλγορίθμου) ενώ το τρίτο είναι η 'ground truth' εικόνα (uncompressed original εικόνα).

Σε περίπτωση 'a', δηλαδή όλων των επιλογών , το δεύτερο argument είναι το Quality Scaling Factor και το τρίτο argument είναι το path για την εικόνα στην οποία θα γίνει encoding/decoding.

Τα paths μπορούν να είναι είτε full paths είτε, αν το αρχείο είναι στο ίδιο directory με το executable, τότε απλά το όνομα του αρχείου. Υπό Windows ένα παράδειγμα χρήσης (από το ίδιο directory με το executable) θα ήταν:

JPEG_ENC.exe a 1.5 input_image.tif

Επιπρόσθετα έχουμε δημιουργήσει και συμπεριλάβει ένα script, το οποίο, αν τρέξει από το path του executable, με την προϋπόθεση ότι αυτός ο φάκελος περιέχει υποστηριζόμενες εικόνες, τρέχει το πρόγραμμα για μία εικόνα με πολλές διαφορετικές τιμές του QSF. Ένα παράδειγμα χρήσης για το script θα ήταν:

JPEG_RUN.bat input_image.tif

ΑΝΑΛΥΣΗ

Ακολουθεί αναλυτική επεξήγηση όλων των μεθόδων και λειτουργιών

Αν τα ορίσματα δεν δωθούν σαν arguments μέσω του command line, ζητείται από τον χρήστη η εισαγωγή ενός γράμματος μέσω της κονσόλας. Το προγράμματος μας πραγματοποιεί έλεγχο των ορισμάτων για να διαπιστώσει αν ο χρήστης έχει δηλώσει ποια λειτουργία επιθυμεί να χρησιμοποιήσει.

Με τα απαραίτητα μηνύματα που εμφανίζονται στην κονσόλα ο χρήστης καθοδηγείται στην εισαγωγή του ονόματος της εικόνας καθώς και το ground truth image αν επιλεχθεί PSNR calculation.

Αν επιλεχθεί encoding ή το decoding της εικόνας τότε ο χρήστης μπορεί να επιλέξει ποιότητα. Η ποιότητα (Quality Scaling Factor) είναι μία παράμετρος η οποία πολλαπλασιάζεται με κάθε στοιχείο του Quantization Table (η default τιμή λοιπόν είναι 1 και διατηρεί το original quantization table). Καλύτερη διατήρηση της πληροφορίας του σήματος μπορεί να επιτευχθεί με μικρότερες τιμές του Quality Scaling Factor (π.χ. 0.5), ενώ λιγότερη με μεγάλες τιμές (π.χ. 3). Οι μικρές τιμές οδηγούν σε καλύτερη διατήρηση της εικόνας αλλά λιγότερη δυνατότητα για συμπίεση, ενώ οι μεγάλες το αντίθετο. Αφού μπορεί να ελεγχθεί το κατά πόσον μπορεί να συμπιεστεί η εικόνα, μπορεί να ελεγχθεί και το bit rate με το οποίο θα στέλναμε τις εικόνες αν τις μεταδίδαμε. Αν η σύνδεση ήταν κακή, μπορούσαμε να ανεβάσουν το Quality Scaling Factor στις επόμενες εικόνες για να επιτύχουμε καλύτερο compression και να στείλουμε την εικόνα με λιγότερα bits.

Σημειώνεται ότι στο decoding δίνεται ως input μια εικόνα που είναι αποτέλεσμα του encoding. Κατά το encoding παράγεται μία εικόνα η οποία περιέχει σε κάθε pixel το quantized αποτέλεσμα του DCT. Αυτό το αποθηκεύουμε πάλι ως tif εικόνες (αλλά δίνεται προσοχή να μην χάνονται οι αρνητικές τιμές).

Έχοντας όλα τα απαραίτητα χρησιμοποιούμε τις εξής μεθόδους (βαθύτερο documentation υπάρχει πάνω από τον ορισμό της κάθε μεθόδου στον κώδικα). Σε αυτές σημειώνεται ότι Image είναι std::vector< std::vector< std::bitset<8> > > και DImage είναι std::vector< std::vector<double> >.

- Image Parser::ReadImage(std::string)
  για το διάβασμα των εικόνων.
- DImage Util::ConvertImageToDoubleImage<T>(Image)
  για την μετατροπή των εικόνων.

PSNR CALCULATION

Στην περίπτωση της λειτουργίας PSNR calculation χρησιμοποιείται η μέθοδος

double Util::PSNR(DImage , DImage)

για τον υπολογισμό του PSNR και έπειτα εκτυπώνεται το αποτέλεσμα στην κονσόλα.

