

Ατομική Διπλωματική Εργασία

**ΔΗΜΙΟΥΡΓΙΑ ΤΡΙΣΔΙΑΣΤΑΤΟΥ ΜΟΝΤΕΛΟΥ
ΚΤΗΡΙΟΥ ΑΠΟ ΦΩΤΟΓΡΑΦΙΑ, ΜΕ ΧΡΗΣΗ
ΓΕΝΕΤΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ**

Σάββα Ιωάννα

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ιούνιος 2006

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Δημιουργία τρισδιάστατου μοντέλου κτηρίου από
φωτογραφία, με χρήση γενετικού προγραμματισμού

Σάββα Ιωάννα

Επιβλέπων Καθηγητής

Γιώργος Χρυσάνθου

Η Ατομική αυτή Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής
του Πανεπιστημίου Κύπρου

Ιούνιος 2006

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέπων καθηγητή μου για αυτή την εργασία, Επίκουρο καθηγητή Δρ. Γιώργο Χρυσάνθου, για το χρήσιμο συμβουλευτικό του έργο.

Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου για την ηθική και υλική υποστήριξη που μου πρόσφερε κατά την διάρκεια της φοίτησης μου, αλλά και κατά την εκπόνηση αυτής της διπλωματικής εργασίας.

Περίληψη

Η δημιουργία τρισδιάστατων μοντέλων από αντικείμενα στον πραγματικό κόσμο για χρήση σε διάφορες εφαρμογές, είναι πολλές φορές χρονοβόρα διαδικασία. Υπάρχουν όμως συσκευές που μπορούν να βοηθήσουν αυτό, όπως τρισδιάστατοι σαρωτές. Η δημιουργία μοντέλων πραγματικών κτηρίων όμως είναι ακόμα πιο δύσκολη.

Η παρούσα Ατομική Διπλωματική Εργασία ασχολείται με την ανάπτυξη ενός προγράμματος για δημιουργία τρισδιάστατων μοντέλων κτηρίων, με είσοδο κάποιες φωτογραφίες τους. Βασικό μέρος της έρευνας αυτής αποτελεί η χρήση γενετικού αλγόριθμου και μιας βιβλιοθήκης με τα αντικείμενα που εμφανίζονται στο κτήριο.

Αρχικά, ο γενετικός αλγόριθμος τοποθετεί τυχαία διάφορα αντικείμενα στην σκηνή και προσπαθεί να δημιουργήσει μια σκηνή που μοιάζει με αυτή που βρίσκεται σε μια εικόνα. Αυτή η εικόνα αποτελείται από μαύρα πρωτεύοντα σχήματα και λευκό φόντο. Η προσέγγιση αυτή βοηθάει να επιβεβαιωθεί αν ο αλγόριθμος είναι λειτουργικός όσον αφορά αυτό το πρόβλημα. Με αυτή την προσέγγιση δοκιμάστηκαν δύο μέθοδοι για αξιολόγηση των ατόμων.

Αργότερα, ο γενετικός αλγόριθμος παίρνει σαν είσοδο εικόνες από αντικείμενα σε πραγματικά κτήρια και προσπαθεί να τα δημιουργήσει. Στόχος είναι να μπορεί να αναγνωρίζει πρώτα τα αντικείμενα στο ζητούμενο κτήριο και να δημιουργεί μια βιβλιοθήκη με αυτά. Ακολούθως, να χρησιμοποιεί την βιβλιοθήκη για να δημιουργήσει το κτήριο. Σε περίπτωση που κάποια αντικείμενα στο κτήριο είναι σχετικά απλά, τότε δεν χρειάζεται να δημιουργηθούν και ακολουθούμε την ίδια προσέγγιση για αυτά στο κτήριο. Με λίγα λόγια η βιβλιοθήκη αποτελείται από τα αντικείμενα που δημιουργήθηκαν και τα πρωτεύοντα σχήματα.

