Εργασία 1: Βελτίωση εικόνων μέσω εξισορρόπησης ιστογράμματος

Πλευρίδη Βασιλική Βαρβάρα (ΑΕΜ:10454)

Απρίλιος 2024

1 Εισαγωγή

Στόχος της παρούσας εργασίας είναι η παρουσίαση αλγοριθμικά και η μελέτη βελτίωσης εικόνας μέσω εξισορρόπησης του ιστογράμματος της. Συγκεκριμένα, η παραπάνω διαδικασία έγινε συμβατικά (**Global Histogram Equalization**), αλλά και προσαρμοστικά (**Adaptive Histogram Equalization**), παρατηρώντας διαφορές μεταξύ τους ως προς τις λεπτομέρειες της εικόνας αλλά και ως προς την εξισορρόπηση του ιστογράμματος.

2 Global Histogram Equalization

Το Global Histogram Equalization αποτελεί μια απλή εκδοχή εξισορρόπησης ιστογράμματος και λειτουργεί ως σημειακός μετασχηματισμός λαμβάνονοντας υπόψη το περιεχόμενο ολόκληρης της εικόνας. Το αντίστοιχο αρχείο υλοποίησης του αλγόριθμου σε python είναι το global_hist_eq.py, στο οποίο υλοποιούνται οι συναρτήσεις get_equalization_transform_of_img (συνάρτηση υπολογισμού μετασχηματισμού εξισορρόπησης) και perform_global_hist_equalization (συνάρτηση εφαμοργής του μετασχηματισμού στην εικόνα εισόδου).

2.1 Συνάρτηση get_equalization_transform_of_img

Στην συνάρτηση αυτή, όπως προαναφέρθηκε, υπολογίζεται ο μετασχηματισμός εξισορρόπησης εικόνας, για όλα τα επίπεδα φωτεινότητας, δηλαδή για στάθμες εισόδου L= 0 - 256. Οι εξισώσεις υπολογισμού του μετασχηματισμού, παρέχονται από την θεωρία και είναι οι εξής:

$$v_k = \sum_{i=0}^{k} p(x_i) \tag{1}$$

$$y_k = \text{round}\left(\frac{v_k - v_0}{1 - v_0} \cdot (L-1)\right) \tag{2}$$

Όπου το v_k είναι η αθροιστική συνάρτηση πιθανοτήτων $p(x_i)$,για i που παίρνουν τιμές από 0 έως k, ενώ y_k είναι ο τελικός μετασχηματισμός.

Σαν αλγόριθμος έχει αρκετά απλή λογική. Αρχικά υπολογίζεται το ιστόγραμμα της εικόνας είσοδου μέσω της βοηθητικής συνάρτησης custom_hist. Η συνάρτηση αυτή παίρνει ως όρισμα τον πίνακα arr του οποίου αναζητείται το ιστόγραμμα και έναν αριθμό minlength , ο οποίος αναφέρεται

στο πλήθος διαφορετικών επιπέδων φωτεινότητας. Η μεταβλητή result είναι ένας NumPy πίνακας με μήκος minlength. Κατά την εκτέλεση του βρόχου, οι μοναδικές τιμές και οι αντίστοιχοι μετρητές αποθηκεύονται στις μεταβλητές unique_values και counts αντίστοιχα. Στη συνέχεια, ο πίνακας result γεμίζει με τον αριθμό των εμφανίσεων της κάθε μοναδικής τιμής στον πίνακα arr. Επιστρέφοντας πίσω στην κύρια συνάρτηση, δημιουργείται ένα for loop το οποίο υπολογίζει τον τελικό μετασχηματισμό μέσω των εξισώσεν (1) και (2), καθώς και με την βοήθεια της εξόδου της συνάρτησης custom_hist.

2.2 Συνάρτηση perform_global_hist_equalization

Η συνάρτηση perform_global_hist_equalization, μέσα από ένα διπλό for loop, μετασχηματίζει την τιμή του κάθε pixel της εικόνας εισόδου καλώντας την συνάρτηση get_equalization_transform_of_img και εν τέλει επιστρέφει τον τελικό μετασχηματισμένο πίνακα της εικόνας.