ENCODING & DECODING

Στην περίπτωση της λειτουργίας του encoding η decoding, γίνεται χρήση είτε της προκαθορισμένης εικόνας είτε αυτής που εισήγαγε ο χρήστης.

Μετά την μετατροπή της εικόνας σε τύπο DImage γίνεται χρήση των μεθόδων

- void DCT::CalculateCosines() και void DCT::CalculateCoefficients() για τους υπολογισμούς των τιμών με βάση την μεθοδολογία του αλγορίθμου DCT
- void DCT::SetQualityScalingFactor(double qualityScalingFactor) για την ανάθεση της ποιότητα της εικόνας.

Encoding

Στην περίπτωση του encoding πραγματοποιείται διπλή for loop για την κάλυψη όλων τον pixel της εικόνας και έπειτα γίνεται χρήση των

- DImage Util::CalculateWindow(DImage, int, int, int, int) για την δημιουργία του παραθύρου 8x8.
- Util::ShiftFor(DImage, int shiftFor) για την μετατόπιση του παραθύρου κατά 'shiftFor'.
- DCT::Transform(DImage) για την μετατροπή του παραθύρου σε DCT εικόνα.
- Quantization::Quantize(DImage) για την κβαντοποίηση της DCT εικόνας.
- void Util::AddWindowToImage(DImage, DImage, int, int, int, int) για την δημιουργία της τελικής εικόνας από την DCT εικόνας.
- Image Util::ConvertDoubleImageToImage(DImage) για την μετατροπή της εικόνας από DImage (double) σε Image.
- και τέλος void Parser::WriteImage(Image, std::string, int, int) για το γράψιμο της Image στο αρχείο.

Decoding

Στην περίπτωση του decoding πραγματοποιείται επίσης διπλή for loop για όλα τα pixel της εικόνας καθώς και χρήση των

- Util::CalculateWindow(DImage, int, int, int, int) για δημιουργία του παραθύρου 8x8.
- Quantization::Dequantize(DImage) για αποκβαντοποίηση του παραθύρου.
- DCT::InverseTransform(DImage) για την αντιστροφή μετατροπή της DCT εικόνας.
- Util::ShiftFor(DImage, shiftFor) για την μετατόπιση του παραθύρου κατά shiftFor.
- Util::AddWindowToImage(DImage, DImage, int, int, int, int) για την δημιουργία της τελικής εικόνας από την DCT εικόνας.
- Util::ConvertDoubleImageToImage(DImage) για την μετατροπή της εικόνας από DImage (double) σε Image.
- και τέλος Parser::WriteImage(Image, std::string, int, int) για το γράψιμο της Image στο αρχείο.

