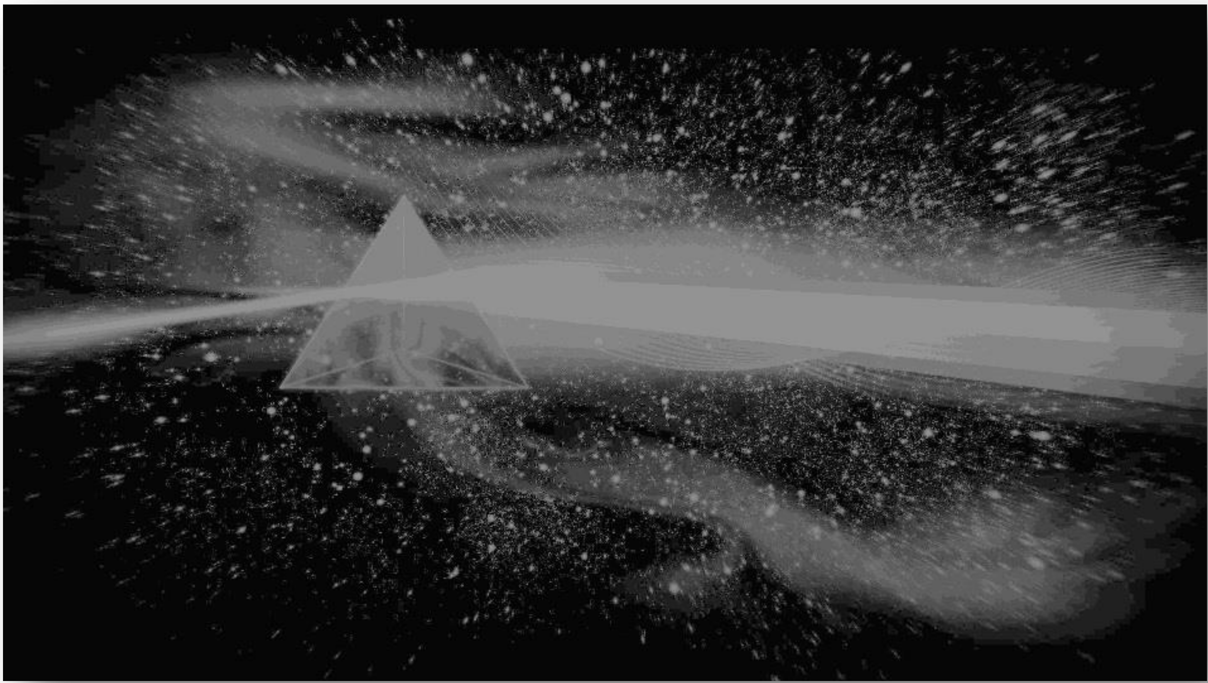
Ψηφιακή Επεξεργασία Εικόνας
-Εργασία 2-

—

Σημειακός μετασχηματισμός και διόρθωση Ιστογράμματος

—

Σημειακός μετασχηματισμός
Μετασχηματισμός βάση ιστογράμματος
Μετασχηματισμός βάση κατανομής



