Επεξεργασία και Ανάλυση Εικόνας

Κεπεσίδης Παναγιώτης aivc22008

## Περιγραφή

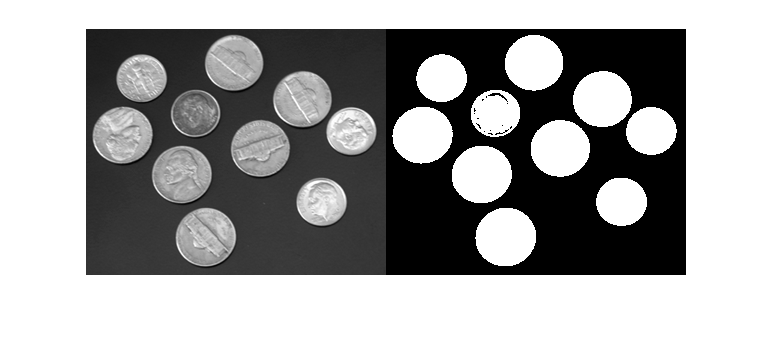
Στην εργασία αυτή υλοποιούνται τεχνικές σε περιβάλλον MATLAB που αφορούν την κατωφλίωση grayscale εικόνων για την μετατροπής τους σε δυαδικές. Συγκεκριμένα υλοποιούνται μια τεχνική σε global επίπεδο με την μέθοδο του Otsu και μια τεχνική τοπικής προσαρμοσμένης κατωφλίωσης υπολογίζοντας την μέση, διάμεση και ενδιάμεση τιμή ενός παραθύρου.

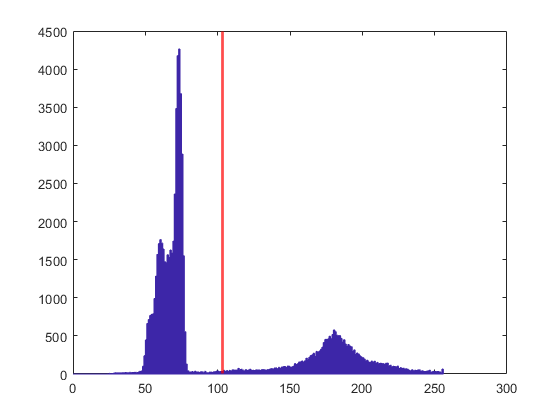
## Α Μέρος

Για το πρώτο μέρος υλοποιήθηκε η μέθοδος Otsu ώστε να βρεθεί η graylevel τιμή που αντιστοιχεί σε κατώφλι με βάση το οποίο η εικόνα θα μετατραπεί σε δυαδική. Αρχικά δέχεται σαν είσοδο μία grayscale εικόνα και γίνεται αρχικοποίηση του T που είναι η τιμή threshold για τους υπολογισμούς. Ο αλγόριθμος ξεκινά με μια επαναληπτική διαδικασία για Τ από 0-255 που εξετάζει όλες τις τιμές threshold. Μέσα σε αυτή την επαναληπτική διαδικασία γίνεται ο υπολογισμός της πιθανότητας της κάθε κλάσης ( , ), ο υπολογισμός της μέσης τιμής για κάθε κλάση ( , ) και την διασπορά της κάθε κλάσης ( , ). Τέλος υπολογίζει την συνολική διασπορά εντός των κλάσεων και την αποθηκεύει στον πίνακα all\_s. Αφού τελειώσει η επαναληπτική διαδικασία για κάθε Τ, ελέγχεται ο πίνακας all\_s και επιστρέφεται η θέση στην οποία βρίσκεται η μικρότερη τιμή . Η θέση αυτή είναι η θέση που θα τοποθετηθεί το Τ για να έχουμε τη βέλτιστη κατωφλίωση στην εικόνα.

