Ψηφιακή Επεξεργασία Εικόνας
-Εργασία 3-

—

Οπτική αναγνώριση χαρακτήρων

—

Προεπεξεργασία εικόνας για ανάγνωση χαρακτήρων
Εντοπισμός περιγράμματος
Αναπαράσταση και σύγκριση περιγραμμάτων
Αναγνώριση χαρακτήρων

Μάστορας Ραφαήλ Ευάγγελος

7918

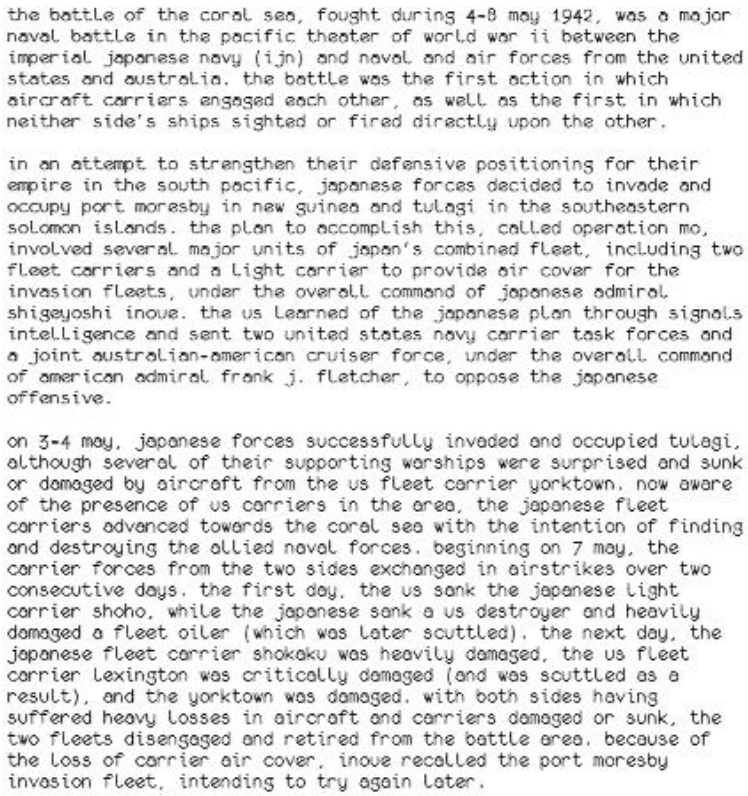
11/6/2017

Περιγραφή της εργασίας

Στην εργασία αυτή μας δόθηκε μια εικόνα κειμένου, την οποία έπρεπε να επεξεργαστούμε κατάλληλα προκειμένου να αναγνωρίσουμε μέσω Matlab το κείμενο που περιέχει. Αρχικά χρειάστηκε να την περιστρέψουμε κατάλληλα. Έπειτα έπρεπε να δημιουργήσουμε περιγραφές για τα γράμματα που μας δόθηκαν , βρίσκοντας τα περιγράμματα τους. Τέλος έπρεπε να χωρίσουμε την εικόνα στα επιμέρους γράμματα που απαρτίζουν το κείμενο και να τα αναγνωρίσουμε.

Προεπεξεργασία εικόνας

Καθώς η εικόνα που μας δόθηκε είχε υποστεί περιστροφή, χρειάστηκε να την αναιρέσουμε. Για να το πετύχω αυτό δημιούργησα την συνάρτηση fixrotation(x). Στην συνάρτηση αυτή χρησιμοποιώ αρχικά ένα Gaussian φίλτρο για να δημιουργήσω θόρυβο και έπειτα ένα ζωνοπερατό φίλτρο για να ενωθούν τα γράμματα μεταξύ τους. Μετά βρίσκω την τον μέγιστο συντελεστή Fourier της εικόνας , αφού πρώτα έχω αφαιρέσει την DC συνιστώσα. Γνωρίζοντας την θέση του μέγιστου συντελεστή Fourier βρίσκω την γωνία theta και την μετατρέπω σε rad. Έπειτα χρησιμοποιώ την γωνία αυτή για να γίνει μια πρώτη περιστροφή της εικόνας. Προκειμένου να βελτιωθεί το αποτέλεσμα ακόμα περισσότερο, αναζητώ σειριακά την βέλτιστη περιστροφή γύρω από την αρχική μου γωνία. Αυτό το κάνω χρησιμοποιώντας την κατακόρυφη προβολή της φωτεινότητας. Συγκεκριμένα ελέγχω σε ποια γωνία έχω περισσότερες λευκές γραμμές και αυτή θα είναι η βέλτιστη.



