***Τσόχλας Ιωάννης***

AM 4993 | cs04993@uoi.gr

**ΜΥΕ037**

**Ψηφιακή Επεξεργασία Εικόνας**

**ΕAΡΙΝΟ ΕΞΑΜΗΝΟ 2023-2024**

***ΕΡΓΑΣΙΑ***

**Project Goal :**

Στόχος της εργασίας είναι η εκμάθηση και εφαρμογή τεχνικών ψηφιακής επεξεργασίας εικόνας χρησιμοποιώντας τη γλώσσα προγραμματισμού Python και βιβλιοθήκες όπως Numpy, SciPy και Matplotlib

 Κατασκευή τμημάτων εικόνας (image patches) και κανονικοποίησή τους.

 Υλοποίηση συνέλιξης (convolution) και χρήση Γκαουσιανού φίλτρου.

 Ανίχνευση ακμών μέσω τελεστών όπως Sobel και LoG.

**Github Reposiotry :**  <https://github.com/GiannisTsochlas/PsifiakiEpejergasiaEikonas>

*Contents*

[Τοπικές περιοχές/Τμήματα εικόνας 2](#_Toc169885741)

[ΤΦιλτράρισμα στο πεδίο του χώρου 3](#_Toc169885741)

[Αξιοσημείωτες Πήγες Μελέτης 10](#_Toc169885742)

**Μέρος 1** **Τοπικές περιοχές/Τμήματα εικόνας (Patches)**

Αξιοσημείωτο θεωρώ να επισημάνω τις αλλαγές πλαισίων που έκανα για να τρέξει ο κώδικας μιας και μετά την υλοποίηση της **image\_patches**, uπήρχαν errors.

* Debug για τη διαχείριση των errors (τα παρακάτω αντικείμενα είναι κενά γιατί δεν τα έχουμε υλοποιήσει ακόμα για αυτό το λόγο βάζουμε μια συνθήκη τύπου try catch για να καταφέρουμε να έχουμε κάποιο αποτέλεσμα στην κονσόλα:

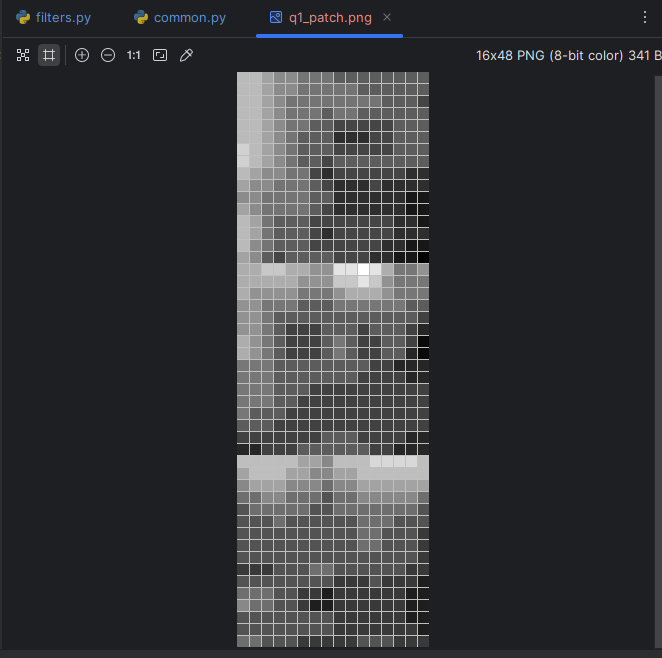
Εικόνα που περιέχει κείμενο, λογισμικό, λογισμικό πολυμέσων, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

* Η κονσόλα μας επισήμανε να αλλάξουμε τη μεταβλητή n.float σε np.float64

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, λογισμικό πολυμέσων

Περιγραφή που δημιουργήθηκε αυτόματα



Τα 3 τμήματα τις εικόνας από το αρχείο grace\_hopper.png

1. Η κανονικοποίηση των τμημάτων της εικόνας ώστε να έχουν μηδενική μέση τιμή έχει αρκετά πλεονεκτήματα . Bοηθά στη βελτίωση της ποιότητας της εικόνας και της απόδοσης των αλγορίθμων επεξεργασίας εικόνας, μειώνοντας παράλληλα την επίδραση του θορύβου.

Εικόνα που περιέχει κείμενο, γραμματοσειρά, στιγμιότυπο οθόνης, λευκό

Περιγραφή που δημιουργήθηκε αυτόματα

**Υπόδειξη:**

Το δεύτερο σενάριο, όπου η τιμή της σκοτεινής αντιστοιχεί σε -1 και η τιμή της φωτεινής σε 1, είναι πιο κατάλληλο για να μετρήσουμε την ομοιότητα χρησιμοποιώντας εσωτερικά γινόμενα. Αυτό οφείλεται στο γεγονός ότι η συμμετρική αυτή κλίμακα παρέχει μια πιο ισορροπημένη και λεπτομερή μέτρηση των διαφορών φωτεινότητας

1. 1. Όταν το αντικείμενο μετακινείται μέσα στην εικόνα, η μηδενική μέση τιμή διασφαλίζει ότι η συνολική φωτεινότητα της εικόνας δεν επηρεάζεται από τη θέση του αντικειμένου, γιατι οι θετικές και αρνητικές τιμές θα αλληλοακυρώνονται.