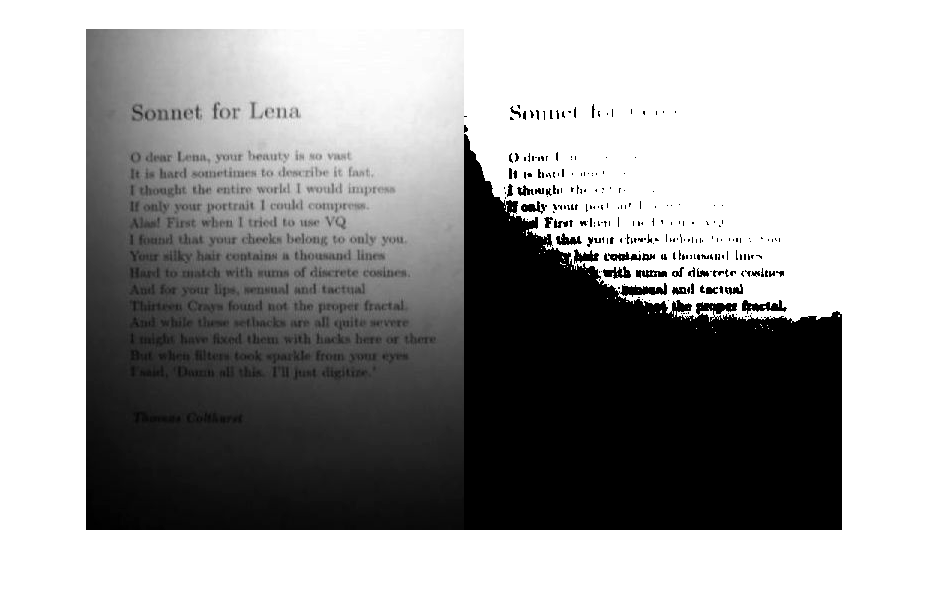
Η συγκεκριμένη μέθοδος εφαρμόστηκε σε διάφορες εικόνες που είχαν κλιμάκωση φωτεινότητας αλλά και σε εικόνες χωρίς κλιμάκωση όπως φαίνεται παρακάτω:

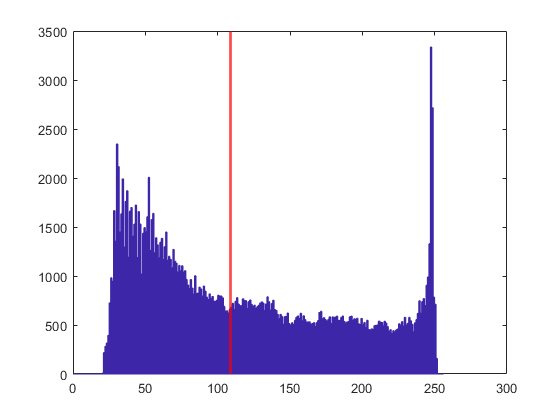
1. Coins Τ= 0.40234 (time= 0.011530sec)





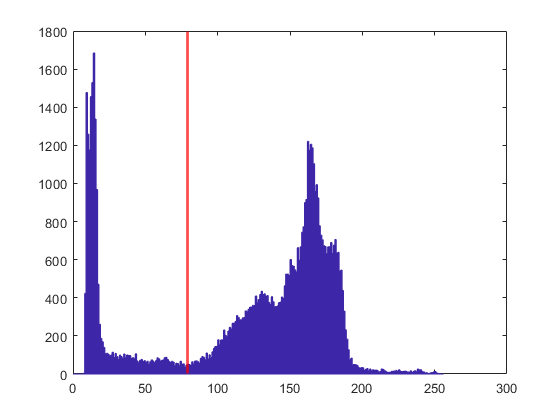
1. Sonet T= 0.42578 (time= 0.1942 sec)