Αρχικά η γωνία βρέθηκε 12.265 και βελτιώθηκε σε 12.015.

Διαχωρισμός γραμμάτων κειμένου

Στην συνέχεια δημιούργησα την συνάρτηση lines=getlines(y) η οποία λαμβάνει την εικόνα και επιστρέφει έναν cell array που περιέχει τις επιμέρους γραμμές. Για να το πετύχω αυτό χρησιμοποίησα την κατακόρυφη προβολή της φωτεινότητας. Κάθε γραμμή κειμένου προσδιορίζεται βρίσκοντας σε ποια γραμμή της εικόνας η φωτεινότητα γίνεται μεγαλύτερη του μηδενός και έπειτα από μερικές γραμμές ξανά μηδενίζεται. Το διάστημα ανάμεσα στον επιμέρους μηδενισμό της φωτεινότητας προσδιορίζει και την ύπαρξη γραμμής κειμένου.

Έπειτα η συνάρτηση letters=getletters(lines) λαμβάνει τις επιμέρους εικόνες των γραμμών του κειμένου και χωρίζει τα γράμματα που βρίσκονται μέσα σε αυτές. Η διαδικασία είναι ίδια με πριν με την διαφορά ότι τώρα χρησιμοποιώ την οριζόντια προβολή φωτεινότητας για να διακρίνω που αρχίζει και που τελειώνει το κάθε γράμμα. Επίσης χρειάστηκε να χρησιμοποιήσω ζωνοπερατό φίλτρο και να διώξω τυχόν ‘σκουπίδια’ με τον μορφολογικό τελεστή imopen.

Οι παραπάνω συναρτήσεις καλούνται μέσω της συνάρτηση readtext(y) που υλοποιεί την συνολική διαδικασία αναγνώρισης των γραμμάτων ανά γραμμή.

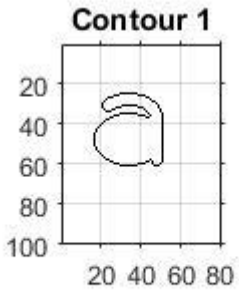
Εντοπισμός περιγραμμάτων

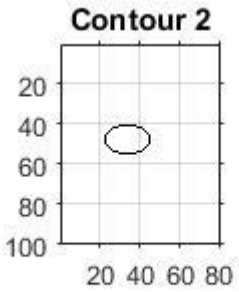
Για τον εντοπισμό των περιγραμμάτων δημιούργησα την συνάρτηση getcontour(x) η οποία λαμβάνει μια εικόνα και εντοπίζει το περίγραμμα του γράμματος που περιέχεται σε αυτή. Ο πίνακας που επιστρέφει είναι οι συντεταγμένες των σημείων που δημιουργούν το περίγραμμα του γράμματος. Αν το γράμμα έχει δύο περιγράμματα τα διαχωρίζει σε διαφορετικό cell. Προκειμένου να βρεθεί το εκάστοτε περίγραμμα κατωφλιώνω αρχικά την αρνητική εικόνα, έπειτα χρησιμοποιώ τον μορφολογικό τελεστή imdilate για να της διαστείλω και αφαιρώντας την διασταλμένη εικόνα από την αρχική μένει ένα περίγραμμα. Το περίγραμμα αυτό το λεπταίνω με τον μορφολογικό τελεστή bwmorph(xtemp,'thin','inf') και στην συνέχεια αναζητώ σειριακά το περίγραμμα. Μόλις βρεθεί κάποιο σημείο με τιμή φωτεινότητας 1 αναζητώ ωρολογιακά τα σημεία γύρω από αυτό και η διαδικασία συνεχίζεται κάθε φορά στο επόμενο γειτονικό σημείο που εντοπίστηκε πρώτο. Τα σημεία αυτά αποθηκεύονται σε έναν πίνακα και παράλληλα διαγράφονται από την αρχική εικόνα. Έτσι αν υπάρχει δεύτερο περίγραμμα αυτό θα βρίσκεται στον αρχικό πίνακα πλέον μόνο του, καθώς το πρώτο περίγραμμα έχει διαγραφεί από αυτή. Η προηγούμενη διαδικασία επαναλαμβάνεται εφόσον υπάρχει δεύτερο περίγραμμα.