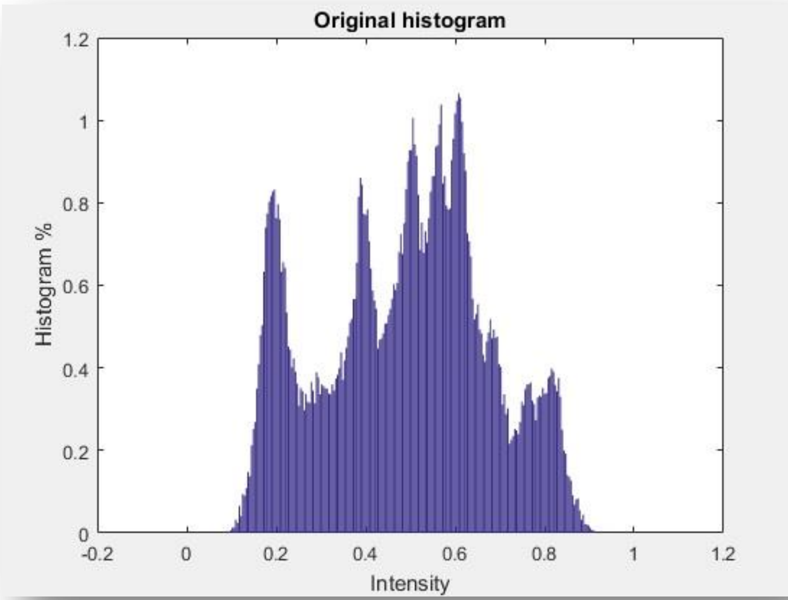
Μάστορας Ραφαήλ Ευάγγελος
7918

21/4/2017

Περιγραφή της εργασίας

Στην εργασία αυτή μας ζητήθηκε να δημιουργήσουμε πρόγραμμα στο matlab, με το οποίο θα γίνεται η επεξεργασία μιας grayscale εικόνας, βάση συγκεκριμένου σημειακού μετασχηματισμού ή κάποιου δοθέν ιστογράμματος και τέλος βάση γνωστής κατανομής. Το πρώτο ζητούμενο είναι η δημιουργία συνάρτησης Y = pointtransform(X, x1, y1, x2, y2) η οποία για δοθέν σημεία (x1,y1) και (x2,y2) παράγει στην έξοδο εικόνα της οποίας οι τιμές ακολουθούν τις ευθείες που ορίζουν τα σημεία (0,0) και (x1,y1) , (x1,y1) και (x2,y2), (x2,y2) και (1,1). Για αυτό το ζητούμενο επίσης μας ζητήθηκε η παραγωγή ασπρόμαυρης εικόνας, μέσω του μετασχηματισμού αυτού. Ως δεύτερο ζητούμενο της εργασίας να δημιουργήσουμε την συνάρτηση Y = histtransform(X, h, v) η οποία μετασχηματίζει την εικόνα εισόδου X στην εικόνα εξόδου Y έτσι ώστε το ιστόγραμμα της Y να προσεγγίζει όσο καλύτερα γίνεται το ιστόγραμμα που περιγράφεται από τα h και v, μέσω ενός greedy αλγορίθμου. Τέλος ζητήθηκε η δημιουργία συνάρτησης h = pdf2hist(d, f), η οποία υπολογίζει τις τιμές του ιστογράμματος h στα διαστήματα που ορίζει το d βάση του function pointer f. Για τα παραπάνω δημιούργησα ένα αρχείο matlab με τίτλο main.m που καλείς τις συναρτήσεις που θα επιλέξει ο χρήστης, μέσω ενός “menu”.

Στα παρακάτω ιστογράμματα με μπλέ μπάρες φαίνεται το ποσοστό των Pixel της εικόνας με την συγκεκριμένη φωτεινότητα. Ενώ με κόκκινη γραμμή και κυκλάκια φαίνεται το ποσοστό της εκάστοτε κατανομής ή του δοθέν ιστογράμματος για συγκρεκιμένη την συγκεκριμένη τιμή.