1. Cameraman T= 0.30859 (time= 0.006855)





## Δεύτερο Μέρος

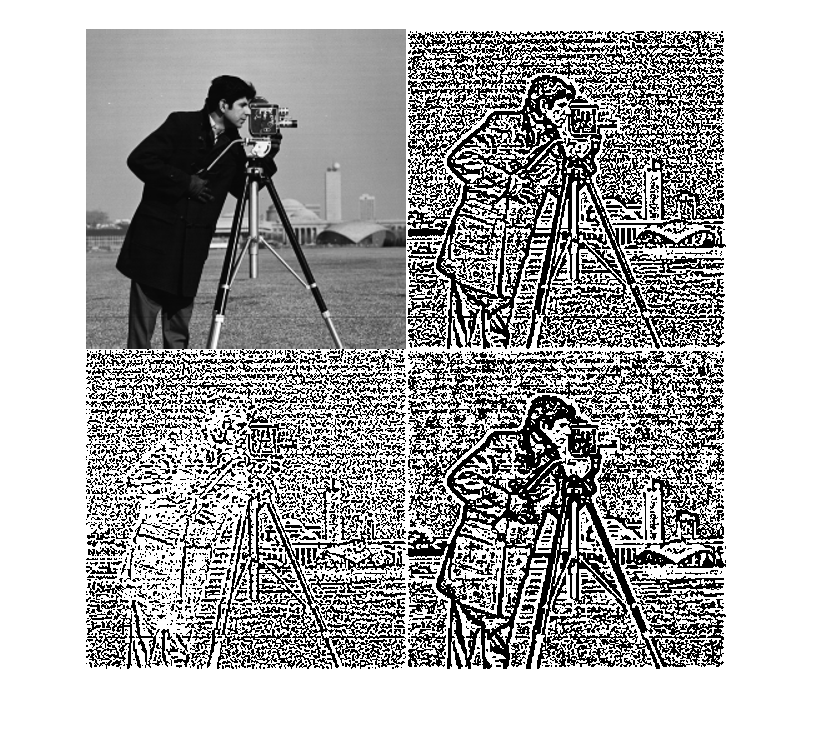
Στο δεύτερο μέρος έγινε υλοποίηση της προσαρμοσμένης κατωφλίωσης με τρεις τεχνικές επιλογής κατωφλίου: μέση, διάμεση και ενδιάμεση graylevel τιμή. Αρχικά ο αλγόριθμος δέχεται σαν είσοδο μία εικόνα, μία τιμή R που είναι το μέγεθος του παραθύρου που θα εφαρμοστεί και μία θετική τιμή C που βελτιώνει τα αποτελέσματα σε περιοχές με ομοιογενή φωτεινότητα. Το φίλτρο που χρησιμοποιείται υλοποιείται με την συνάρτηση colfilt() που παίρνει σαν ορίσματα έναν πίνακα που περιέχει την τιμή κάθε pixel της εικόνας με τις ίδιες διαστάσεις, τις διαστάσεις του παραθύρου, είδος του block και μία συνάρτηση που εφαρμόζεται στα στοιχεία του παραθύρου αφού έχει μετατραπεί σε στήλη επιστρέφοντας έναν πίνακα με τα αποτελέσματα αυτών των πράξεων.

Σε σχόλια μέσα στον κώδικα έχουν κρατηθεί και συναρτήσεις που δημιουργούν το παράθυρο και εφαρμόζουν τις απαραίτητες συναρτήσεις με ίδια αποτελέσματα, αλλά για λόγους ομοιογένειας διατηρήθηκε η colfilt().

Για κάθε φωτογραφία εφαρμόζονται διαδοχικά και οι τρεις τεχνικές και χρονομετρούνται μόνο οι πράξεις που γίνονται ξεχωριστά. Για την σύγκριση της μεθόδου θα χρησιμοποιηθεί μια φωτογραφία με degradation και μία χωρίς.