Η προσέγγιση αυτή έχει προοπτικές, παρατηρώντας τα αποτελέσματα αυτής τις εργασίας, αλλά χρειάζεται ακόμη πολλή δουλειά για να ολοκληρωθεί.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1	1
ΕΙΣΑΓΩΓΗ	1
1.1 ΣΤΟΧΟΣ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	1
1.2 ΠΕΡΙΟΡΙΣΜΟΣ ΣΤΟΧΩΝ	2
1.3 ΠΕΡΙΟΧΗ ΕΦΑΡΜΟΓΗΣ	2
1.4 ΕΝΑΛΛΑΚΤΙΚΕΣ ΜΕΘΟΔΟΙ	3
1.5 ΕΠΙΣΚΟΠΗΣΗ ΑΝΑΦΟΡΑΣ	5
ΚΕΦΑΛΑΙΟ 2	7
ΓΕΝΕΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	7
2.1 ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΚΑΙ ΟΡΟΛΟΓΙΑ	7
2.2 ΓΕΝΕΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	8
2.3 ΠΕΔΙΟ ΕΡΕΥΝΑΣ	10
2.4 ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ	10
2.5 ΚΩΔΙΚΟΠΟΙΗΣΗ ΧΡΩΜΟΣΩΜΑΤΩΝ	10
2.6 ΟΙ ΑΡΧΙΚΕΣ ΔΟΜΕΣ	12
2.7 ΤΙΜΗ ΙΚΑΝΟΤΗΤΑΣ	13
2.8 ΕΠΙΛΟΓΗ	13
2.9 ΠΡΩΤΕΥΟΥΣΕΣ ΔΙΑΔΙΚΑΣΙΕΣ	16
2.10 ΔΕΥΤΕΡΕΥΟΥΣΕΣ ΔΙΑΔΙΚΑΣΙΕΣ	17
2.11 ΚΑΤΑΣΤΑΣΗ ΤΟΥ ΕΞΕΛΙΚΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ	19
2.12 ΚΡΙΤΗΡΙΟ ΤΕΡΜΑΤΙΣΜΟΥ	19
2.13 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΤΟΥ ΑΠΟΤΕΛΕΣΜΑΤΟΣ	19
2.14 ΠΑΡΑΜΕΤΡΟΙ ΕΛΕΓΧΟΥ	19
2.15 ΒΗΜΑΤΑ ΓΕΝΕΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ	20
2.16 ΓΙΑΤΙ ΓΕΝΕΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ	21
2.17 ΒΙΒΛΙΟΘΗΚΕΣ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ	23
ΚΕΦΑΛΑΙΟ 3:	26
ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΡΕΥΝΑ	26
3.1 ΣΤΟΧΟΣ ΕΡΕΥΝΑΣ	26
3.2 ΙΔΕΑ ΓΙΑ ΕΝΑΡΞΗ ΕΡΕΥΝΑΣ	26
3.3 ΔΗΜΙΟΥΡΓΙΑ ΒΙΒΛΙΟΘΗΚΗΣ	27
3.4 ΑΝΑΠΑΡΑΣΤΑΣΗ ΧΡΩΜΟΣΩΜΑΤΩΝ	29
3.5 ΦΩΤΟΓΡΑΦΙΕΣ ΚΑΙ ΤΥΠΟΙ ΚΤΗΡΙΩΝ	30
ΚΕΦΑΛΑΙΟ 4:	32
ΥΛΟΠΟΙΗΣΗ ΓΕΝΕΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ ΚΑΙ ΜΕΘΟΔΟΣ ΑΞΙΟΛΟΓΗΣΗΣ	32
4.1 ΡΟΛΟΣ GAUL	32
4.2 ΑΡΧΙΚΗ ΙΔΕΑ	35
4.3 ΑΝΑΠΑΡΑΣΤΑΣΗ ΛΥΣΗΣ	36
4.4 ΒΙΒΛΙΟΘΗΚΗ ΑΝΤΙΚΕΙΜΕΝΩΝ	37
4.5 ΣΤΑΘΕΡΟΣ ΑΡΙΘΜΟΣ ΑΝΤΙΚΕΙΜΕΝΩΝ	37
4.6 ΥΛΟΠΟΙΗΣΗ ΓΕΝΕΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ	38
4.7 ΓΡΑΦΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ	39
4.8 ΑΞΙΟΛΟΓΗΣΗ ΙΚΑΝΟΤΗΤΑΣ	40
4.9 ΠΡΟΣΟΜΟΙΩΣΕΙΣ	40
4.10 ΣΥΜΠΕΡΑΣΜΑΤΑ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	46