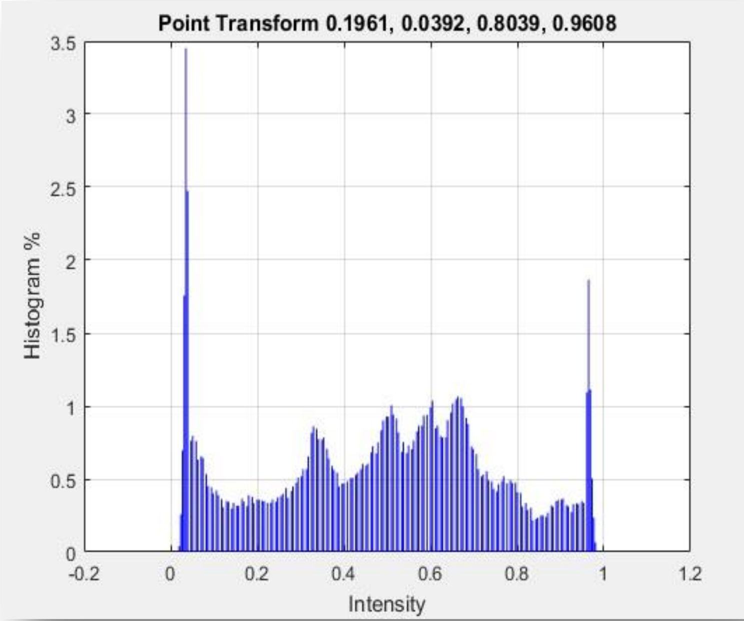
Αρχική εικόνα σε grayscale

Σημειακός μετασχηματισμός

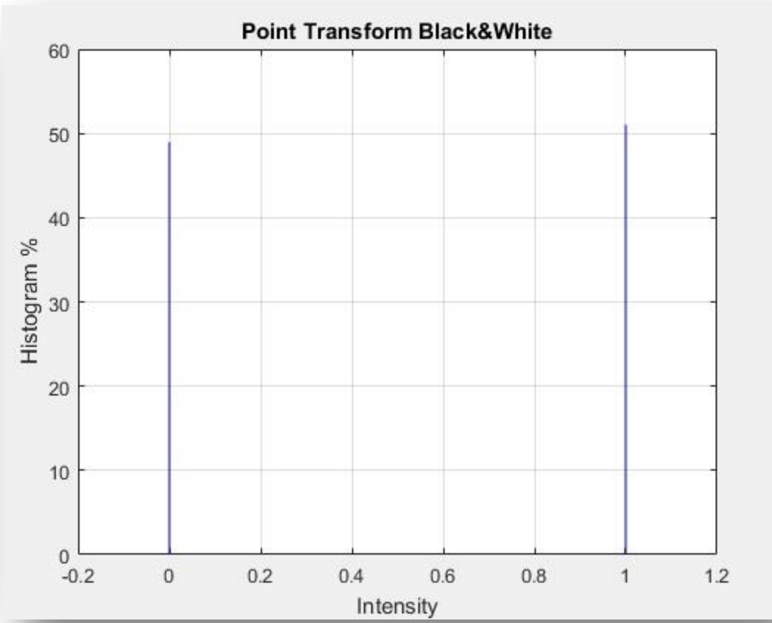
Στην συνάρτηση Y = pointtransform(X, x1, y1, x2, y2) τα pixel της εικόνας X που είναι ανάμεσα στο διάστημα [0,x1] μετασχηματίζονται σε τιμές πάνω στην ευθεία που ορίζουν τα σημεία (0,0) και (x1,y1). Αντίστοιχα τα pixel της εικόνας X που είναι ανάμεσα στα διαστήματα [x1,x2] και [x2,1] μετασχηματίζονται σε τιμές πάνω στις ευθείες που ορίζουν τα σημεία , (x1,y1) και (x2,y2), (x2,y2) και (1,1). Αυτό γίνεται μέσω της γνωστής εξίσωσης της ευθείας που διέρχεται από δυο σημεία :

$$\psi = \lambda(\chi - \chi 1) + \psi 1 \text{ όπου } \lambda = \frac{\psi 2-\psi 1}{\chi 2-\chi 1}$$

Τα αποτελέσματα για [x1, y1,x2, y2] = [0.1961, 0.0392, 0.8039, 0.9608]



Τα αποτελέσματα για [x1, y1,x2, y2]= [0.5, 0, 0.5, 1] παράγουν την ασπρόμαυρη εικόνα που ζητήθηκε.