3 Adaptive Histogram Equalization

Το Adaptive Histogram Equalization πραγματοποιεί την εξισορρόπηση της τιμής ενός σημείου, με βάση τις κατανομές των περιοχών που περιβάλλουν αυτό. Με άλλα λόγια, χωρίζει την εικόνα σε περιοχές διαστάσεων region_len_h,region_len_w και βρίσκοντας τα 4 κοντινότερα κέντρα της κάθε περιοχής στο εκάστοτε pixel, επιλέγονται οι περιοχές που χρησιμοποιούνται στην διγραμμική παρεμβολή για την εύρεση του τελικού μετασχηματισμού. Το αντίστοιχο αρχείο υλοποίησης του αλγόριθμου σε python είναι το adaptive_hist_eq.py, στο οποίο υλοποιούνται οι συναρτήσεις calculate_eq_transformations_of_regions (ορισμός των μη επικαλυπτόμενων contextual regions και εύρεση του μετασχηματισμένου πίνακα του κάθε region) και perform_adaptive_hist_equalization (εύρεση τελικού μετασχηματισμού των pixel, ανάλογα με την θέση τους στην εικόνα). Για την τελευταία συνάρτηση, χρησιμοποιήθηκε μια βοηθητική συνάρτηση interference_transform, η οποία υλοποιεί ξεχωριστά την εύρεση του μετασχηματισμού για συγκεκριμένου pixel εισόδου και θα αναλυθεί και αυτή παρακάτω.

3.1 Συνάρτηση calculate_eq_transformations_of_regions

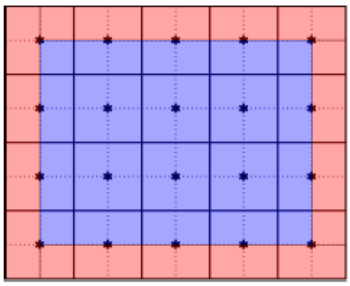
Στην συνάρτηση αυτή, υλοποιείται ένα διπλό for loop, έτσι ώστε να εντοπιστούν όλα τα regions ως προς και τις δύο διαστάσεις. Ως κλειδί (σε μορφή tuple) του κάθε region,ορίζεται το ζεύγος των δεικτών της κορυφής της κοντινότερης στην αρχή των αξόνων, όπως αυτο περιγράφεται και στην εκφώνηση της εργασίας. Μετά τον ορισμό της περιοχής του κάθε region, καλείται η συνάρτηση get_equalization_transform_of_img, έτσι ωστε να βρεθεί ο τελικός μετασχηματισμός του region και στην συνέχεια να τοποθετηθεί μαζί με τα υπόλοιπα σε ένα dictionary με κλειδιά τα tuples που ορίστηκαν παραπάνω.

3.2 Συνάρτηση perform_adaptive_hist_equalization

Η συνάρτηση αυτή καλεί την calculate_eq_transformations_of_regions για την ευρεση των μετασχηματισμών όλων των regions και στην συνέχεια, με παρόμοια λογική με αυτή της συνάρτησης perform_global_hist_equalization, χρησιμοποιεί ένα διπλό for loop, μετασχηματίζοντας έτσι όλα τα pixel, χρησιμοποιόντας την συνάρτηση interference_transform και στην συνέχεια τοποθετόντας τα στον τελικό πίνακα εξόδου.

3.3 Συνάρτηση interference_transform

Ο αλγόριθμος ακολουθεί διαφορετική διαδικασία εύρεσης μετασχηματσιμού, ανάλογα με το αν οι συντεταγμένες του pixel, βρίσκονται στην κόκκινη (outer points)ή στην μπλε (inner points) όπως αυτή φαίνεται στο Σχήμα 1.



