



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εργασία 1: Πλήρωση Τριγώνων

Δήμητρα Γαλάνη
ΑΕΜ 10331

Μάθημα: Γραφική με Υπολογιστές
Διδάσκων: Ντελόπουλος Αναστάσιος
Καθηγητής Α.Π.Θ.

7 Απριλίου 2024

Περιεχόμενα

1	Συνάρτηση Vector Interpolation	2
1.1	Λειτουργία	2
1.2	Κλήση συνάρτησης	2
2	Συνάρτηση f_triangle	4
2.1	Λειτουργία	4
2.2	Κλήση συνάρτησης	4
3	Συνάρτηση f_shading - Flat Shading	5
3.1	Λειτουργία	5
3.2	Κλήση συνάρτησης	5
4	Συνάρτηση gouraud_shading	6
4.1	Λειτουργία	6
4.2	Κλήση συνάρτησης	7
5	Συνάρτηση render_img	8
5.1	Λειτουργία	8
5.2	Κλήση συνάρτησης	8
6	Παραδοχές	9
7	Demos	10
7.1	Flat Shading	10
7.2	Gouraud Shading	10

Κεφάλαιο 1

Συνάρτηση Vector Interpolation

Η συνάρτηση `vector_interp` παίρνει ως ορίσματα τις θέσεις δύο σημείων καθώς και τα δύο διανύσματα που αντιστοιχούν στις θέσεις των σημείων αυτών. Σκοπός της συνάρτησης είναι να επιστρέψει την τιμή του διανύσματος κάνοντας γραμμική παρεμβολή των δύο διανυσμάτων σε ένα επιθυμητό σημείο. Το επιθυμητό σημείο καθορίζεται από τις μεταβλητές `coord` και `dim`. Η συνάρτηση αυτή χρησιμοποιείται στην πλήρωση τριγώνων κατά Gouraud.

1.1 Λειτουργία

Εάν το `dim = 1` θεωρώ ότι κάνω παρεμβολή οριζοντίως, δηλαδή τα σημεία βρίσκονται στο ίδιο ύψος, όπως θα δούμε παρακάτω στην εργασία χρησιμοποιείται όταν κάνω γραμμική παρεμβολή σε ένα scan line όπου όλα τα pixel έχουν την ίδια τεταγμένη y . Σε αυτήν την περίπτωση υπολογίζω τον συντελεστή α ως εξής:

```
alpha = (coord - p1[0]) / (p2[0] - p1[0])
```

Ο συντελεστής α εκφράζει το πόσο κοντά βρίσκεται το σημείο p στο σημείο p_2 αναλογικά ως προς το μήκος του ευθύγραμμου τμήματος p_1p_2 . Οπότε κάνοντας γραμμική παρεμβολή για τα διανύσματα V_1 και V_2 χρησιμοποιώντας τον συντελεστή αυτόν θα υπολογίζεται ανάλογα, το διάνυσμα V . Αν, για παράδειγμα, τα διανύσματα V_1, V_2 περιέχουν τις τιμές του χρώματος δύο σημείων σε RGB τιμές, η τιμή του χρώματος του σημείου p θα εξαρτάται από το πόσο κοντά βρίσκεται αυτό το σημείο σε κάθε άκρο του ευθύγραμμου τμήματος και με βάση τον συντελεστή α το χρώμα κάθε άκρου θα συμβάλλει περισσότερο ή λιγότερο.

```
V = V1 * (1-alpha) + V2 * alpha
```

Με όμοιο τρόπο δουλεύει η συνάρτηση για `dim = 2` μόνο που σε αυτήν την περίπτωση θεωρώ ότι κάνω γραμμική παρεμβολή καθέτως. Οπότε εργάζομαι με τις τεταγμένες των σημείων.

Στην συνάρτηση αυτήν ενσωματώνεται και η περίπτωση που τα δύο σημεία ταυτίζονται. Σε αυτήν την περίπτωση θεωρώ ότι το διάνυσμα του σημείου θα πάρει τον μέσο όρο των άλλων δύο διανυσμάτων. Περισσότερες πληροφορίες για την λειτουργία της συνάρτησης υπάρχουν στα σχόλια του κώδικα.