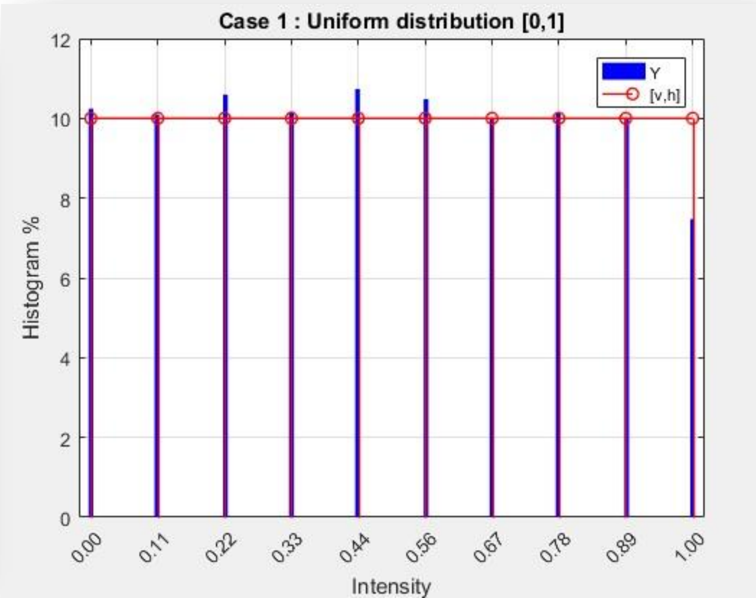
Μετασχηματισμός με βάση το ιστόγραμμα

Ο μετασχηματισμός με βάση το ιστόγραμμα γίνεται στην συνάρτηση Y = histtransform(X, h, v) , η οποία δίνει σαν έξοδο την εικόνα Y της οποίας το ιστόγραμμα προσεγγίζει το ιστόγραμμα που περιγράφεται απο τα h και v. Η προσέγγιση επιτυγχάνεται μέσω ενός greedy αλγορίθμου, ο οποίος μετασχηματίζει όλα τα pixel της ίδιας φωτεινότητας στην ίδια στάθμη v(i) και εφόσον ο αριθμός των pixel διά του συνολικού αριθμού των Pixel της εικόνας δεν ξεπερνάει το h(i) συνεχίζει για τις επόμενες τιμές φωτεινότητας. Αν ο αριθμός των μετασχηματισμένων pixel προς το συνολικό αριθμό των pixel ξεπεράσει το όριο h(i) τότε συνεχίζουμε τον μετασχηματισμό στην επόμενη στάθμη v(i+1). Ο έλεγχος αυτός όμως γίνεται αφού έχουν μετασχηματιστεί κάθε φορά όλα τα pixel της ίδιας φωτεινότητας.
Για να γίνει ο αλγόριθμος αυτός εύκολα υλοποιήσιμος ταξινομώ τον πίνακα της εικόνας X και κρατάω παράλληλα τις αρχικές θέσεις των pixel σε ένα πίνακα Index. Έτσι κάθε φορά που αλλάζω μία τιμή απο τον πίνακα X στον πίνακα Y αυτό γίνεται χρησιμοποιώντας το εκάστοτε index, προκειμένου να μπει στην σωστή θέση.

Τα αποτελέσματα για τις περιπτώσεις που ζητήθηκαν φαίνονται παρακάτω.

```
% Case 1
L = 10;
v = linspace (0, 1, L);
h = ones([1, L]) / L;
```

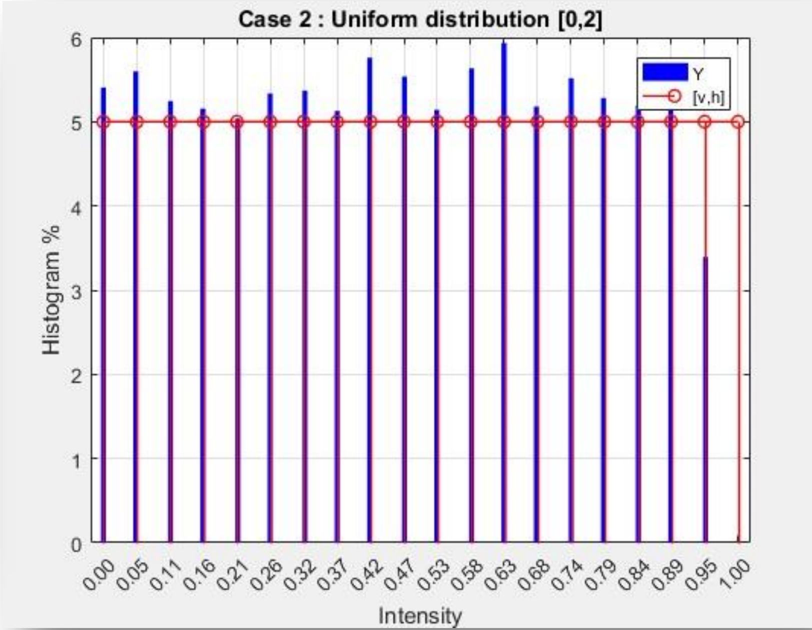
Τα h και v στην περίπτωση αυτή περιγράφουν μία ομοιόμορφη κατανομή στο [0,1]



Βλέπουμε οτι ο αλγόριθμος προσεγγίζει αρκετά καλά την κατανομή έχοντας λίγες τιμές με απόκλιση απο την δοθέν κατανομή.

% Case 2
L = 20;
v = linspace (0, 1, L);
h = ones([1, L]) / L;

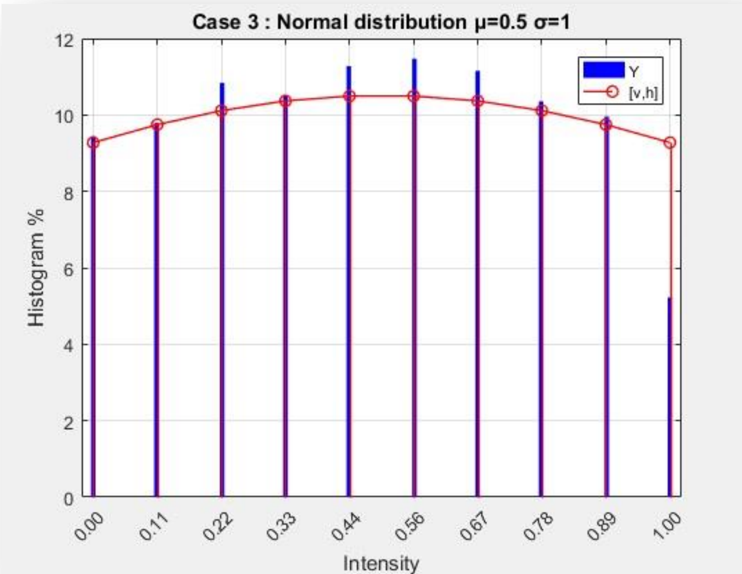
Τα h και v στην περίπτωση αυτή περιγράφουν μία ομοιόμορφη κατανομή στο [0,1]



Εδώ βλέπουμε οτι για αυτά τα διαστήματα ο αλγόριθμος προσεγγίζει λιγότερο καλά την κατανομή. Γενικά παρατηρήθηκε ότι όσο ανεβαίνει ο αριθμός των υποδιαστημάτων, η προσέγγισή του greedy αλγορίθμου αποκλίνει περισσότερο απο τις κατανομές.