4.11 ΑΛΛΑΓΗ ΚΑΤΕΥΘΥΝΣΗΣ	46
ΚΕΦΑΛΑΙΟ 5:.....	48
ΒΕΛΤΙΩΜΕΝΗ ΜΕΘΟΔΟΣ ΑΞΙΟΛΟΓΗΣΗΣ	48
5.1 ΔΕΥΤΕΡΗ ΜΕΘΟΔΟΣ ΑΞΙΟΛΟΓΗΣΗΣ	48
5.2 Η ΓΛΩΣΣΑ CG ΚΑΙ ΤΑ SHADERS.....	49
5.3 ΕΠΕΚΤΑΣΕΙΣ OPENGL	50
5.4 FRAME BUFFER OBJECTS.....	50
5.5 ARB OCCLUSION QUERY	52
5.6 ΥΛΟΠΟΙΗΣΗ ΑΞΙΟΛΟΓΗΣΗΣ.....	52
5.7 ΠΡΟΣΟΜΟΙΩΣΕΙΣ	53
5.8 ΣΥΜΠΕΡΑΣΜΑΤΑ ΑΠΟΤΕΛΕΣΜΑΤΩΝ.....	57
ΚΕΦΑΛΑΙΟ 6:	59
ΕΣΤΙΑΣΗ ΣΤΗΝ ΕΥΡΕΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ	59
6.1 ΠΡΟΣΕΓΓΙΣΗ	59
6.2 ΑΛΛΑΓΕΣ ΣΤΟ ΥΦΙΣΤΑΜΕΝΟ ΠΡΟΓΡΑΜΜΑ.....	60
6.3 ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΤΡΟΠΗΣ	60
6.4 ΔΗΜΙΟΥΡΓΙΑ ΜΟΝΤΕΛΟΥ	61
6.5 ΙΔΑΝΙΚΕΣ ΦΩΤΟΓΡΑΦΙΕΣ.....	62
6.6 ΠΡΟΣΟΜΟΙΩΣΕΙΣ	62
6.7 ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΕΩΝ	68
ΚΕΦΑΛΑΙΟ 7.....	70
ΠΗΓΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ.....	70
7.1 ΠΗΓΕΣ ΠΛΗΡΟΦΟΡΙΑΣ.....	70
7.2 ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ	71
ΚΕΦΑΛΑΙΟ 8	72
ΣΥΜΠΕΡΑΣΜΑΤΑ	72
8.1 ΑΝΑΣΚΟΠΗΣΗ ΤΩΝ ΣΤΟΧΩΝ	72
8.2 ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ.....	72
8.3 ΤΕΛΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ	73
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	74
ΠΑΡΑΡΤΗΜΑ.....	76

Κεφάλαιο 1

Εισαγωγή

- 1.1 Στόχος της διπλωματικής εργασίας
 - 1.2 Περιορισμός στόχων
 - 1.3 Περιοχή εφαρμογής
 - 1.4 Εναλλακτικές μέθοδοι
 - 1.5 Επισκόπηση Αναφοράς
-

1.1 Στόχος της διπλωματικής εργασίας

Στόχος της διπλωματικής εργασίας είναι η ανάπτυξη ενός προγράμματος υπολογιστή, το οποίο θα παίρνει σαν είσοδο μια ή περισσότερες φωτογραφίες ενός κτηρίου. Με την χρήση γενετικού προγραμματισμού και των φωτογραφιών θα δημιουργεί ένα τρισδιάστατο μοντέλο του κτηρίου. Στα πλαίσια αυτής της εργασίας, θα γίνει μελέτη και κατανόηση της θεωρίας του γενετικού προγραμματισμού και πόσο ικανή μέθοδος είναι για επίλυση προβλημάτων. Ακολούθως θα μελετηθεί με πιο τρόπο και με ποιες παραμέτρους μπορεί να εφαρμοστεί ένας γενετικός αλγόριθμος, για την επίλυση ενός τέτοιου προβλήματος. Επίσης θα γίνει έρευνα για το είδος των κτηρίων για τα οποία θα μπορεί να δημιουργηθεί μια τρισδιάστατη αναπαράσταση με αυτή την μέθοδο, αλλά επίσης και για το είδος των κτηρίων με τα οποία θα ασχοληθεί αυτή η εργασία. Ακολούθως θα καθοριστεί το είδος των φωτογραφιών οι οποίες θα είναι χρήσιμες για να χρησιμοποιηθούν. Αφού γίνει η υλοποίηση του προγράμματος θα διεξαχθούν πειράματα για μέτρηση της αποδοτικότητας του προγράμματος. Από

αυτές τις μετρήσεις θα αποφασιστεί κατά πόσο η εκπόνηση αυτής της εργασίας ήταν επιτυχής, αλλά και κατά πόσο θα μπορεί να επεκταθεί στο μέλλον.