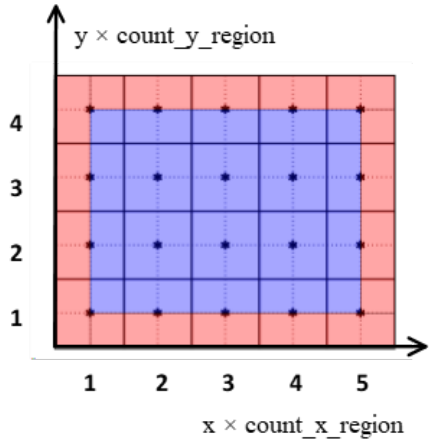
Σχήμα 1: Διαφορετικές ζώνες των σημείων μιας εικόνας

3.3.1 Outer Points

Για την περίπτωση ενός outer point, πρέπει να βρεθεί το region στο οποίο ανήκει έτσι ώστε να μετασχηματιστεί στην συνέχεια κατάλληλα με την βοήθεια του κατάλληλου πίνακα από το regions_transform dictionary. Για την εύρεση του region λοιπόν, χρησιμοποιήθηκαν οι μεταβλητές count_x_region, count_y_region,οι οποίες ουσιαστικά βρίσκουν τις συντεταγμένες του ζητούμενου region σε κλίμακα region_len_h,region_len_w.Στο Σχήμα 2, παρουσιάζεται αυτή η λογική. Τέλος, επιλέγεται ο κατάλληλος μετασχηματισμός και υπολογίζεται η τιμή εξόδου. Σχετικά με την εύρεση του μετασχηματισμού, η αντίστοιχη γραμμή κώδικα είναι:

```
t=regions_transform[(count_x_region-1)*region_len_h,/
(count_y_region-1)*region_len_w]
```

Τα count_x_region, count_y_region είναι ουσιαστικά η πάνω δεξιά γωνία του region. Για αυτό τον λόγο, γίνεται η ανάλογη αφαίρεση (-1 σε καθε συντεταγμένη) και έπειτα η αναγωγή σε κανονική κλίμακα (*region_len_h), για να προκύψει έτσι το σωστό κλειδί.



Σχήμα 2: Regions σε κλίμακα counts

3.3.2 Inner Points

Για την περίπτωση ενός inner point,γίνεται χρήση του αλγορίθμου της διγραμμικής παρεμβολής ο οποίος χρησιμοποιεί τους παρακάτω τύπους:

$$y = (1-a)(1-b)T_{-,-}(x) + (1-a)bT_{+,-}(x) + a(1-b)T_{-,+}(x) + abT_{+,+}(x) \tag{3}$$

όπου

$$a = \frac{w_P - w_-}{w_+ - w_-}$$
$$b = \frac{h_P - h_-}{h_+ - h_-}$$

Τα $T_{-,-}, T_{+,-}, T_{-,+}, T_{+,+}$, αποτελούν τους μετασχηματισμούς των γειτονικών regions. Για την εύρεση αυτών, αρκεί να βρεθούν τα κέντρα τους τα οποία προσδιορίζουν το ορθογώνιο εντος του οποίου βρίσκεται το εκάστοτε σημείο ενδιαφέροντος. Ακολουθώντας λοιπόν την λογική της παραπάνω συνάρτησης, ορίζονται οι μεταβλητές count_x_centers, count_y_centers, οι οποίες βρίσκουν τις συντεταγμένες του ζητούμενου κέντρου σε κλίμακα region_len_h,region_len_w, ξεκινώντας όμως την αρχή των αξόνων από το σημείο$(region_len_h/2, region_len_w/2)$. Έτσι εξηγείται η προσθαφαίρεση της των τιμών $region\_len\_h/2, region\_len\_w/2$ στις ανάλογες γραμμές του κώδικα. Να σημειωθεί ότι το κέντρο που υπολογίζεται είναι αυτο που αντιστοιχεί στον μετασχηματισμό $T_{+,+}$ και στην συνέχεια, γνωρίζοντας τις διαστάσεις του ορθογωνίου που σχηματίζεται, βρίσκονται και τα υπόλοιπα.