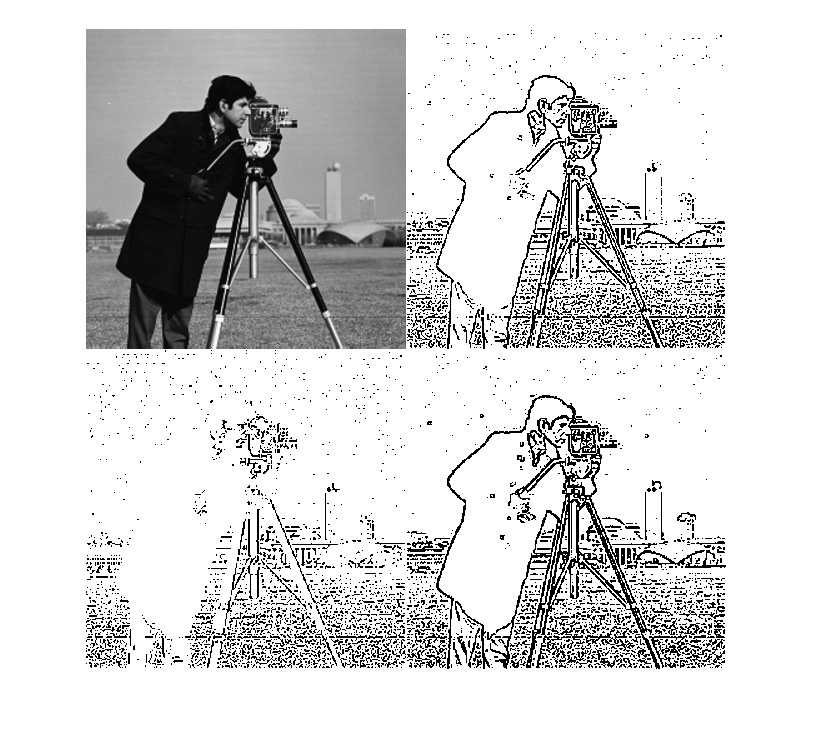
Πρώτα θα εφαρμοστεί σε μία φωτογραφία χωρίς degradation, στην εικόνα cameraman.tif:

* Τα αποτελέσματα με μέγεθος παραθύρου 5 και C=0,1 καταλήγουν σε εικόνες που δεν είναι «καθαρές». Οι χρόνοι υπολογισμών είναι με την σειρά που εμφανίζονται οι εικόνες 0.175, 0.052, 0.049.

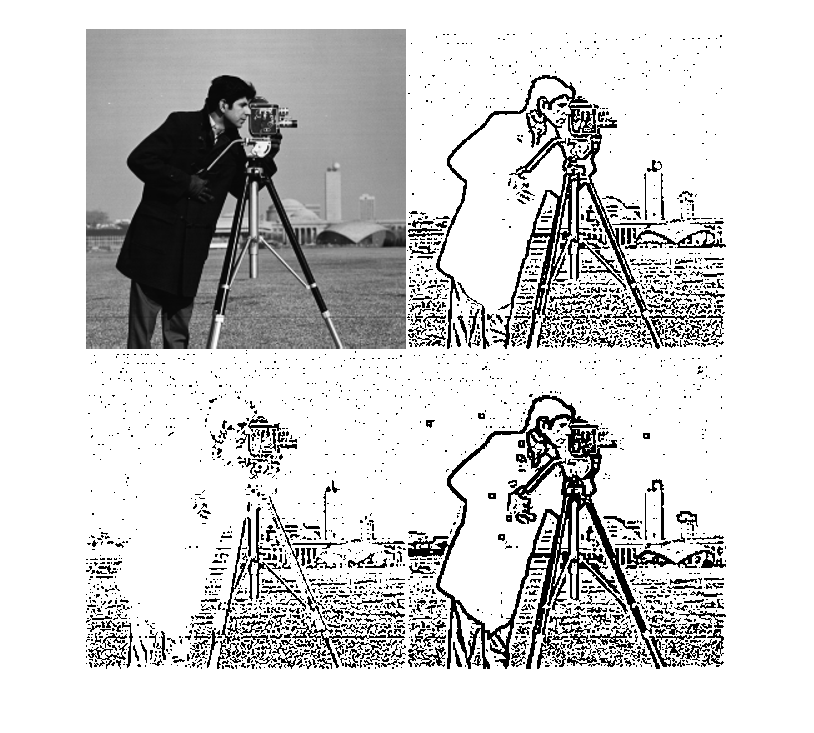


Εικόνα 1 Εμφανίζεται η αρχική εικόνα, δεξιά το mean, κατω αριστερα το median και κάτω δεξια η ενδιάμεση τιμή.