1.2 Περιορισμός στόχων

Η περιοχή που εξετάζει αυτή η εργασία είναι τεράστια. Οι πιθανοί τύποι αρχιτεκτονικής κτηρίων και η ποικιλία φωτογραφιών οι οποίες μπορούν να παρθούν για ένα κτήριο είναι άπειρη. Επίσης η δημιουργία ενός τρισδιάστατου κτηρίου με την χειρονακτική μέθοδο, χρειάζεται χρόνο, εμπειρία και πολλές φορές προηγμένες τεχνικές για να αποδώσει το επιθυμητό αποτέλεσμα και να είναι ρεαλιστική. Για τους λόγους αυτούς δεν είναι ρεαλιστικό να ελπίζει κανείς ότι μπορούν να καλυφθούν όλα όσα θα μπορούσαν σε αυτή την εργασία.

Για να περιοριστεί η εργασία σε κάτι το οποίο να είναι προσιτό, η προσοχή θα δοθεί σε ένα υποσύνολο του τι είναι δυνατό να επιτευχθεί, που θα είναι οι κύριοι στόχοι της εργασίας.

Οι κύριοι στόχοι είναι:

- Μελέτη γενετικών αλγορίθμων
- Περιορισμός των τύπων κτηρίων με τα οποία θα ασχοληθεί η εργασία.
- Περιορισμός των τύπων φωτογραφιών με τα οποία θα ασχοληθεί η εργασία.
- Υλοποίηση γενετικού αλγορίθμου που θα βοηθά στην δημιουργία/εύρεση του κτηρίου.
- Υλοποίηση μεθόδου αξιολόγησης/επιβεβαίωσης της ορθότητας του κτηρίου.
- Δημιουργίας τρισδιάστατης πλευράς κτηρίου με χρήση μιας φωτογραφίας.

1.3 Περιοχή εφαρμογής

Όπως έχει αναφερθεί και πριν η δημιουργία ενός τρισδιάστατου κτηρίου επηρεάζεται από πολλές παραμέτρους για να δώσει ένα ρεαλιστικό αποτέλεσμα. Ήδη υπάρχουν οι τρισδιάστατοι σαρωτές, οι οποίοι εν μέρει λύνουν το πρόβλημα της μεταφοράς ενός πραγματικού αντικειμένου μέσα στον υπολογιστή. Στο τομέα

των κτηρίων υπάρχουν διαθέσιμα κάποια προγράμματα, τα οποία όμως απευθύνονται σε πιο έμπειρους χρήστες. Η ανάπτυξη αυτού του προγράμματος, α μπορούσε να προσφέρει μια δυνατότητα μοντελοποίησης κτηρίων χωρίς την απαίτηση εμπειρίας από τον χρήστη. Θα μπορεί να προσφέρει την δυνατότητα να υπάρχει διαθέσιμος ένας εναλλακτικός και ίσως ποιο εύκολος τρόπος δημιουργίας τρισδιάστατων κτηρίων, τα οποία θα μπορούν είτε και όπως είναι, δηλαδή όπως τα παράγει το πρόγραμμα, είτε με κάποιες μετατροπές να χρησιμοποιηθούν σε εφαρμογές. Επίσης θα μπορούσε να επεκταθεί και σε πρόγραμμα δημιουργίας τρισδιάστατων σκηνών.

1.4 Εναλλακτικές μέθοδοι

Υπάρχουν κι άλλες προτεινόμενες μέθοδοι για δημιουργία ενός τρισδιάστατου μοντέλου ενός πραγματικού κτηρίου. Οι μέθοδοι στις οποίες θα αναφερθώ χρησιμοποιούν, όπως και η έρευνα την οποία παρουσιάζω εδώ, φωτογραφίες ενός πραγματικού κτηρίου για να δημιουργήσουν το μοντέλο του.