1.2 Κλήση συνάρτησης

Για να καλέσω την συνάρτηση θα πρέπει τα ορίσματα να έχουν τις εξής ιδιότητες:

- Οι p_1, p_2 πρέπει να είναι πίνακες μεγέθους 1×2 .
- Οι V_1, V_2 μπορούν να είναι πίνακες οποιουδήποτε μεγέθους αρκεί να έχουν και οι δύο το ίδιο μέγεθος.
- Η `coord` είναι μια τιμή, η οποία συμβολίζει την τεταγμένη του σημείου στο οποίο θέλω να γίνει η γραμμική παρεμβολή όταν το `dim = 1`, ενώ όταν `dim = 2` συμβολίζει την τεταγμένη του σημείου αυτού.

- Η *dim* είναι μια τιμή που πρέπει να παίρνει είτε την τιμή 1 είτε την τιμή 2, ανάλογα με το εάν επιθυμώ να κάνω γραμμική παρεμβολή οριζοντίως ή καθέτως.

Στην εργασία ένας τρόπος κλήσης της είναι:

```
img[y, x] = vector_interp(sorted_vertices[0], sorted_vertices[1], sorted_vcolors[0],  
                           sorted_vcolors[1], x, 1)
```

Κεφάλαιο 2

Συνάρτηση `f_triangle`

Έχει υλοποιηθεί μια βοηθητική συνάρτηση `f_triangle`, η οποία έχει ως όρισμα έναν πίνακα `vertices` μεγέθους 3×2 , ο οποίος περιλαμβάνει τις συντεταγμένες των τριών κορυφών ενός τριγώνου και επιστρέφει κάποια χρήσιμα μεγέθη όπως η κλίση της ευθείας κάθε ακμής και τα άνω και κάτω όρια του τριγώνου.

2.1 Λειτουργία

Η συνάρτηση αυτή παίρνει τα ζεύγη των κορυφών που σχηματίζουν μια ακμή και βρίσκει το μέγιστο και το ελάχιστο της ακμής αυτής τόσο στον άξονα x όσο και στον άξονα y , με σκοπό να καθοριστούν τα όρια κάθε ακμής.

Επίσης, υπολογίζει την κλίση της ευθείας που ορίζεται από τα δύο αυτά σημεία καθώς και τον συντελεστή b . Είναι σημαντικό να είναι γνωστές κάθε στιγμή οι εξισώσεις των ευθειών που σχηματίζουν το τρίγωνο, προκειμένου να μπορούν να υπολογιστούν εύκολα τα ενεργά οριακά σημεία. Λαμβάνεται υπόψη και η περίπτωση που η ακμή είναι κάθετη.

Τέλος, η συνάρτηση αυτή επιστρέφει την μέγιστη και την ελάχιστη τεταγμένη σε τύπο `int`, οι οποίες οριοθετούν τις τεταγμένες που θα δουλέψουν οι αλγόριθμοι πλήρωσης του τριγώνου. Περισσότερες πληροφορίες για την λειτουργία της συνάρτησης υπάρχουν στα σχόλια του κώδικα.

2.2 Κλήση συνάρτησης

Για να καλέσω την συνάρτηση θα πρέπει το όρισμα, ο πίνακας `vertices`, να έχει μέγεθος 3×2 . Εφόσον επιστρέφει παραπάνω από μία τιμές πρέπει να διασφαλίσω ότι όλες αυτές οι τιμές θα αποθηκευτούν κάπου όποτε η κλήση της πρέπει να γίνει με τον παρακάτω τρόπο:

```
# Calculate the variables needed for the triangle definition
xmin_xmax_array, ymin_ymax_array, slope, b, ymin, ymax = f_triangle(vertices)
```

Κεφάλαιο 3

Συνάρτηση `f_shading` - Flat Shading

Ο σκοπός της συνάρτησης είναι να εντοπίσει όλα τα pixels που βρίσκονται εντός των ορίων του τριγώνου και να τα χρωματίσει με το μέσο όρο των χρωμάτων των τριών κορυφών.