* Αλλάζοντας το C σε μεγαλύτερο νούμερο (5) και R=3 παρατηρούμε ότι εμφανίζονται τα περιγράμματα των αντικειμένων αλλά λεπτομέρειες όπως το χέρι του φωτογράφου χάνονται . Με καλύτερο αποτέλεσμα να το εμφανίζει με την μέση και την ενδιάμεση τιμή και σε χρόνους 0.010, 0.009, 0.014.



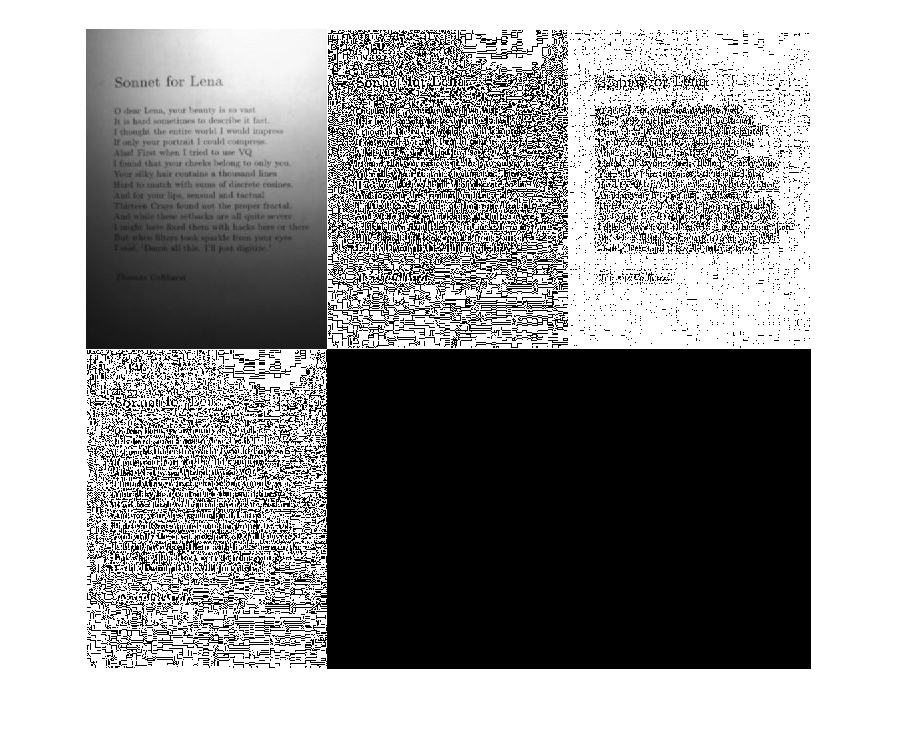
* Για να διακρίνονται με αρκετή λεπτομέρεια τα αντικείμενα που απεικονίζονται εκτελούμε τις πράξεις με C=5 και R=5. Οι διαδικασίες για κάθε τεχνική διαρκεί 0.015, 0.018, 0.032.