Photo3D

Το Photo3D είναι ένα λογισμικό το οποίο που δημιουργεί τρισδιάστατα μοντέλα από φωτογραφίες. Ο χρήστης του μπορεί εύκολα να δημιουργήσει μια συλλογή από μοντέλα και συστάσεις (texture) από τον πραγματικό κόσμο. Το λογισμικό αυτό, δίνει την δυνατότητα δημιουργίας ενός τρισδιάστατου μοντέλου με μόνο μια φωτογραφία και με την χρήση περισσότερων φωτογραφιών δημιουργεί πιο πολύπλοκα μοντέλα. Το μοντέλο που δημιουργείται εξάγεται σε VRML. Το προϊόν αυτό δημιουργήθηκε από την εταιρία Softcube Co., Ltd., η οποία ειδικεύεται σε τρισδιάστατα γραφικά για SGI και Windows.

Παράδειγμα δημιουργίας μοντέλου με χρήση μιας φωτογραφίας



Φωτογραφία



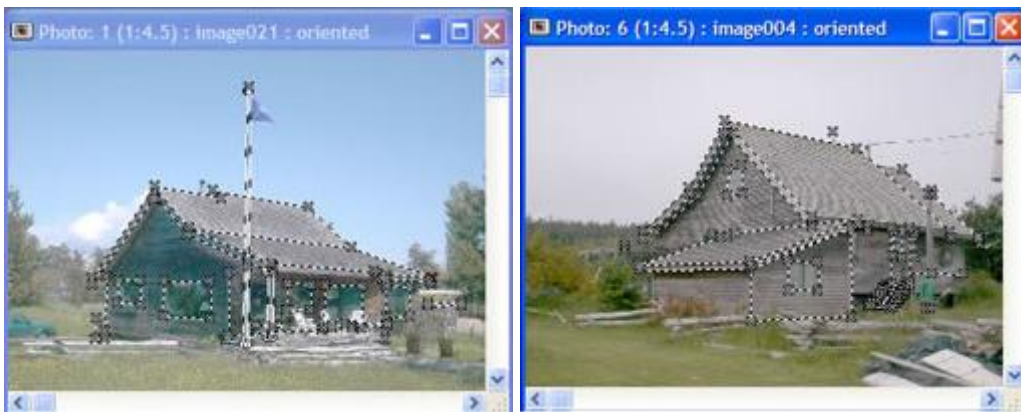
3D μοντέλο

PhotoModeler

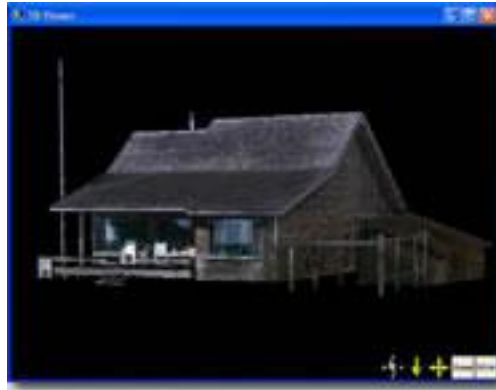
Το photoModeler είναι ένα λογισμικό για Windows, κατάλληλο για υπολογισμούς μετρήσεων και κατασκευή τρισδιάστατων μοντέλων πραγματικών αντικειμένων και σκηνών με την χρήση φωτογραφιών. Για την δημιουργία του μοντέλου, ο χρήστης εισάγει στο πρόγραμμα κάποιες φωτογραφίες (δύο ή περισσότερες) και μετά επιλέγει/σημειώνει κάποια χαρακτηριστικά γνωρίσματά τους. Μετά από αυτή την διαδικασία, το photoModeler είναι έτοιμο να επεξεργαστεί τα δεδομένα και να δημιουργήσει δεδομένα που αντιπροσωπεύουν ένα τρισδιάστατο μοντέλο. Μπορεί να δημιουργήσει μοντέλα για διάφορα αντικείμενα και όχι μόνο κτήρια.

Αναπτύχθηκε από την εταιρεία Eos Systems Inc.

Παράδειγμα δημιουργίας μοντέλου με χρήση φωτογραφιών



Κάποιες φωτογραφίες



Τελικό μοντέλο

1.5 Επισκόπηση Αναφοράς

Σε αυτό το σημείο της εισαγωγής θα δοθεί μια μικρή περιγραφή του τι θα ακολουθήσει στα επόμενα κεφάλαια αυτής της αναφοράς.