3.4 Συνάρτηση no_interference_transform

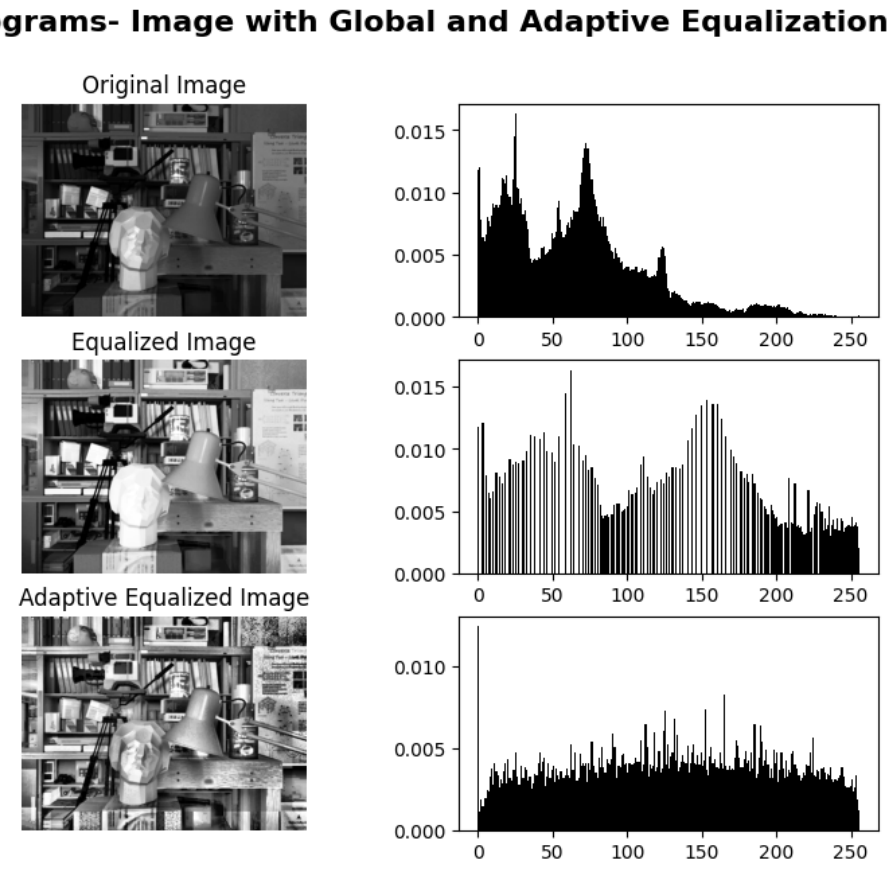
Η συνάρτηση αυτή αποτελεί μια απλοποίηση αυτής της perform_adaptive_hist_equalization σε συνδιασμό με την interference_transform, καθώς ακολουθεί την διαδικασία υπολογισμού του outer point για τον μετασχηματισμό του κάθε πιξελ. Στόχος της είναι να παρατηρήσει τις ασυνέχειες που θα υπάρξουν στα όρια μεταξύ των regions, χωρίς την χρήση της παρεμβολής.

4 Αρχείο demo.py

Σε αυτό το αρχείο επιδεικνύεται η λειτουργία των ζητούμενων συναρτήσεων, χρησιμοποιώντας την εικόνα input_img.pngσαν είσοδο. Αφού γίνει η μετατροπή της εικόνας σε grayscaleόπως αυτή περιγράφεται στην εκφώνηση της εργασίας, δημιουργείται ένα subplot, το οποίο τυπώνει την αρχική, την συμβατικά εξισορροπημένη και την προσαρμοστικά εξισορροπημένη εικόνα καθώς και τα αντίστοιχα ιστογράμματα τους. Επίσης, γίνεται παρούσιαση της σύγκρισης της εφαμοργής AHE με και χωρίς χρήση της παρεμβολής γειτονικών περιοχών, σε ένα subplot.