2.Η αλλαγή κλίμακας μπορεί να επηρεάσει την ένταση των τιμών των εικονοστοιχείων.

3.Η σταθερή μέση τιμή βοηθά στη διατήρηση της σύγκρισης ανεξάρτητα από την ένταση του φωτός.

# Συμπέρασμα

Η χρήση τμημάτων με μηδενική μέση τιμή είναι ιδιαίτερα χρήσιμη για προβλήματα όπως η αντιστοίχιση και η αναγνώριση αντικειμένων.

**Μέρος 2****Φιλτράρισμα στο πεδίο του χώρου**

**Ζήτημα 2: Συνέλιξη και Γκαουσιανό φίλτρο (Convolution and Gaussian Filter)**

1. Επιλέγουμε ένα συγκεκριμένο μέγεθος φίλτρου k. Για το παράδειγμα αυτό, ας θεωρήσουμε ότι το k=3

Εικόνα που περιέχει γραμματοσειρά, γραφικός χαρακτήρας, κείμενο, λευκό

Περιγραφή που δημιουργήθηκε αυτόματα

Ορίζουμε ένα 2Δ φίλτρο Gaussian G μεγέθους 3×3 .

Εικόνα που περιέχει γραμματοσειρά, κείμενο, γραφικός χαρακτήρας, λευκό

Περιγραφή που δημιουργήθηκε αυτόματα

Ορίζουμε δύο 1Δ φίλτρα Gaussian Gy​ και Gx ​:

**Εικόνα που περιέχει κείμενο, γραμματοσειρά, λευκό, τυπογραφία

Περιγραφή που δημιουργήθηκε αυτόματα**

Eφαρμόζουμε τη συνέλιξη με το 1Δ φίλτρο

Gaussian κατά τον άξονα x:

Εικόνα που περιέχει γραμματοσειρά, κείμενο, γραφικός χαρακτήρας, λευκό

Περιγραφή που δημιουργήθηκε αυτόματαΕφαρμόζουμε τη συνέλιξη με το 1Δ φίλτρο

Gaussian κατά τον άξονα y:

Oι διακυμάνσεις των 1Δ φίλτρων είναι σχ​ και σy τότε η διακύμανση του 2Δ φίλτρου είναι:

σ22Δ= σ2χ +σ2y

​Για συμμετρικά φίλτρα ισχύει σx=σy=σ, η διακύμανση του 2Δ φίλτρου είναι:

σ22Δ= 2σ2

1. *Code commitment done*
2. Εικόνα που περιέχει κείμενο, λογισμικό, λογισμικό πολυμέσων, τοποθεσία web

   Περιγραφή που δημιουργήθηκε αυτόματα**Aποτέλεσμα εκτέλεσης του αρχείου** “filters.py”

Το Gaussian φιλτράρισμα εφαρμόζει μία θόλωση στην εικόνα, μειώνοντας τον θόρυβο και τις λεπτομέρειες, οδηγώντας σε μια πιο ομαλή και μαλακή όψη της εικόνας.

Παρατηρώντας πιο προσεκτικά μπορούμε να το διαπιστώσουμε και στην υλοποίηση μας .

Εικόνα που περιέχει κείμενο, φιλμ ακτινογραφίας, μαύρο, ιατρική απεικόνιση

Περιγραφή που δημιουργήθηκε αυτόματα

1. Είναι καλή ιδέα για ένα φίλτρο εξομάλυνσης να έχει άθροισμα συντελεστών ίσο με 1 για τους εξής λόγους:

* **Διατήρηση της Φωτεινότητας**: Όταν οι συντελεστές του φίλτρου αθροίζουν στο 1, η συνολική φωτεινότητα της εικόνας δεν μεταβάλλεται.
* **Σταθερότητα της Έντασης**: Ένα μη κανονικοποιημένο φίλτρο μπορεί να αυξήσει ή να μειώσει τις εντάσεις της εικόνας, προκαλώντας ανεπιθύμητες αλλαγές στη φωτεινότητα και τον αντίθεση.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Όπως φαίνεται και από τα plot που δημιούργησαμε

1. Οι παράγωγοι της εικόνας υπολογίζονται ως εξής:–

Ix(x,y)=I(x+1,y)−I(x−1,y)=>

Ix​(x,y)=(−1⋅I(x−1,y))+(0⋅I(x,y))+(1⋅I(x+1,y))=>

kx​=[−1,0,1]

Iy(x,y)=I(x,y+1)−I(x,y−1)=>

Iy​(x,y)=(−1⋅I(x,y−1))+(0⋅I(x,y))+(1⋅I(x,y+1))=>

[-1]

Ky= [0]

[1]

1. Code commitment done
2. **Plots:**

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Μετά την εφαρμογή της ρουτίνας edge\_detection() στις δύο εικόνες, μπορούμε να παρατηρήσουμε τα εξής:

* **Αρχική Εικόνα**: Οι κλίσεις είναι πιο έντονες και οι άκρες είναι πιο ευδιάκριτες. Η εικόνα περιέχει περισσότερο θόρυβο, ο οποίος επηρεάζει τις κλίσεις.
* **Φιλτραρισμένη Εικόνα**: Οι κλίσεις είναι πιο ομαλές και οι άκρες είναι λιγότερο ευδιάκριτες σε σχέση με την αρχική εικόνα.