Κεφάλαιο 2 : Γενετικός Προγραμματισμός

Σε αυτό το κεφάλαιο θα αναφέρω στις βασικές αρχές του γενετικού προγραμματισμού και των γενετικών αλγορίθμων, καθώς και άλλες έννοιες τις οποίες πρέπει να γνωρίζει ο αναγνώστης πριν προχωρήσει στην ανάγνωση της υπόλοιπης εργασίας. Επίσης θα αναφέρω τα βήματα που χρειάζεται η εκτέλεση ενός γενετικού αλγόριθμου. Τέλος, θα συζητήσω γιατί κάποιος να θέλει να χρησιμοποιήσει γενετικούς αλγόριθμους για επίλυση κάποιου προβλήματος και για κάποιες βιβλιοθήκες γενετικών αλγορίθμων που υπάρχουν διαθέσιμες και μπορεί να χρησιμοποιήσει.

Κεφάλαιο 3 : Έναρξη έρευνας

Σε αυτό το κεφάλαιο θα αναφέρω ποια ήταν η ιδέα η οποία οδήγησε σε αυτή την έρευνα, πέρα από τα όρια αυτής της διπλωματικής εργασίας, και ποιες είναι οι ιδέες για υλοποίηση της.

Κεφάλαιο 4 : Υλοποίηση γενετικού αλγόριθμου και μέθοδος αξιολόγησης

Εδώ θα συζητήσω ποια είναι η αρχική ιδέα υλοποίησης του γενετικού αλγόριθμου και δείξω πως θα κάνω την γραφική αναπαράσταση του αποτελέσματος. Επίσης θα αναφέρω με ποιο τρόπο θα αξιολογώ τις λύσεις που μου δίνει ο γενετικός αλγόριθμος. Θα αναφερθώ στα πειράματα που έκανα και τις μετρήσεις, αλλά και στους λόγους που με οδήγησαν στην αλλαγή της μεθόδου αξιολόγησης των λύσεων.

Κεφάλαιο 5 : Βελτιωμένη μέθοδος αξιολόγησης

Σε αυτό το κεφάλαιο θα μιλήσω για την νέα μέθοδο αξιολόγησης των λύσεων του γενετικού αλγόριθμου, αλλά και σε κάποια θεωρία την οποία έπρεπε να μελετήσω να για να μπορέσω να υλοποιήσω προγραμματιστικά αυτή την μέθοδο. Επίσης θα παραθέσω τα πειράματα τα οποία έκανα και τα αποτελέσματα τους.

Κεφάλαιο 6 : Προσέγγιση έγχρωμης φωτογραφίας

Σε αυτό το σημείο θα αναφέρω πώς προχώρησα σε χρήση έγχρωμων φωτογραφιών και ποια είναι η τελική προσέγγιση του προβλήματος. Θα παρουσιάσω τα τελικά μου πειράματα και αποτελέσματα, και θα αναφερθώ σε ποιου τύπου φωτογραφίες θα δουλεύει το τελικό πρόγραμμα.

Κεφάλαιο 7 : Πηγές και εργαλεία

Εδώ θα αναφέρω τις πηγές τις οποίες μελέτησα για να προχωρήσω στην επίλυση αυτού του προβλήματος, αλλά και τα εργαλεία τα οποία χρησιμοποίησα για την υλοποίηση του συστήματος.

Κεφάλαιο 8 : Συμπεράσματα

Στο τελευταίο κεφάλαιο της αναφοράς θα αξιολογήσω την εργασία αυτή με βάση τους στόχους που έθεσα στην αρχή και αν το σύστημα που παράχθηκε είναι αυτό που αναμενόταν.

