Γραφική με Υπολογιστές 2021

Εργασία #1: Πλήρωση Τριγώνων

Ζητούμενα

Α. Συνάρτηση γραμμικής παρεμβολής

Να δημιουργηθεί μία συνάρτηση η οποία θα υλοποιεί γραμμική παρεμβολή ανάμεσα σε δύο τρισδιάστατες τιμές C_1 και C_2 με βάση τις δισδιάστες συντεταγμένες δύο σημείων x_1 και x_2 των κορυφών ενός τριγώνου.

$$value = interpolate\_color(x_1, x_2, x, C_1, C_2)$$

όπου:

- x_1 και x_2 οι αντίστοιχες συνιστώσες των δισδιάστατων συντεταγμένων δύο κορυφών ενός τριγώνου (είτε οι οριζόντιες, είτε οι κάθετες συντεταγμένες).
- C_1 και C_2 οι τρισδιάστατες τιμές χρώματος που αντιστοιχούν στις κορυφές των συντεταγμένων x_1 και x_2.
- x το σημείο στο οποίο θα εφαρμοστεί η παρεμβολή.
- $value$ είναι η τιμή που προκύπτει από γραμμική παρεμβολή των C_1 και C_2.

Η συνάρτηση αυτή θα σας είναι χρήσιμη και σε επόμενες εργασίες.

Β. Συναρτήσεις Πλήρωσης Τριγώνων

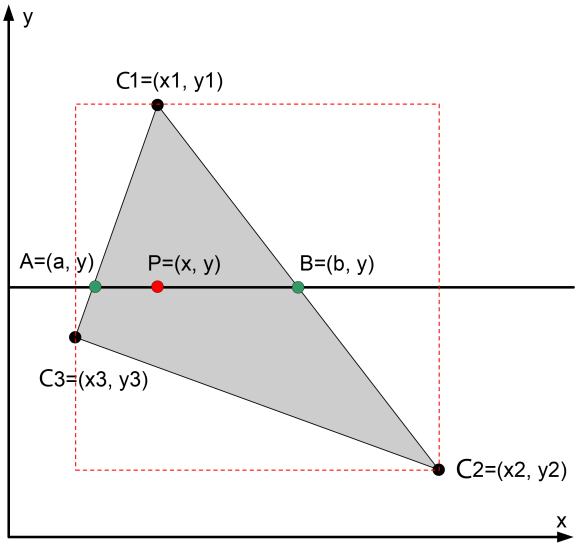
Σκοπός της εργασίας είναι η υλοποίηση ενός αλγορίθμου πλήρωσης τριγώνων με βάση τον αλγόριθμο πλήρωσης πολυγώνων που περιγράφεται στις σημειώσεις. Αφού προσαρμόσετε κατάλληλα τον αλγόριθμο των σημείωσεων ώστε να χειρίζεται την ειδική περίπτωση των τριγώνων εξετάστε τις δύο διαφορετικές εκδοχές απόδοσης χρώματος που περιγράφονται στη συνέχεια.

Έστω τρίγωνο που ορίζεται από κορυφές με ακέραιες συντεταγμένες. Υποθέστε καμβά διάστασης $M \times N$. Δημιουργήστε τη συνάρτηση,

$$Y = shade\_triangle(img, verts2d, vcolors, shade\_t)$$

Όπου:

- img είναι η εικόνα (πίνακας διάστασης $M \times N \times 3$) με τυχόν προϋπάρχοντα τρίγωνα.
- $verts2d$ είναι ακέραιος πίνακας διάστασης 3×2 που σε κάθε γραμμή περιέχει τις δισδιάστατες συντεταγμένες μιας κορυφής του τριγώνου.
- $vcolors$ είναι πίνακας διάστασης 3×3 που σε κάθε γραμμή περιέχει το χρώμα μιας κορυφής του τριγώνου σε μορφή RGB (με τιμές στο διάστημα $[0, 1]$).
- $shade\_t$ είναι όρισμα τύπου string το οποίο μπορεί να πάρει τιμές $[" flat", "gouraud"]$, όπου στη κάθε περιπτώση θα πραγματοποιείται η αντίστοιχη παραλλαγή του αλγόριθμου πλήρωσης τριγώνων, και διαφέρει στον τρόπο υπολογισμού του χρώματος.
- Y είναι πίνακας διάστασης $M \times N \times 3$ που για όλα τα σημεία του τριγώνου περιέχει τις υπολογισμένες χρωματικές συνιστώσες (R_i, G_i, B_i) καθώς και τα προϋπάρχοντα τρίγωνα της εισόδου img (επικαλύπτοντας τυχόν κοινά χρωματισμένα σημεία που προϋπήρχαν από την πλήρωση άλλων τριγώνων).



Σχήμα 1: Παράδειγμα χρωματισμού

Υλοποίηση με όρισμα $shade\_t = " flat"$

Η $shade\_triangle$ με όρισμα $shade\_t = " flat"$ θα αποδίδει σε κάθε τρίγωνο ένα μοναδικό χρώμα. Συγκεκριμένα, κάθε τρίγωνο θα χρωματίζεται με το χρώμα που προκύπτει ως το κέντρο βάρους (μέσος όρος) του χρώματος των κορυφών του.

Υλοποίηση με όρισμα $shade\_t = "gouraud"$

Η $shade\_triangle$ με όρισμα $shade\_t = "gouraud"$ θα υπολογίζει το χρώμα των σημείων του τριγώνου με γραμμική παρεμβολή από το χρώμα των κορυφών του. Συγκεκριμένα, για το χρωματισμό

του τριγώνου, με αναφορά στα σημεία του σχήματος 1, πρώτα θα υπολογίζεται το χρώμα στις θέσεις A και B, με χρήση της συνάρτησης $interpolate\_color$ για τις τιμες χρωμάτων των κορυφών $V1, V3$ και $V1, V2$ αντίστοιχα (δίνοντας τις κατάλληλες συνιστώσες των συντεταγμένων των κορυφών του τριγώνου). Η πρώτη αυτή φάση τα υλοποιείται μία φορά για κάθε scanline y.

Σε δεύτερη φάση, θα πρέπει πάλι με την χρήση της $interpolate\_color$ να γίνει γραμμική παρεμβολή για κάθε σημείο $P = (x, y)$ που ανήκει στο τρέχων scanline.

Β. Συναρτήση χρωματισμού αντικειμένου

Να υλοποιήσετε τη συνάρτηση:

$$img = render(verts2d, faces, vcolors, depth, shade\_t)$$

Όπου:

- img είναι έγχρωμη εικόνα διάστασης $M \times N \times 3$. Η εικόνα θα περιέχει K χρωματισμένα τρίγωνα τα οποία σχηματίζουν την προβολή ενός 3D αντικειμένου στις 2 διαστάσεις.
- $verts2d$ είναι ο πίνακας με τις κορυφές των τριγώνων της εικόνας. Ο πίνακας $verts2d$ είναι διάστασης $L \times 2$ και περιέχει τις συντεταγμένες ενός πλήθους L κορυφών. Για απλούστευση υποθέστε ότι όλες βρίσκονται εντός του καμβά.
- $faces$ είναι ο πίνακας που περιέχει τις κορυφές των K τριγώνων. Ο πίνακας είναι διάστασης $K \times 3$. Η i-στη γραμμή του πίνακα δηλώνει τις τρεις κορυφές που σχηματίζουν το τρίγωνο (με αναφορά σε κορυφές του πίνακα $verts2d$ και αρίθμηση που ξεκινά από το 1).
- $vcolors$ είναι ο πίνακας με τα χρώματα των κορυφών. Ο πίνακας $vcolors$ είναι διάστασης $L \times 3$. Η i-στη γραμμή του πίνακα δηλώνει τις χρωματικές συνιστώσες της αντίστοιχης κορυφής.
- $depth$ είναι ο πίνακας που δηλώνει το βάθος της κάθε κορυφής πριν την προβολή του αντικειμένου στις 2 διαστάσεις. Ο πίνακας $depth$ είναι διάστασης $L \times 1$
- $shade\_t$ είναι παρόμοια μεταβλητή ελέγχου (τύπου string) που καθορίζει τη συνάρτηση χρωματισμού (Gouraud ή Flat) που θα χρησιμοποιηθεί, και μπορεί να πάρει τιμές $[" flat", "gouraud"]$.
- M και N είναι το ύψος και το πλάτος του καμβά αντίστοιχα.