3.1 Λειτουργία

Αρχικά, υπολογίζεται ο μέσος όρος των χρωμάτων των τριών κορυφών και καλείται η συνάρτηση `f_triangle`. Ο κώδικας για την πλήρωση του τριγώνου θα τρέξει μια φορά για κάθε ένα scan line. Αρχικά βρίσκω τις ενεργές πλευρές για το εκάστοτε scan line με την βοήθεια του `ymin_ymax_array`. Στην συνέχεια, βρίσκω τα οριακά ενεργά σημεία από την εξίσωση της ευθείας, τα οποία τα ταξινομώ κατά αύξουσα τιμή x . Τα active limit points δεν είναι πάντα ακέραιοι, ενώ τα έχουν pixel ακέραιες συντεταγμένες εξ' ορισμού, οπότε το πρώτο pixel που θα ανήκει μέσα στο τρίγωνο θα είναι το ceiling του πρώτου active limit point και το τελευταίο pixel θα είναι το floor του τελευταίου.

Επειδή, οι υπολογισμοί των active limit points γίνονται με floating points, ενδέχεται να υπάρχουν κάποια errors στα τελευταία δεκαδικά ψηφία, για τον λόγο αυτό κάνω στρογγυλοποίηση σε ένα αρκετά προχωρημένο δεκαδικό ψηφίο, προκειμένου αφενός να προλαμβάνονται τέτοια λάθη αλλά να μην αλλοιώνεται το αποτέλεσμα των πράξεων. Τέλος, αξίζει να σημειωθεί ότι περνάω τις τιμές στην εικόνα με αυτόν τον τρόπο: `img[y, x]`, καθώς κατά την αναπαράσταση της εικόνας με την χρήση της βιβλιοθήκης `matplotlib` τα pixels γίνονται προσβάσιμα με τον εξής τρόπο: `[row_index, column_index]` το οποίο αντιστοιχεί στις `[y, x]` καρτεσιανές συντεταγμένες. Περισσότερες πληροφορίες για την λειτουργία της συνάρτησης υπάρχουν στα σχόλια του κώδικα.

3.2 Κλήση συνάρτησης

Η συνάρτηση αυτή καλείται εντός της συνάρτησης `render_image`. Για να καλέσω την συνάρτηση θα πρέπει να συμπεριλάβω τα παρακάτω ως ορίσματα:

- την εικόνα `img`, έναν πίνακα $M \times N \times 3$ με τα ήδη προϋπάρχοντα χρωματισμένα τρίγωνα, την οποία ανανεώνει χρωματίζοντας το εν λόγω τρίγωνο.
- τον πίνακα `vertices` μεγέθους 3×2 , ο οποίος περιλαμβάνει τις συντεταγμένες των τριών κορυφών του τριγώνου.
- τον πίνακα `vcolors` μεγέθους 3×3 , ο οποίος περιλαμβάνει τα χρώματα των τριών κορυφών του τριγώνου σε RGB τιμές.

Ως έξοδο επιστρέφει την ανανεωμένη εικόνα, οπότε ένας τρόπος κλήσης της είναι:

```
img = f_shading(img, vertices_triangle[i], vcolors_triangle[i])
```

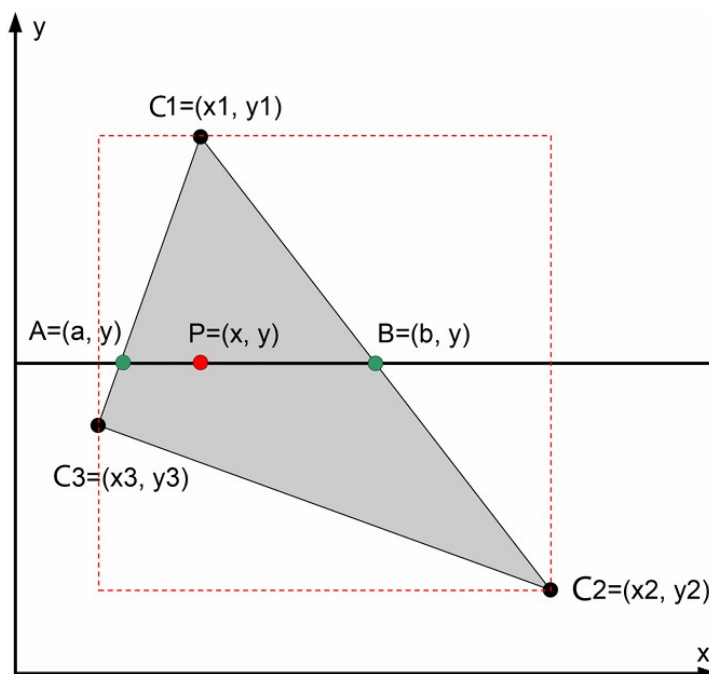
Κεφάλαιο 4

Συνάρτηση gouraud_shading

Ο σκοπός της συνάρτησης είναι να εντοπίσει όλα τα pixels που βρίσκονται εντός των ορίων του τριγώνου και να χρωματίσει το καθένα με κατάλληλο χρώμα το οποίο εξαρτάται από την θέση του εκάστοτε pixel και υπολογίζεται με γραμμική παρεμβολή από το χρώμα των κορυφών του.