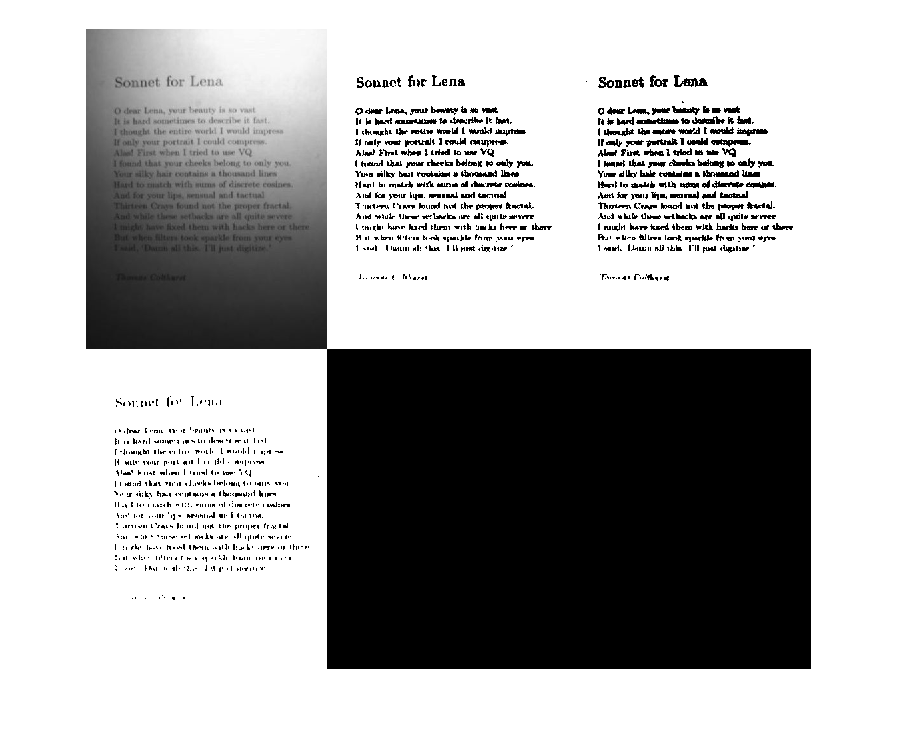
Παρατηρώντας τα αποτελέσματα και των τριών τεχνικών γίνεται ξεκάθαρο ότι με την μέση και ενδιάμεση τιμή έχουμε καλυτέρα αποτελέσματα καθώς τόσο ο φωτογράφος όσο και η κάμερα διακρίνονται πολύ πιο εύκολα.

Εφαρμόζοντας την ίδια μέθοδο και τις ίδιες τεχνικές και σε φωτογραφία με degradation όπως η φυτογραφία με το κείμενο (sonnet.jpg) παρατηρούμε ότι δεν μπορούμε να χρησιμοποιήσουμε τα ίδια νούμερα για το C και το R όπως προηγουμένως.

* Για C=0.1 και μέγεθος παραθύρου R=3 η εικόνα είναι πολύ χειρότερη από την αρχική και δεν διακρίνεται κανένα στοιχείο. Χρόνοι υπολογισμού 0.021, 0.024, 0.048.



* Η συγκεκριμένη εικόνα είναι αρκετά δύσκολο να επεξεργαστεί έτσι ώστε να εμφανίζει τα γράμματα του κειμένου καθαρά. Όμως με C=10 και R=25 το κείμενο ξεχωρίζει και τα γράμματα είναι πιο ευδιάκριτα χωρίς όμως να επιτυγχάνεται η ανάγνωση του κειμένου. Οι χρόνοι όμως είναι 1.231, 1.515, 2.348. Πολύ μεγαλύτεροι από τις προηγούμενες δοκιμές.



**Ερώτηση:**

Με βάση των δοκιμών που έγιναν στις παραπάνω εικόνες γίνεται σαφές ότι η προσαρμοσμένη κατωφλίωση έχει πολύ καλύτερα και ξεκάθαρα αποτελέσματα σε εικόνες με κλιμάκωση φωτεινότητας, ενώ η μέθοδος του Otsu εμφανίζει τα αντικείμενα που είναι το κυρίως θέμα πολύ καλύτερα με μεγάλη λεπτομέρεια και γρηγορότερα. Έτσι πρέπει να εξετάζεται η εφαρμογή και με τις δυο μεθόδους ανάλογα με το είδος της εικόνας και την λεπτομέρεια που ζητάμε. Μια καθολική εφαρμογή της προσαρμοσμένης κατωφλίωσης μπορεί να κοστίζει σε πόρους (χρονοβόρα) και να οδηγεί σε μη επαρκή αποτελέσματα.