**Ζήτημα 3: Τελεστής Sobel**

Εικόνα που περιέχει γραμματοσειρά, κείμενο, λευκό, τυπογραφία

Περιγραφή που δημιουργήθηκε αυτόματα

1. Η οριζόντια παράγωγος της φιλτραρισμένης εικόνας I′ είναι:

Εικόνα που περιέχει γραμματοσειρά, γραφικός χαρακτήρας, καλλιγραφία, τυπογραφία

Περιγραφή που δημιουργήθηκε αυτόματαΗ εφαρμογή του φίλτρου Sobel Sx σε μια εικόνα που έχει φιλτραριστεί με το Gaussian φίλτρο Gs προσεγγίζει την οριζόντια παράγωγο της φιλτραρισμένης εικόνας. Δηλαδή:

Με αυτόν τον τρόπο, αποδεικνύουμε ότι οι τελεστές Sobel μπορούν να χρησιμοποιηθούν για να προσεγγίσουν τις παραγώγους της φιλτραρισμένης εικόνας.

1. Code commitment done
2. Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, διάγραμμα, σκίτσο/σχέδιο

   Περιγραφή που δημιουργήθηκε αυτόματα

 **I∗Sx​**: Αναδεικνύονται οι οριζόντιες ακμές στην εικόνα, καθώς το φίλτρο Sobel Sx​ είναι σχεδιασμένο να εντοπίζει αλλαγές στην ένταση κατά την οριζόντια κατεύθυνση.

 **I∗Sy** Αναδεικνύονται οι κατακόρυφες ακμές στην εικόνα, καθώς το φίλτρο Sobel Sy​ είναι σχεδιασμένο να εντοπίζει αλλαγές στην ένταση κατά την κατακόρυφη κατεύθυνση.

 **Gradient Magnitude**: Το μέγεθος της κλίσης αναδεικνύει όλες τις ακμές, συνδυάζοντας τις πληροφορίες από τις παραγώγους κατά x & y .

**Ζήτημα 4: Λαπλασιανή της Γκαουσιανής της εικόνας (LoG)**

Τα φίλτρα LoG (Laplacian of Gaussian) χρησιμοποιούνται για την ανίχνευση ακμών στην εικόνα.

**Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, μέτρο/χάρακας

Περιγραφή που δημιουργήθηκε αυτόματα**

**LoG Filter 1**: Είναι μικρότερο σε μέγεθος και μπορεί να ανιχνεύσει λεπτές ακμές και μικρές μεταβολές.

**LoG Filter 2**: Είναι μεγαλύτερο και είναι πιο κατάλληλο για την ανίχνευση μεγαλύτερων και πιο εκτεταμένων ακμών στην εικόνα

Tο Laplacian τονίζει περιοχές της εικόνας όπου υπάρχει απότομη μεταβολή στην ένταση των εικονοστοιχείων. Οι ανιχνευμένες περιοχές είναι τα εικονοστοιχεία όπου το φίλτρο έχει υψηλή απόκριση κατ’ απόλυτη τιμή

# Αξιοσημείωτες Πήγες Μελέτης

 [Εcourse Ψηφιακή Επεξεργασία Εικόνας](https://ecourse.uoi.gr/course/view.php?id=573)

 [OpenGenus IQ: Laplacian Filter](https://iq.opengenus.org/laplacian-filter/)

 [Automatic Addison: How the Laplacian of Gaussian Filter Works](https://automaticaddison.com/how-the-laplacian-of-gaussian-filter-works/)

 [TheAILearner: Laplacian of Gaussian (LoG)](https://theailearner.com/2019/05/25/laplacian-of-gaussian-log/)

 [University of Edinburgh: Laplacian of Gaussian (LoG)](http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm)

 [StackExchange: Gaussian Blurring](https://dsp.stackexchange.com/questions/1039/why-do-we-normalize-the-gaussian-filter)

 Scipy Documentation: Convolution and Gaussian Filters

 [AskPython: Gaussian Kernel](https://www.askpython.com/python-modules/numpy/gaussian-kernel-matrix-numpy)

 [Delft Stack: Gaussian Kernel](https://www.delftstack.com/howto/python/gaussian-kernel-python/)

 [Wikipedia: Sobel Operator](https://en.wikipedia.org/wiki/Sobel_operator)

 OpenCV Documentation: Sobel

 GeeksforGeeks: Sobel Operator

 TheAILearner: Sobel Edge Detection