4.1 Λειτουργία

Η λειτουργία της συνάρτησης είναι παρόμοια με την λειτουργία της συνάρτησης flat_shading. Ωστόσο υπάρχουν αρκετά σημεία που τις διαφοροποιούν καθώς έχουμε να διαχειριστούμε κατάλληλα και το διαφορετικό χρώμα των pixels. Η διαχείριση των διαφορετικών παραδοχών εξηγείται σε παρακάτω κεφάλαιο. Στην γενική περίπτωση βρίσκω με τον ίδιο τρόπο τις ενεργές πλευρές για το εκάστοτε scan line και τα αντίστοιχα active limit points. Αρχικά για κάθε ενεργή πλευρά βρίσκω το χρώμα στο ύψος του scan line με γραμμική παρεμβολή στον άξονα y που εξαρτάται από τα χρώματα των κορυφών της πλευράς αυτής. Δηλαδή, βρίσκω το χρώμα στα σημεία A, B με βάση τα χρώματα των κορυφών C_1, C_3 και C_1, C_2 . Έπειτα, αφότου ταξινομήσω τα ενεργά οριακά σημεία κατά αύξουσα τιμή x, εφαρμόζω γραμμική παρεμβολή για τα σημεία του scan line ως προς το άξονα x και χρωματίζω τα pixel ανάλογα με το πόσο απέχουν από τα δύο ενεργά οριακά σημεία A και B, τα οποία θεωρώ ότι έχουν το χρώμα που βρήκα πριν από την γραμμική παρεμβολή κατά y. Περισσότερες πληροφορίες για την λειτουργία της συνάρτησης υπάρχουν στα σχόλια του κώδικα.



Σχήμα 4.1: Γενική περίπτωση τριγώνου

4.2 Κλήση συνάρτησης

Η συνάρτηση αυτή καλείται εντός της συνάρτησης `render_image`. Για να καλέσω την συνάρτηση θα πρέπει να συμπεριλάβω τα παρακάτω ως ορίσματα:

- την εικόνα `img`, έναν πίνακα $M \times N \times 3$ με τα ήδη προϋπάρχοντα χρωματισμένα τρίγωνα, την οποία ανανεώνει χρωματίζοντας το εν λόγω τρίγωνο.
- τον πίνακα `vertices` μεγέθους 3×2 , ο οποίος περιλαμβάνει τις συντεταγμένες των τριών κορυφών του τριγώνου.
- τον πίνακα `vcolors` μεγέθους 3×3 , ο οποίος περιλαμβάνει τα χρώματα των τριών κορυφών του τριγώνου σε RGB τιμές.

Ως έξοδο επιστρέφει την ανανεωμένη εικόνα, οπότε ένας τρόπος κλήσης της είναι:

```
img = gouraud_shading(img, vertices_triangle[i], vcolors_triangle[i])
```

Τέλος, αξίζει να σημειωθεί ότι για την ανάπτυξη του κώδικα χρησιμοποιήθηκε ο ψευδοκώδικας των σημειώσεων στην σελίδα 35 του κεφαλαίου 3.2. ΠΛΗΡΩΣΗ ΚΛΕΙΣΤΩΝ ΣΧΗΜΑΤΩΝ FILLING με τις κατάλληλες αλλαγές τόσο για να προσαρμοστεί στα πρότυπα ενός τριγώνου όσο και για να συμπεριληφθούν όλες οι ειδικές περιπτώσεις τριγώνων.

Κεφάλαιο 5

Συνάρτηση `render_img`

Σκοπός της συνάρτησης είναι να χρωματίσει τα τρίγωνα που της δίνονται με σειρά προτεραιότητας που καθορίζεται από το βάθος του κάθε τριγώνου.