Στο εσωτερικό της συνάρτησης $render$ θα καλείται η $shade\_triangle$, η οποία ανάλογα με την τιμή της μεταβλητής $shade\_t$, θα καλείται η ανάλογη ρουτίνα για το χρωματισμό των εσωτερικών σημείων κάθε τριγώνου. **Η σειρά με την οποια πρέπει να χρωματιστούν τα τρίγωνα προκύπτει από τον πίνακα βάθους** $depth$. Ο χρωματισμός θα πρέπει να ξεκινάει από τα μακρινότερα (αυτά με το μεγαλύτερο βάθος) τρίγωνα και να συνεχίζει με τα κοντινότερα. Το βάθος ενός τριγώνου υπολογίζεται ως το κέντρο βάρους τους βάθους των κορυφών του.

Θεωρήστε δεδομένο ότι:

- Το background του καμβά είναι λευκό (rgb = (1,1,1)).
- Ο καμβάς έχει διαστάσεις $M = 512$, $N = 512$.
- Προσπαθήστε όπου είναι δυνατό στον κώδικα, σας να χρησιμοποιήσετε vectorization προκειμένου να αποφύγετε υπερβολικά μεγάλους χρόνους εκτέλεσης (ο τρόπος για να το κάνετε αυτό στη βιβλιοθήκη numpy, βλ. 1, είναι παρόμοιος με το matlab).

Παραδοτέα

1. Οι παραπάνω συναρτήσεις σε μορφή **σχολιασμένου** πηγαίου κώδικα Python ($= v3.7$) με σχόλια γραμμένα στα **αγγλικά** ή **greeklish**. (κοινώς, **μη γράφετε σχόλια με ελληνικούς χαρακτήρες)**.
2. Δύο scripts επίδειξης με ονόματα demo_gouraud.py, demo_flat.py, τα οποία θα καλούνται χωρίς εξωτερικά ορίσματα και θα:
   (α΄) Φορτώνουν τα δεδομένα εισόδου (πίνακες $verts2d$, $vcolors$, $faces$, $depth$ της render) από το αρχείο hw1.npy που σας δίνεται.
   (β΄) Πραγματοποιούν τον χρωματισμό των τριγώνων του αντικειμένου χρησιμοποιώντας τη συναρτήση render (με την επιλογή του αντιστοιχού τύπου χρωματισμού, "gouraud" ή "flat").
   (γ΄) Αποθηκεύουν την τελική εικόνα (για τον τρόπο αποθήκευσης δείτε το Υπόμνημα).
3. Αναφορά με:
   - Περιγραφή της λειτουργίας και του τρόπου κλήσης των προγραμμάτων,
   - Περιγραφή της διαδικασίας χρωματισμού των τριγώνων που ακολουθήσατε και παρουσίαση του αντίστοιχου ψευδοκώδικα
   - Περιγραφή των παραδοχών που χρησιμοποιήσατε.
   - Τα ενδεικτικά αποτελέσματα που παράγονται από τα demos.