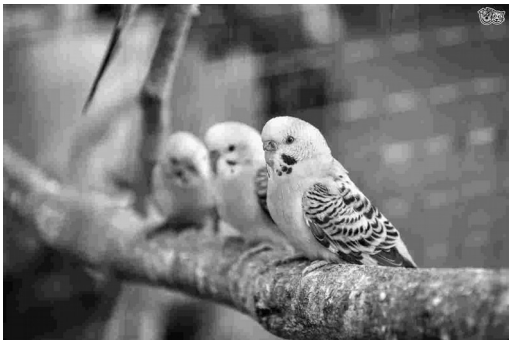
ΠΑΡΑΔΕΙΓΜΑΤΑ

Original Picture: budgies.tif

*QSF=Quality Scaling Factor

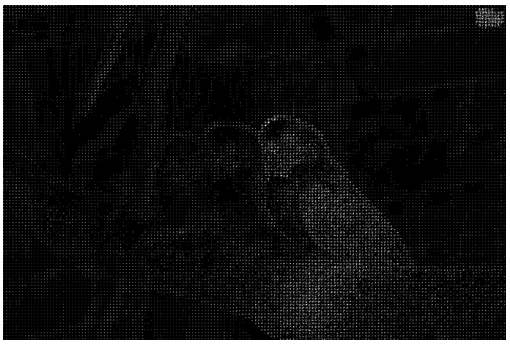
QSF=0.5 PSNR = 47.5181

QSF=1 PSNR = 40.6731



QSF=5 PSNR = 33.1118

QSF=0.5 encoded file



QSF=1 encoded file

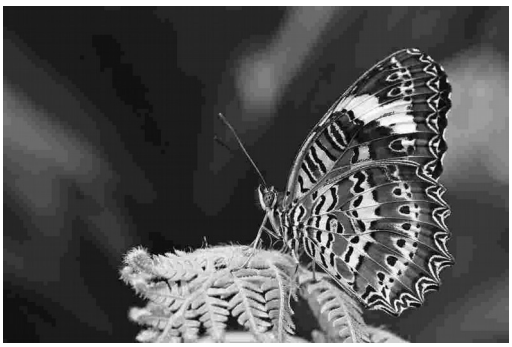
QSF=5 encoded file

Original Picture: butterfly.tif

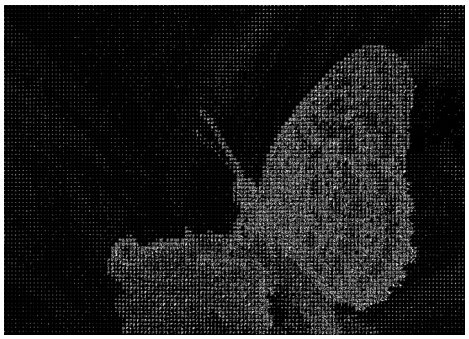
*QSF=Quality Scaling Factor

QSF=0.5 PSNR = 37.772

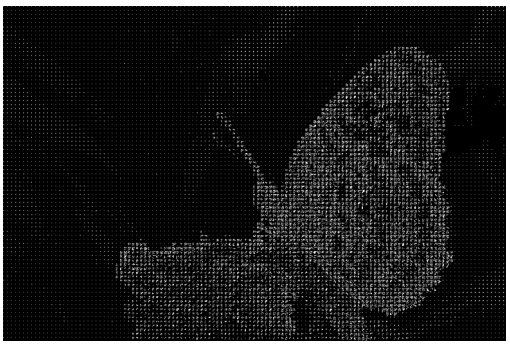
QSF=1 PSNR = 33.9594



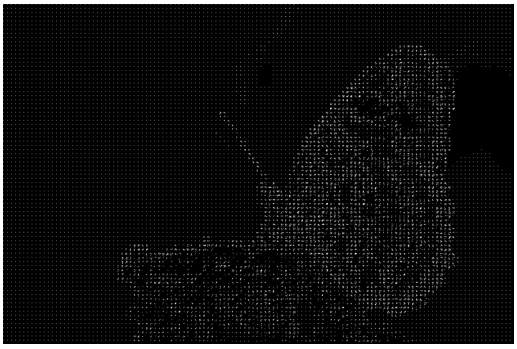
QSF=5 PSNR = 28.1332



QSF=0.5 encoded file



QSF=1 encoded file



QSF=5 encoded file

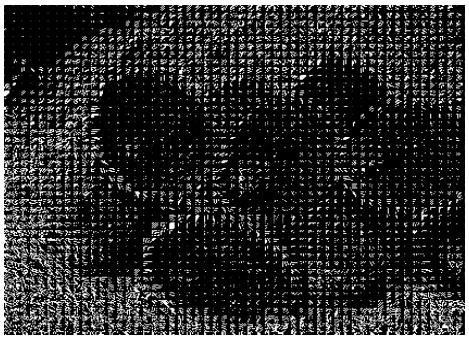
*QSF=Quality Scaling Factor

Original Picture: eggs.tif

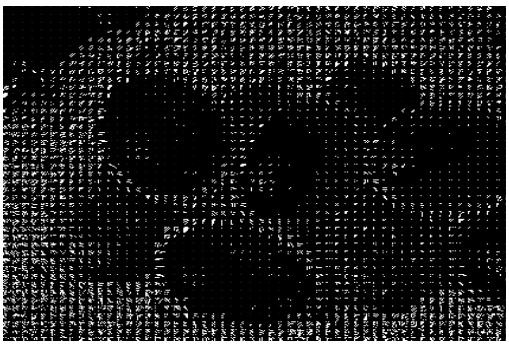
QSF=0.5 PSNR = 20.082

QSF=1 PSNR = 19.878

QSF=5 PSNR = 19.0602



QSF=0.5 encoded file



QSF=1 encoded file

QSF=5 encoded file

ΣΥΜΠΕΡΑΣΜΑΤΑ

QSF	bokeh	budgies	butterfly	cell	eggs	sailboat	shark	sheep	taj	town	AVERAGE	MEDIAN	MIN	MAX
0.5	39.03	47.60	37.85	41.07	28.81	36.72	38.31	28.31	35.16	48.24	38.10963	38.08315	28.3054	48.2382
1	36.38	40.72	34.06	39.11	27.70	31.98	35.95	25.76	32.46	30.39	33.45115	33.25925	25.7641	40.7186
1.5	35.01	39.03	32.38	37.95	26.95	30.71	34.76	24.64	31.01	29.56	32.19988	31.6971	24.6366	39.0253
2	34.08	37.80	31.35	36.91	26.54	29.92	33.98	23.92	30.08	28.20	31.27901	30.7175	23.9228	37.8045
2.5	33.34	36.64	30.59	36.13	26.09	29.32	33.35	23.46	29.36	27.64	30.59278	29.9792	23.4615	36.6421
3	32.75	35.77	29.99	34.89	25.72	28.87	32.84	23.09	28.79	26.94	29.96675	29.43145	23.0903	35.7741
3.5	32.21	35.01	29.48	34.58	25.45	28.49	32.40	22.81	28.30	26.50	29.52315	28.98935	22.808	35.0058
4	31.73	34.34	29.04	34.42	25.12	28.17	32.02	22.57	27.87	26.01	29.12941	28.6034	22.5701	34.4227
4.5	31.31	33.74	28.62	34.06	24.85	27.90	31.66	22.34	27.49	25.66	28.76362	28.25785	22.3406	34.0597
5	30.94	33.16	28.25	33.60	24.68	27.64	31.32	22.15	27.15	25.30	28.41994	27.94505	22.1532	33.6008

Πειραματικά δεδομένα με διαφορετικό Quality Scaling Factor(0.5-5)