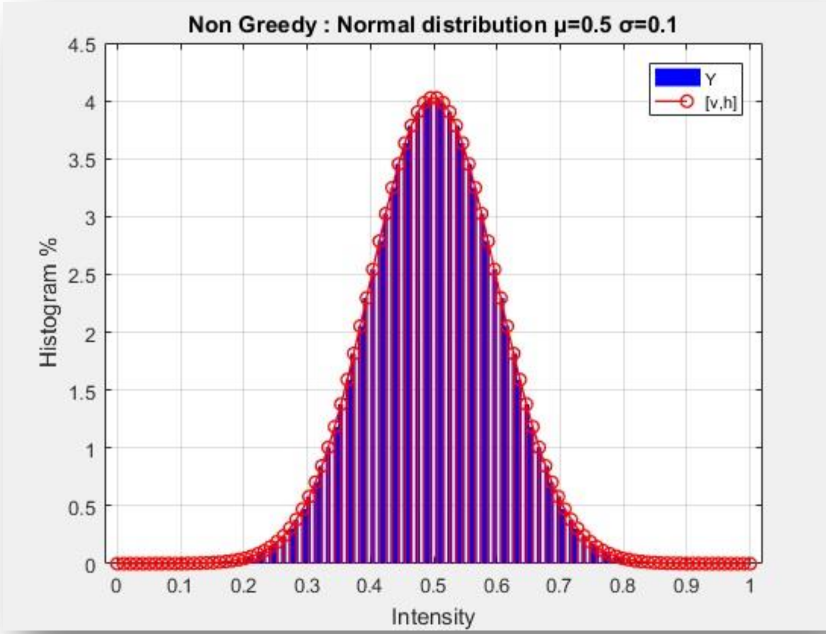
% Case 3
L = 10;
v = linspace (0, 1, L);
h = normpdf(v, 0.5) / sum(normpdf(v, 0.5));

Τα h και v στην περίπτωση αυτή περιγράφουν κανονική κατανομή με μ=0.5 και σ=1.



Και εδώ ο αλγόριθμος προσεγγίζει ικανοποιητικά την κατανομή , έχοντας πάλι όμως τιμές που αποκλίνουν.

Παρατηρούμε πως ο αλγόριθμος προσεγγίζει ικανοποιητικά τα δοθέν ιστογράμματα , ωστόσο μερικές τιμές αποκλίνουν από τις εκάστοτε τιμές του δοθέν ιστογράμματος. Αυτό οφείλεται στην φύση του greedy αλγορίθμου, δηλαδή στο γεγονός ότι τιμές με ίδια φωτεινότητα θα μπούνε στην ίδια στάθμη. Έτσι οι τελευταίες στάθμες καταλήγουν να έχουν λιγότερα pixel. Επίσης για μικρότερο αριθμό διαστημάτων, οι κατανομές προσεγγίζονται καλύτερα.
Χρησιμοποιώντας έναν non greedy αλγόριθμο μπορούμε να προσεγγίσουμε τέλεια το κάθε ιστόγραμμα.



Παράδειγμα ιστογράμματος με τέλεια προσέγγιση, χωρίς greedy αλγόριθμο

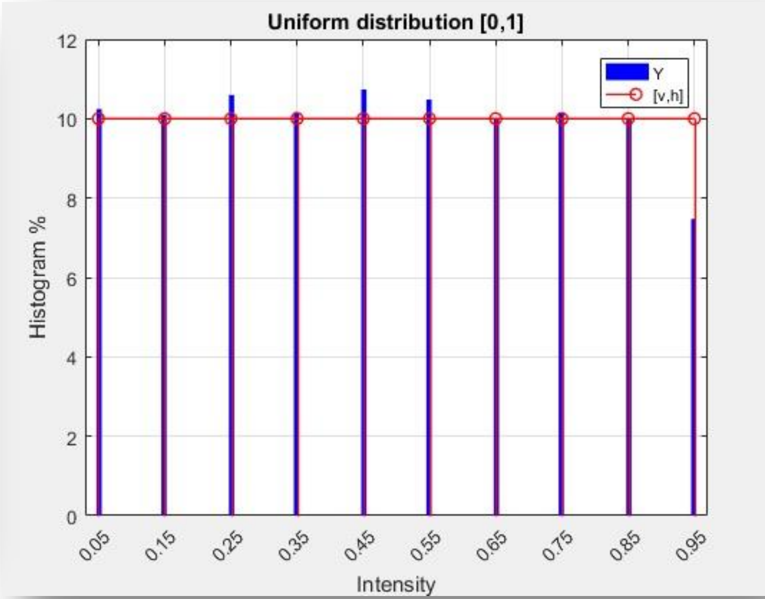
Εκτίμηση ιστογράμματος από πυκνότητα πιθανότητας κατανομής