Υποβολή εργασίας

- Οι εργασίες είναι **αυστηρά** ατομικές.
- Υποβάλετε ένα και μόνο αρχείο, τύπου zip.
- Το όνομα του αρχείου πρέπει να είναι `AEM.zip`, όπου `AEM` είναι τα τέσσερα ψηφία του Α.Ε.Μ. σας.
- Το προς υποβολή αρχείο πρέπει να περιέχει τα αρχεία κώδικα python (version = 3.7) και το αρχείο `report.pdf` το οποίο θα είναι η αναφορά της εργασίας.
- Η αναφορά πρέπει να είναι ένα αρχείο τύπου PDF, και να έχει όνομα `report.pdf`.
- Όλα τα αρχεία κώδικα πρέπει να είναι αρχεία κειμένου τύπου UTF-8, και να έχουν κατάληξη `.py`.
- **Το αρχείο τύπου zip που θα υποβάλετε ΔΕΝ πρέπει να περιέχει κανέναν εσωτερικό υποφάκελο.**
- Για την ονομασία των αρχείων που περιέχονται στο προς υποβολή αρχείο, χρησιμοποιείτε μόνο αγγλικούς χαρακτήρες, και όχι ελληνικούς ή άλλα σύμβολα, πχ "#", "$", "%" κλπ.

Προσοχή: Θα αξιολογηθούν μόνο όσες εργασίες έχουν demos που τρέχουν!

1 Υπόμνημα

Σε αυτό το υπόμνημα ακολουθούν οδηγίες για την εγκατάσταση της python, καθώς και τις βιβλιοθήκες που θα σας χρησιμεύσουν για την υλοποίηση της εργασίας.

Εγκατάσταση

Για την εγκατάστση της python σε περιβάλλον Windows μπορείτε πολύ εύκολα να κατεβάσετε τον αντίστοιχο installer από το επίσημο site της python. Μεταβείτε στην έκδοση 3.7, κατεβάστε τον installer, και κάνετε install.

Για την εγκατάσταση της python σε περιβάλλον Linux (και συγκεκριμένα Ubuntu ≥ 18.04) μπορείτε να κάνετε τις ακόλουθες ενέργειες σε ένα terminal:

sudo apt update
sudo apt install software-properties-common
sudo add-pat-repository ppa:deadsnakes/ppa
sudo apt install python3.7

Code editor

Ένας ευρέως διαδεδομένος code editor (για πολλές γλώσσες προγραμματισμού αλλά ειδικά και για python) είναι το VS Code της Microsoft, ο οποίος πάλι μπορεί να εγκατασταθεί πολύ εύκολα και σε Windows και σε Ubuntu.

Το VS Code δέχεται extensions (τα οποία μπορείτε να εγκαταστήσετε πολύ εύκολα μέσω της ίδιας της εφαρμογής). Για τη λειτουργία του VS Code με Python και για να μπορείτε να χρησιμοποιήσετε debugger, καλό θα ήταν να εγκαταστήσετε το extension για την Python.

Υλοποίηση και βιβλιοθήκες

Για την υλοποίηση της εργασίας θα χρησιμοποιηθεί σε μεγάλο βαθμό η βιβλιοθήκη **numpy**, μία πολύ ευρέως γνωστή βιβλιοθήκη για χρήση και επεξεργασία πινάκων και τανυστών, της οποίας οι δυνατότητες αντιστοιχούν με σχεδόν 1-1 τρόπο με το Matlab. Για να εγκαταστήσετε τη βιβλιοθήκη numpy μπορείτε πολύ εύκολα σε ένα τερματικό να τρέξετε την ακόλουθη εντολή:

pip install numpy

pip είναι ο package manager της python, και σχεδόν όλες οι βιβλιοθήκες της python μπορούν να εγκατασταθούν με παρόμοιες εντολές.

Ακόμα μία χρήσιμη βιβλιοθήκη είναι η OpenCV (pip install opencv-python), η οποία μπορεί να σας χρησιμεύσει για να διαβάζετε και να αποθηκεύετε εικόνες. Επίσης μία ακόμα βιβλιοθήκη, χρήσιμη για διαδραστική απεικόνιση εικόνων (και με συντακτικό πολύ παρόμοιο με matlab) είναι η matplotlib (pip install matplotlib).

Για οποιαδήποτε απορία σχετικά με την εργασία, την python και τη χρήση της για την υλοποίση της εργασίας, μη διστάσετε να επικοινωνήσετε με τον κ. Αντώνη Καρακώττα (akarakott@ece.auth.gr)