Παρατηρούμε ότι η κωδικοποίηση jpeg με χαμηλό quality scaling factor (δηλαδή καλή ποιότητα) δίνει πολύ καλά αποτελέσματα που είναι δύσκολο να ξεχωρίσουν από την αρχική εικόνα. Αυτό φαίνεται και στο PSNR (Peak Signal-to-Noise Ratio).

Γενικώς παρατηρήσαμε ότι παρόλο που το PSNR βοηθάει στο να κατανοήσουμε αν μια εικόνα έχει αποκατασταθεί καλά μετά το encoding, είναι ένα μετρικό που βοηθάει περισσότερο στο να συγκρίνουμε αποκαταστάσεις της ίδιας εικόνας παρά διαφορετικών εικόνων μεταξύ τους. Αυτό οφείλεται στην φύση της ανθρώπινη όρασης (αυτό που θεωρούμε καλό μπορεί να μην είναι το "μαθηματικά" καλύτερο) αλλά και στη ποιοτητα και στο μέγεθος της αρχικής εικόνας και ίσως και σε πράγματα όπως η αντίθεση και το βάθος χρώματος της.

Ακόμα πρέπει να τονίσουμε ότι έχει σημασία και η χρήση. Παράδειγματος χάρη μια ιστοσελίδα μπορεί να προτιμάει μεγαλύτερη συμπίεση (και άρα χειρότερη ποιότητα) για να φορτώνει πιο γρήγορα και να είναι πιο ελαφριά.

Γενικως προτείνουμε quality scaling factor X για ιστοσελίδες και quality scaling factor Y για ικανοποιητική διατήρηση της λεπτομέρειας της εικόνας. Υπενθυμίζεται ότι χαμηλές τιμές του quality scaling factor οδηγούν σε μεγαλύτερη συμπίεση.

ΒΙΒΛΙΟΓΡΑΦΙΑ & ΒΙΒΛΙΟΘΗΚΕΣ

- Για τον σκοπό της καλύτερης διαχείρισης των αρχείων .tiff raw format χρησιμοποιήθηκε η εξωτερική βιβλιοθήκη TinyTIFF.

- https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/dct.htm

- Για την επιλογή του format .tiff για τις raw εικόνες λήφθηκε υπόψιν το συγκεκριμένο άρθρο από το Wikipedia.org που χαρακτηρίστηκα αναφέρει:
«Many raw file formats, including IIQ (Phase One), 3FR (Hasselblad), DCR, K25, KDC (Kodak), CRW CR2 CR3 (Canon), ERF (Epson), MEF (Mamiya), MOS (Leaf), NEF (Nikon), ORF (Olympus), PEF (Pentax), RW2 (Panasonic) and ARW, SRF, SR2 (Sony), are based on the TIFF file format».

- Πηγή κατανόησης και επαναχρησιμοποίησης τεχνικών και μεθόδων ήταν οι διαφάνειες του μαθήματος της Τεχνολογίας Πολυμέσων.

- Το προκαθορισμένο τραπέζι κβαντοποίησης που χρησιμοποιήθηκε πάρθηκε αυτούσιο από το αντίστοιχο άρθρο που αφορά Quantization.