Για το ζητούμενο αυτό υλοποιήθηκε η συνάρτηση h = pdf2hist(d, f) , η οποία σαν ορίσματα παίρνει το d που αντιπροσωπεύει τις τιμές διαδοχικών διαστημάτων και ως f την συνάρτηση κάποιας κατανομής. Συγκεκριμένα για την f δημιούργησα τρεις δοκιμάστικές συναρτήσεις, την uniformpdf1 που επιστρέφει ομοιόμορφη κατανομή στο [0,1] , την uniformpdf2 που επιστρέφει ομοιόμορφη κατανομή στο [0,2] και την normalpdf που επιστρέφει κανονική κατανομή με μέση τιμή 0.5 και τυπική απόκλιση 0.1. Αυτές οι συναρτήσεις δίνονται στην f μέσω του τελέστη @ (δηλαδή f=@uniformpdf) και η f δίνεται σαν όρισμα στην παραπάνω συνάρτηση pdf2hist. Για να βρούμε την τιμή του ιστογράμματος σε ένα διάστημα [d(i),d(i+1)] , ολοκληρώνουμε την συνάρτηση στο διάστημα που θέλουμε μέσω της συνάρτησης h(i)=integral(f,d(i),d(i+1)). Αυτό γίνεται για όλα τα δοθέν διαστήματα. Η αρχική μου υλοποίηση ήταν με symbolic functions, ωστόσο ζητήθηκε να αλλαχθεί διότι υπολογίζει την αναλυτική ολοκλήρωση της κατανομής και όχι απλά αριθμητική. Ωστόσο τα αποτελέσματα ήταν ακριβώς ίδια.

Μετα το πέρας της διαδικασίας αυτής χρειάζεται να κανονικοποιήσουμε τις τιμές του ιστογράμματος προκειμένου να ισχύει $\sum_{i=1}^{n} h(i) = 1$. Η κανονικοποίηση γίνεται διαιρώντας όλες τις τιμές του πίνακα h με το άθροισμα των τιμών του h. Έτσι πάντα ισχύει ότι το άθροισμα των τιμών του πίνακα h θα ισούτε με 1. Έπειτα οι τιμές του ιστογράμματος h δίνονται στην συνάρτηση Y = histtransform(X, h, v) για να μετασχηματιστεί η εικόνα.
Τα αποτελέσματα απο την προσέγγιση των κατανομών που ζητήθηκαν φαίνονται παρακάτω.

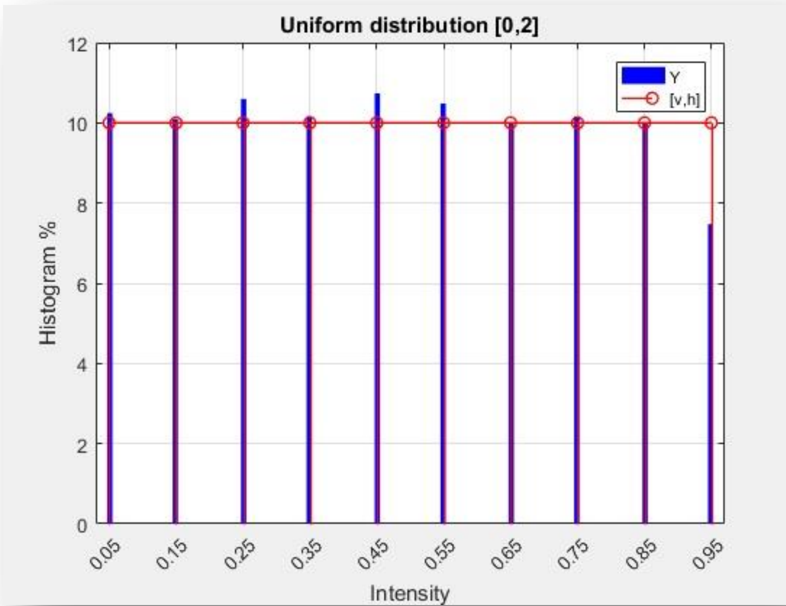
Ομοιόμορφη κατανομή στο [0,1] : `f=@uniformpdf1`