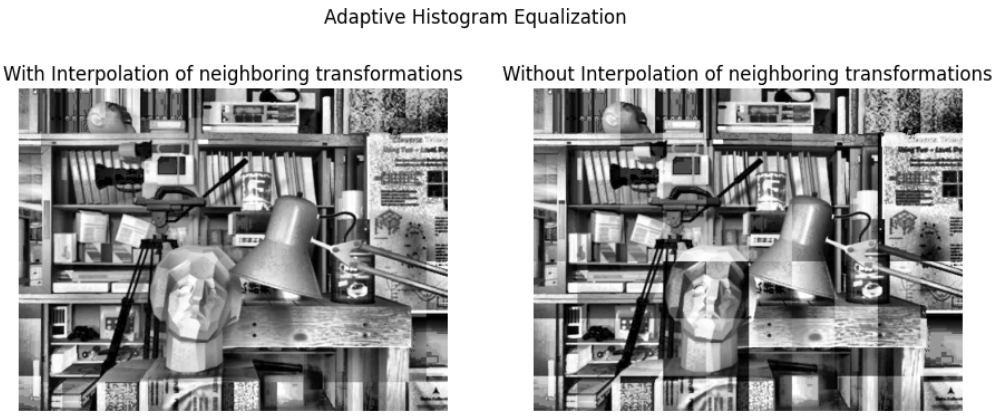
5 Σχολιασμός και αποτελέσματα

Τα ζητούμενες εικόνες και τα ιστογράμματά τους παρουσιάζονται παρακάτω:



Όπως φαίνεται από τις εικόνες,οι λεπτομέρειες της εικόνας της εικόνας που προκύπτει από την προσαρμοστική εξισορρόπηση ιστογράμματος, αναδεικνύονται αποτελεσματικότερα. Στην περίπτωση του Global Histogram Equalization, φαίνεται να μην γίνεται καλή προσαρμογή της αντίθεσης, ειδικά σε σημεία μεγαλύτερης φωτεινότητας που τείνουν προς το άσπρο. Σχετικά με τα ιστογράμματα, όπως ήταν αναμενόμενο, την καλύτερη εξισορρόπηση την έχει το ιστόγραμμα της προσαρμοστικά εξισορροπημένης εικόνας. Παρόλα αυτά, και στις 2 εξισορροπήσεις το εύρος δυναμικού εικόνας είναι το βέλτιστο, καλύπτοντας όλα τα επίπεδα φωτεινότητας στο δυνατό φάσμα.

Σε περίπτωση που στην εφαρμογή AHE δεν χρησιμοποιηθεί η παρεβολή γειτονικών μετασχηματισμών, αλλά χρησιμοποιειθεί για την κάθε contextual region, ο δικός της μετασχηματισμός εξισορρόπησης και μόνο, περιμένουμε να παρατηρηθεί μια ασυνέχεια στα σύνορα μεταξύ των περιοχών. Για να επαληθευτεί η παραπάνω υπόθεση, καλείται η συνάρτηση no_interference_transform και για την καλύτερη σύγκριση και παρατήρηση των εικονών, παρουσιάζονται μαζί σε ένα subplot.



Η ασυνέχεια μεταξύ των regions, είναι εύκολα ορατή και έτσι η χρησιμότητα του αλγορίθμου της διγραμμικής παρεμβολής είναι εμφανής.

Να σχολιαστεί επίσης ότι για τον έλεγχο της καλής λειτουργίας των συναρτήσεων, χρησιμοποιήθηκαν σαν είσοδο κι άλλες εικόνες. Ένα παράδειγμα (το οποίο δεν ενσωματώνεται στο demo.py) φαίνεται παρακάτω:

Τέλος, παρατηρήθηκε ότι έχοντας σαν είσοδο μια μονόχρωμη εικόνα και συγκεκριμένα χρώματος μαύρου, ο αλγόριθμος δεν τρέχει. Αυτό συμβαίνει γιατί το v_0 είναι ίσο με 0 και έτσι, η διαίρεση στην σχέση (2) γίνεται με το 0.Αυτό δεν είναι έγκυρο μαθηματικά και μπορεί να οδηγήσει σε σφάλματα στον αλγόριθμο. Ίσως θα έπρεπε να προστεθεί ένας έλεγχος στην περίπτωση αυτή,ο οποίος αγνοείται στην παρούσα υλοποίηση καθώς ο αλγόριθμος προσανατολίζεται σε εικόνες με διαφορετές στάθμες φωτεινότητας.