5.1 Λειτουργία

Αρχικά υπολογίζει το βάθος κάθε τριγώνου ως το μέσο βάθος των κορυφών του και ταξινομεί τα τρίγωνα με βάση το βάθος αυτό. Για κάθε τρίγωνο καλεί την επιθυμητή συνάρτηση `f_shading` ή `gouraud_shading` ανάλογα με την τιμή `shading`, έχοντας εντοπίσει τις συντεταγμένες και τα χρώματα του επιθυμητού τριγώνου. Περισσότερες πληροφορίες για την λειτουργία της συνάρτησης υπάρχουν στα σχόλια του κώδικα.

5.2 Κλήση συνάρτησης

Η συνάρτηση αυτή καλείται εντός των δύο demos. Για να καλέσω την συνάρτηση θα πρέπει να συμπεριλάβω τα παρακάτω ως ορίσματα:

- τον πίνακα `faces`, μεγέθους $K \times 3$, κάθε γραμμή περιέχει τα τρία `id` των κορυφών του τριγώνου τα οποία χρησιμοποιούνται για να προσπελάσει τους πίνακες `vertices` και `vcolors` και να εντοπίσει τις πληροφορίες του κάθε τριγώνου.
- τον πίνακα `vertices`, μεγέθους $L \times 2$, ο οποίος περιλαμβάνει τις συντεταγμένες όλων των κορυφών που θα χρησιμοποιηθούν για τον σχηματισμό όλων των τριγώνων.
- τον πίνακα `vcolors`, μεγέθους $L \times 3$, ο οποίος περιλαμβάνει τα χρώματα όλων των κορυφών που θα χρησιμοποιηθούν για τον σχηματισμό όλων των τριγώνων σε RGB τιμές.
- τον πίνακα `depth`, μεγέθους $L \times 1$, ο οποίος περιλαμβάνει τα βάθη όλων των κορυφών που θα χρησιμοποιηθούν για τον σχηματισμό όλων των τριγώνων.
- την μεταβλητή `shading`, τύπου `string`, η οποία παίρνει τις τιμές "f" ή "g" ανάλογα με το ποια μέθοδο πλήρωσης τριγώνων θέλω να χρησιμοποιήσω.

Ως έξοδο επιστρέφει την εικόνα, οπότε ένας τρόπος κλήσης της είναι:

```
image = functions.render_img(faces, vertices, vcolors, depth, shading)
```

Κεφάλαιο 6

Παραδοχές

1. **Το τρίγωνο είναι ένα σημείο:** Όταν οι συντεταγμένες και των τριών κορυφών ταυτίζονται θεωρούμε ότι το τρίγωνο είναι ένα σημείο. Στην συνάρτηση `flat_shading` δεν τίθεται θέμα το τι χρώμα θα χρωματιστεί το συγκεκριμένο `pixel`, καθώς εξ' ορισμού θα πάρει το χρώμα με το μέσο όρο των χρωμάτων των τριών κορυφών. Στην συνάρτηση `gouraud_shading` θεωρούμε ότι και σε αυτήν την περίπτωση το `pixel` θα πάρει το χρώμα με το μέσο όρο των χρωμάτων των τριών κορυφών, καθώς και οι τρεις κορυφές του ταυτίζονται.
2. **Το τρίγωνο είναι μια οριζόντια γραμμή:** Όταν δηλαδή οι τεταγμένες των τριών κορυφών ταυτίζονται. Παρομοίως, αυτή η περίπτωση δεν χρειάζεται ιδιαίτερη αντιμετώπιση στην συνάρτηση `flat_shading`, καθώς τα `pixel` έχουν ίδιο χρώμα, αρκεί να την λάβουμε υπόψην μας.

Στην συνάρτηση `gouraud_shading` όμως για αυτήν την περίπτωση έχουμε τρεις διαφορετικές υποπεριπτώσεις:

- **Δύο κορυφές ταυτίζονται στην αρχή του ευθυγράμμου τμήματος.** Για να χειριστούμε αυτήν την περίπτωση βρίσκουμε το μέσο χρώμα των δύο κορυφών που ταυτίζονται και έπειτα με το νέο αυτό χρώμα ως την αρχή και το χρώμα της άλλης κορυφής ως το τέλος του ευθύγραμμου τμήματος χρωματίζουμε με γραμμική παρεμβολή τα υπόλοιπα `pixel`.
 - **Δύο κορυφές ταυτίζονται στο τέλος του ευθυγράμμου τμήματος.** Δουλεύουμε με αντίστοιχο τρόπο με την πάνω υποπερίπτωση.
 - **Οι τρεις κορυφές βρίσκονται σε διαφορετικά σημεία.** Σε αυτήν την περίπτωση χρωματίζουμε με γραμμική παρεμβολή χρώματος τα δύο υποδιαστήματα που προκύπτουν.
3. **Το τρίγωνο έχει μια οριζόντια πλευρά:** Όταν η πλευρά αυτή είναι στο πρώτο `scan line` δεν χρειάζεται ειδική διαχείριση καθώς εμπίπτει στην γενική περίπτωση. Σε αυτήν την περίπτωση θεωρούμε τις ενεργές πλευρές ως τις άλλες δύο πλευρές του τριγώνου, επομένως και σε αυτήν την περίπτωση τα ενεργά οριακά σημεία είναι δύο. Στην περίπτωση που η οριζόντια πλευρά είναι στο τελευταίο `scan line` λειτουργούμε με παρόμοιο τρόπο αλλά χειριζόμαστε την περίπτωση αυτή σε ξεχωριστό μπλοκ κώδικα, καθώς όπως αναφέρεται και στα σχόλια με τους ελέγχους που χρησιμοποιούνται το τελευταίο `scan line` δεν ενσωματώνεται στην γενική περίπτωση.
 4. **Το τρίγωνο έχει μια κάθετη πλευρά:** Σε αυτήν την περίπτωση λαμβάνεται υπόψη ότι η κλίση της ευθείας είναι άπειρη και χειρίζεται κατάλληλα με ελέγχους.
 5. **Το τρίγωνο είναι μια κάθετη πλευρά:** Παρομοίως με την οριζόντια πλευρά υπάρχουν οι υποπεριπτώσεις που αναφέραμε αλλά λόγω της φύσης του αλγορίθμου ξετάζει τα `pixel` της πλευράς αυτής σε ξεχωριστό `scan line`. Αρκεί να λάβουμε υπόψην μας τις περιπτώσεις που δύο κορυφές ταυτίζονται.
 6. **Το τρίγωνο δεν εμπίπτει σε καμία από τις παραπάνω κατηγορίες:** Τότε ακολουθείται ο αλγόριθμος για τις γενικές περιπτώσεις που έχει καταγραφεί παραπάνω.

Κεφάλαιο 7

Demos

Και τα δύο demos έχουν ίδια δομή και δεν απαιτούνται εξωτερικά ορίσματα για να τρέξουν. Αρχικά καθορίζεται η μεταβλητή shading στο ένα είναι f και στο άλλο g. Έπειτα φορτώνονται τα δεδομένα από το αρχείο hw1.npy και αφότου αντιστοιχηθούν με τους κατάλληλους πίνακες-ορίσματα της συνάρτησης render_img καλούμε την συνάρτηση αυτή. Θεωρούμε ότι τα χρώματα των κορυφών από το αρχείο που μας δίνεται βρίσκεται σε μορφή RGB. Η βιβλιοθήκη opencv, η οποία χρησιμοποιείται για να αποθηκευτεί η εικόνα που παράγεται από την συνάρτηση render_img, έχει κωδικοποίηση για τα χρώματα BGR, ενώ ο πίνακας image που έχει παραχθεί περιέχει τα χρώματα σε μορφή RGB. Οπότε είναι αναγκαίο προτού αποθηκεύσουμε την εικόνα να κάνουμε την μετατροπή από RGB σε BGR. Τέλος, προβάλλεται η εικόνα με την βοήθεια της βιβλιοθήκης matplotlib.

7.1 Flat Shading

Με την χρήση της μεθόδου Flat Shading παρατηρούμε ότι τα τρίγωνα που συνθέτουν την εικόνα είναι εμφανή.



Σχήμα 7.1: Αναπαράσταση εικόνας με Flat Shading μέθοδο.

7.2 Gouraud Shading

Με την χρήση της μεθόδου Gouraud Shading παρατηρούμε ότι τα τρίγωνα που συνθέτουν την εικόνα είναι χρωματισμένα με ομαλό τρόπο, με αποτέλεσμα τα τρίγωνα να μην είναι εμφανή και να παρατη-

ρούμε περισσότερες λεπτομέρειες στην εικόνα.



Σχήμα 7.2: Αναπαράσταση εικόνας με Gouraud Shading μέθοδο.