Τα περιγράμματα που ζητήθηκαν εμφανίζονται παρακάτω.

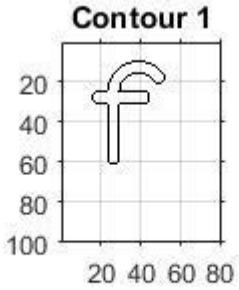
Περιγράμματα:

‘a’

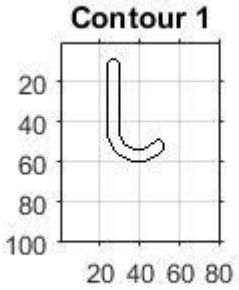




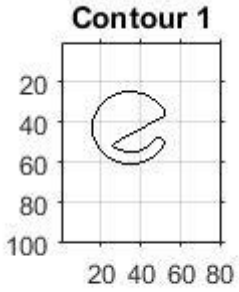
‘f’

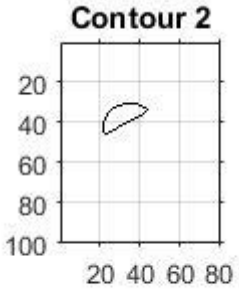


‘l’



‘e’





Αναγνώριση κειμένου

Η συνάρτηση readtext(y) όπως προανέφερα καλεί τις συναρτήσεις που περιεγράφηκαν παραπάνω. Επίσης δημιουργεί τους περιγραφές των δοθέν γραμμάτων αλλά και των γραμμάτων που βρέθηκαν στην εικόνα. Για την περιγραφή των γραμμάτων χρησιμοποιήθηκε ο τύπος $r[i] = x[i] + jy[i]$. Αφού βρεθούν αυτές οι περιγραφές, για κάθε γράμμα της εικόνας

αναζητείται μέσω της συνάρτησης getChar(r1Ref,r2Ref,r1,r2) ο πιο κοντινός περιγραφέας αναφοράς. Καθώς όμως οι περιγραφές των περιγραμμάτων δεν έχουν ίδιο μέγεθος χρειάστηκε αρχικά να κάνω παρεμβολή με την συνάρτηση intep1. Έπειτα βρίσκοντας το τετραγωνικό σφάλμα ανάμεσα στην μετασχηματισμό Fourier του περιγραφέα αναφοράς και του περιγραφέα του γράμματος της εικόνας και παίρνοντας το μικρότερο τετραγωνικό σφάλμα, βρίσκω το γράμμα που αντιστοιχεί στον περιγραφέα αναφοράς μέσω της συνάρτησης char(64+index) , όπου index το νούμερο του βέλτιστου περιγραφέα.

Στο σημείο αυτό να αναφέρω ότι δυστυχώς δεν κατάφερα να αναγνωρίσω σωστά το κείμενο και για αυτό τον λόγο δεν παρατίθεται το αποτέλεσμα. Ωστόσο τα περιγράμματα των γραμμάτων αναφοράς εμφανίζονται σωστά (εκτός των γραμμάτων k,p,w) και ο διαχωρισμός των γραμμάτων της εικόνας είναι επιτυχής.