Για την κατανομή αυτή χώρισα το [0,1] σε 10 διαστήματα, ανά 0.1



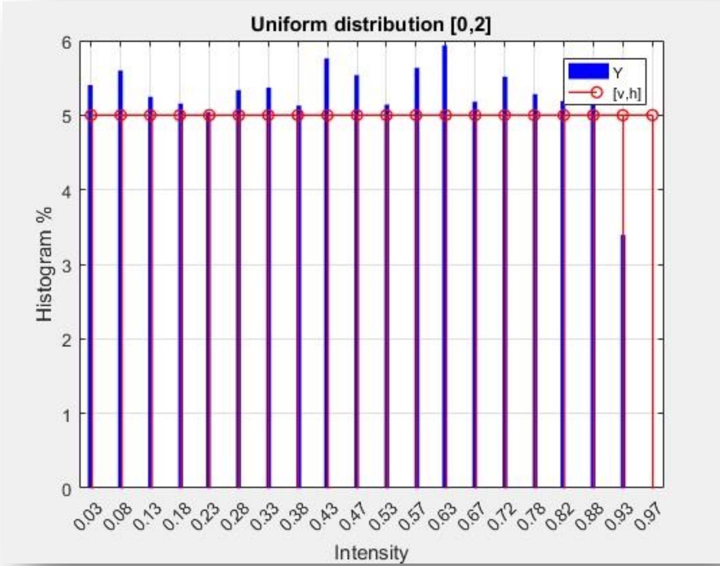
Ομοιόμορφη κατανομή στο [0,2] : `f=@uniformpdf2`

Για την κατανομή αυτή χώρισα το [0,1] σε 10 διαστήματα, ανά 0.1.

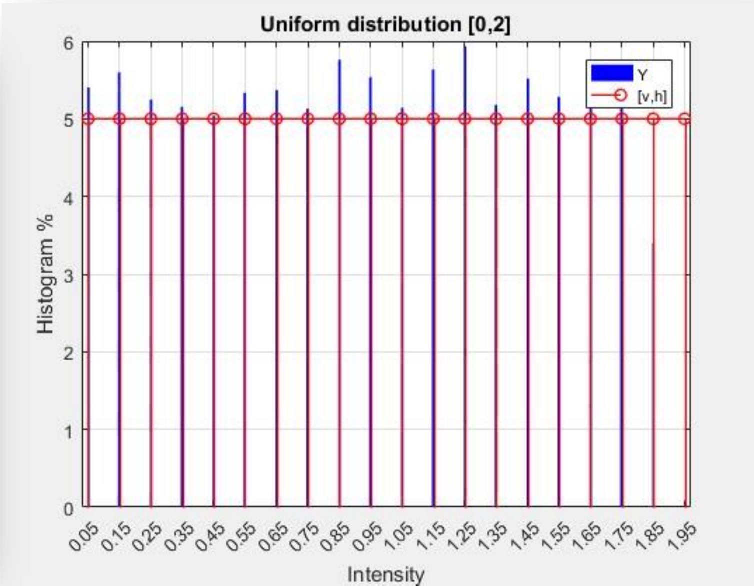


Όπως φαίνεται το ιστόγραμμα είναι ίδιο με την προηγούμενη κατανομή, και συνεπώς και η εικόνα. Παρατηρήθηκε πως για ίδια διαστήματα οι ομοιόμορφη κατανομή στο [0,1] και η ομοιόμορφη στο [0,2] παράγουν ίδια ιστογράμματα. Αυτό συμβαίνει εξαιτίας της κανονικοποίησης του ιστογράμματος.

Τώρα αν χωρίσουμε το [0,1] σε περισσότερα διαστήματα θα έχουμε καλύτερη ποιότητα οπτικά, ωστόσο όπως φαίνεται παρακάτω το ιστόγραμμα αποκλίνει περισσότερο απο την κατανομή.

