

Υπολογιστική Όραση (ΜΥΕ046)

2η σειρά ασκήσεων

Ακαδημαϊκό έτος: 2019-2020
Διδάσκων: Γιώργος Σφήκας

Ημερομηνία παράδοσης: 14 Μαΐου 2020

- Όλες οι προγραμματιστικές ασκήσεις πρέπει να παραδοθούν σε γλώσσα Python 3.
- Οι ασκήσεις είναι ατομικές - δεν επιτρέπεται η μεταξύ σας συνεργασία για την λύση τους.
- Δεν επιτρέπεται να χρησιμοποιήσετε κώδικα που τυχόν θα βρείτε στο web (εκτός από όπου σημειώνεται στην εκφώνηση ότι επιτρέπεται). Η χρήση κώδικα τρίτων θα έχει σαν αποτέλεσμα τον αυτόματο μηδενισμό σας.
- Ο κώδικάς σας πρέπει να σχολιαστεί εκτενώς.
- Κάθε άσκηση να αντιστοιχεί σε διαφορετικό folder στο github repository που σας αντιστοιχεί.
- Στο github να ανεβάσετε και αρχείο με όνομα README.md όπου να γράφετε το ονοματεπώνυμο σας, αριθμό μητρώου, e-mail, και έτος φοίτησης.
- Μην αφήσετε τις ασκήσεις για την τελευταία στιγμή! Αρχίστε να δουλεύετε σε αυτές το συντομότερο δυνατόν.

Άσκηση 1

Σε αυτή την άσκηση πρέπει να επεξεργαστείτε το ίδιο χειρόγραφο που είχατε δει και στην πρώτη σειρά ασκήσεων. Κατεβάστε το πλήρες χειρόγραφο από το URL http://cs.uoi.gr/~sfikas/teaching/assignment1_3/trikoupi6_low.png αν δεν το έχετε ήδη.

Στόχος είναι να υλοποιήσετε έναν αλγόριθμο προσαρμοστικής κατωφλίωσης, επεκτείνοντας την βασική μέθοδο Otsu για αυτό τον σκοπό. Αυτό σημαίνει ότι δεν θα υπάρχει ένα κατώφλι για όλη την εικόνα, αλλά ένα διαφορετικό κατώφλι για κάθε ξεχωριστό εικονοστοιχείο. Για κάθε εικονοστοιχείο να λάβετε υπ'όψιν τα εικονοστοιχεία γύρω από μία γειτονιά (δηλαδή, σαν να ήταν αυτά όλη η εικόνα κάθε φορά), και χρησιμοποιώντας αυτά να υπολογίσετε κατώφλι μεγιστοποιώντας την αντικειμενική συνάρτηση του Otsu. Παράμετρος που θα δίνει ο χρήστης είναι το μέγεθος της γειτονιάς, $window_size \times window_size$ σε pixels.

Να προσέξετε τι εικονοστοιχεία θα λάβετε υπ'όψιν στην γειτονιά όταν υπολογίζετε κατώφλι για εικονοστοιχεία κοντά στα όρια της εικόνας. Όπου το παράθυρο της γειτονιάς 'βγαίνει' από τα όρια της εικόνας, μην κάνετε κάποια υπόθεση για τις τιμές εκτός εικόνας (δηλαδή, μην θεωρήσετε zero-padding).

Θα πρέπει να παραδώσετε:

- Κώδικα python που να εκτελείται με την εντολή: `python3 adaptive.py <input filename> <output filename> <window size>`
- Jupyter notebook που να περιλαμβάνει παράδειγμα χρήσης του κώδικα στο χειρόγραφο που δίνεται, για ένα αριθμό από διαφορετικά μεγέθη παράθυρου.

Ο κώδικάς σας θα πρέπει να δουλεύει είτε η εικόνα που δίνει ο χρήστης είναι grayscale (όπως το χειρόγραφο που δίνεται) είτε η εικόνα είναι έγχρωμη. Αν η εικόνα εισόδου είναι έγχρωμη, να μετατρέπεται σε grayscale στον κώδικα σας παίρνοντας απλά τον μέσο όρο των καναλιών *Red*, *Green*, *Blue*.

Άσκηση 2

Να υπολογίσετε τις παραγώγους των παρακάτω εκφράσεων ως προς x , όπου $D \in \mathbb{R}^{n \times n}$, $f, x \in \mathbb{R}^n$. Στην θέση των $\alpha, \beta, \gamma, \delta$ αντικαταστήστε τα ψηφία του αριθμού μητρώου σας (π.χ. αν $AM=4231$, τότε $\alpha = 4, \beta = 2, \gamma = 3, \delta = 1$). Μπορείτε να χρησιμοποιήσετε τα αποτελέσματα που συζητήσαμε στο μάθημα.

α) $\beta f^T x + 3$

β) $\gamma \|x\|^2 + 2f^T x + x^T(D + \beta I)f + \delta f^T D^T f$

Για την παρακάτω συνάρτηση βρείτε για ποια τιμή του διανύσματος x η παράγωγος μηδενίζεται.

γ) $g(x) = \frac{1}{2}x^T(D^T D + \alpha I)x + f^T x + 10$

Άσκηση 3



Εικ 1: Η εικόνα που θα επεξεργαστείτε στην άσκηση 3.

Το ζητούμενο είναι να εφαρμοστεί κατάλληλος γεωμετρικός μετασχηματισμός στην εικόνα 1 (κατεβάστε την από http://cs.uoi.gr/~sfikas/teaching/assignment2_3/granma.jpg), ώστε να σχηματιστεί εικόνα μεγέθους 1000×1000 τέτοια που να περιέχει μόνο την εφημερίδα, και αυτή με τις πλευρές της να συμπίπτουν στα όρια της εικόνας. Παράδειγμα αποτελέσματος μπορείτε να δείτε στην εικόνα 2. Πρώτα ζητήστε από τον χρήστη τα σημεία που αντιστοιχούν στις 4 γωνίες του αντικειμένου. Αυτό να το κάνετε εμφανίζοντας την εικόνα, και ζητώντας από τον χρήστη να κλικάρει στις γωνίες της εφημερίδας (εδώ μπορείτε να χρησιμοποιήσετε τις συναρτήσεις `setMouseCallback` και `waitKey` της OpenCV ή οποιαδήποτε άλλη συνάρτηση).

Για να εφαρμόσετε τον μετασχηματισμό, επίσης μπορείτε να χρησιμοποιήσετε την “έτοιμη” συνάρτηση `warpPerspective` της OpenCV. Εκεί που οπωσδήποτε δεν πρέπει να χρησιμοποιήσετε κάτι έτοιμο εκτός από βασικές πράξεις και συναρτήσεις γραμμικής άλγεβρας, είναι στον υπολογισμό του πίνακα 3×3 που αντιστοιχεί στον προβολικό μετασχηματισμό (ή “perspective transform” ή “ομογραφία”) που πρέπει να εφαρμοστεί ώστε να πάρουμε το ζητούμενο αποτέλεσμα. Αυτός ο πίνακας έχει 9 στοιχεία, από τα οποία τα 8 αντιστοιχούν σε μεταβλητές παραμέτρους. Ακολουθήστε τα όσα συζητήσαμε στο μάθημα για να υπολογίσετε τις κατάλληλες τιμές.

Θα πρέπει να παραδώσετε:

- Κώδικα python που να εκτελείται με την εντολή: `python3 warp.py <input filename> <output filename>`
- Jupyter notebook που να περιλαμβάνει παράδειγμα χρήσης του κώδικα.



Εικ 2: Παράδειγμα ζητούμενου στην άσκηση 3.