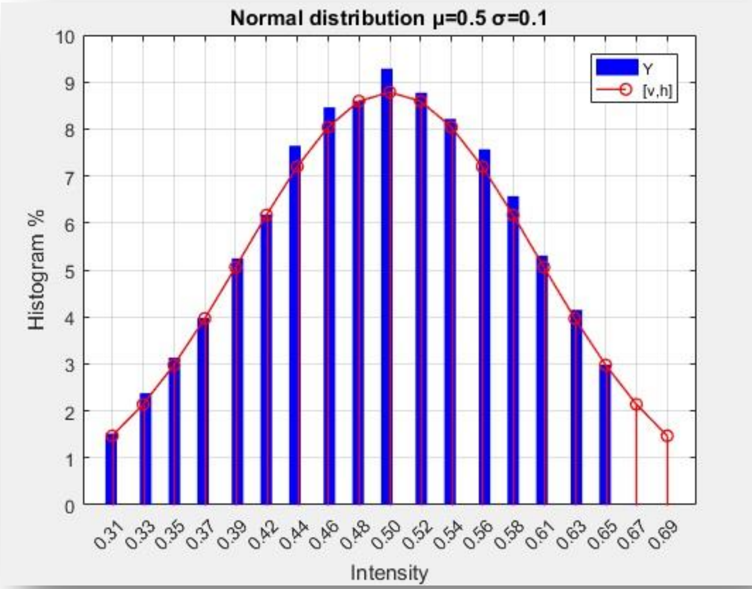


Σε περίπτωση που χωρίζαμε το διάστημα [0,2] αντί του [0,1] η εικόνα θα εμφανίσει κορεσμό στη φωτεινότητα 1, καθώς λόγο grayscale όσες τιμές είναι μεγαλυτερες του 1, μετασχηματίζονται σε 1. Ωστόσο προσεγγίζει καλύτερα την κατανομή καθώς υπάρχουν και οι τιμές ανάμεσα στο [1,2]. Παρακάτω φαίνεται το διάγραμμα και η εικόνα για το διάστημα [0,2] σε 20 υποδιαστήματα.



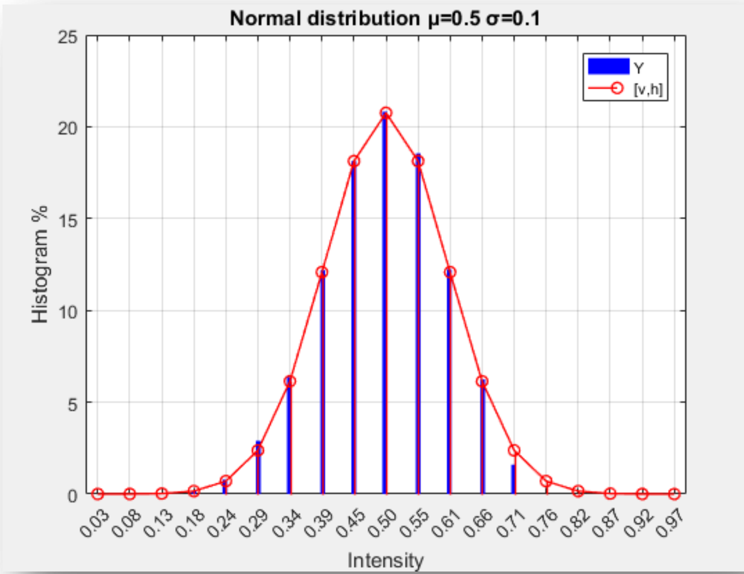
Κανονική κατανομή με μέση τιμή μ=0.5 και τυπική απόκλιση σ=0.1 :

Για την κατανομή αυτή χώρισα το διάστημα [μ-2σ,μ+2σ] σε 20 διαστήματα καθώς για την κανονική κατανομή το 95% των τιμών βρίσκεται στο διάστημα (μ ± 2σ). Δοκίμασα επίσης το διάστημα (μ ± 3σ) ωστόσο αποκλίνει περισσότερο από την δοθέν κατανομή, λόγο του greedy αλγορίθμου.



Η συγκεκριμένη προσέγγιση παρατηρούμε πώς κόβει τις χαμηλές και υψηλές φωτεινότητες.

Τώρα αν χωρίσουμε την κανονική κατανομή σε 20 διαστήματα ανάμεσα στο [0,1] θα έχουμε παρόμοιο αποτέλεσμα.



Πάλι κόβονται οι χαμηλές και υψηλές συχνότητες δραματικά ενώ παράλληλα το οπτικό αποτέλεσμα είναι χειρότερο. Ωστόσο προσεγγίζει καλύτερα την κατανομή που δόθηκε, χωρίς να έχει τιμές που υπερβαίνουν σημαντικά την κατανομή.

Όλα τα παραπάνω διαστήματα επιλέχθηκαν μετά απο δοκιμές ώστε το ιστόγραμμα να προσσεγίζει καλύτερα τις εκάστοτε κατανομές. Παρατηρήθηκε ότι για μεγαλύτερο αριθμό n διαστημάτων, το ιστόγραμμα απέκλινε σημαντικά απο την εκάστοτε κατανομή.