Κεφάλαιο 2

Γενετικός Προγραμματισμός

- 2.1 Βασικές αρχές και ορολογία
 - 2.2 Γενετικός Προγραμματισμός
 - 2.3 Πεδίο έρευνας
 - 2.4 Γενετικοί Αλγόριθμοι
 - 2.5 Κωδικοποίηση χρωμοσωμάτων
 - 2.6 Οι αρχικές δομές
 - 2.7 Τιμή ικανότητας
 - 2.8 Επιλογή
 - 2.9 Πρωτεύουσες διαδικασίες
 - 2.10 Δευτερεύουσες διαδικασίες
 - 2.11 Κατάσταση του εξελικτικού συστήματος
 - 2.12 Κριτήριο τερματισμού
 - 2.13 Προσδιορισμός του αποτελέσματος
 - 2.14 Παράμετροι ελέγχου
 - 2.15 Βήματα Γενετικού Αλγορίθμου
 - 2.16 Γιατί Γενετικός Αλγόριθμος
 - 2.17 Βιβλιοθήκες Γενετικών Αλγορίθμων
-

2.1 Βασικές αρχές και ορολογία

Στη φύση όλοι οι ζωντανοί οργανισμοί περιέχουν ένα σύνολο γενετικών δεδομένων, που ονομάζεται «γονιδίωμα». Με κάποιο τρόπο, αυτά τα γενετικά δεδομένα κωδικοποιούν όλα τα φυσικά χαρακτηριστικά του οργανισμού. Ένα

συγκεκριμένο σύνολο γενετικής πληροφορίας είναι ένας «γενότυπος» και ένα συγκεκριμένο σύνολο φυσικών χαρακτηριστικών ή αλλιώς γνωρίσματα, είναι ένας «φαινότυπος». Μπορεί να υπάρξει μια άμεση ένα προς ένα χαρτογράφηση των γενοτύπων στους φαινοτύπους. Αυτά τα φυσικά χαρακτηριστικά καθορίζουν πόσο ταιριαστός είναι ένας οργανισμός στο περιβάλλον που βρίσκεται. Η καταλληλότητα ενός δεδομένου οργανισμού στο περιβάλλον του μετριέται συνήθως ως «τιμή ικανότητας». Υπολογιστικά, είναι συνηθισμένο να αξιολογείται η ικανότητα ενός οργανισμού κατευθείαν, χωρίς να λαμβάνεται υπόψη κάποιο είδος φυσικής εμφάνισης.

Η ιδέα «Επιβίωση του ικανότερου», όπως εισάγεται από τον Charles Darwin, είναι ευρέως γνωστή. Ενώ στη φύση η "ικανότητα" σχετίζεται με τη δυνατότητα του οργανισμού να επιζήσει και να αναπαραγάγει, στους γενετικούς αλγορίθμους η "ικανότητα" είναι το αξιολογημένο αποτέλεσμα κάποιας "αντικειμενικής λειτουργίας". Οι οργανισμοί με καλύτερη «ικανότητα» είναι πιο πιθανό να επιλεγούν για αναπαραγωγή είτε μέσω κάποιου μηχανισμού ανταγωνισμού για να ζευγαρώσουν, είτε ως αποτέλεσμα του να χαθούν οι λιγότερο κατάλληλοι οργανισμοί. Με αυτό τον τρόπο τα γονίδια που κωδικοποιούν τα ευεργετικά χαρακτηριστικά, διαδίδονται μέσω των επόμενων γενεών του πληθυσμού, εις βάρος των γονιδίων που κωδικοποιούν τα καταστρεπτικά χαρακτηριστικά.

Το γονιδίωμα στη φύση αποθηκεύεται υπό μορφή του DNA. Στους ηλεκτρονικούς υπολογιστές υπάρχει περισσότερη ευελιξία, αλλά ο κύριος παράγοντας που ελέγχει εάν ένα συγκεκριμένο πρόβλημα αναζήτησης επιλύεται επιτυχώς, ή όχι, εξαρτάται από το πώς το πρόβλημα κωδικοποιήθηκε στο γονιδίωμα. Σήμερα χρησιμοποιούνται κοινώς πολλές διαφορετικές μέθοδοι κωδικοποίησης των γενετικών πληροφοριών, στις οποίες θα αναφερθώ αργότερα.

2.2 Γενετικός Προγραμματισμός

Μια από τις κύριες προκλήσεις της επιστήμης πληροφορικής είναι η δημιουργία ενός προγράμματος, που να κάνει αυτό που χρειάζεται να γίνει χωρίς να χρειαστεί να του πεις πώς να το κάνει. Ο γενετικός προγραμματισμός αντιμετωπίζει αυτή την

πρόκληση, προσφέροντας μια αυτοματοποιημένη μεθοδολογία εμπνευσμένη από την φυσική επιλογή του Δαρβίνου (Darwinian natural selection) και άλλες διαδικασίες εμπνευσμένες από την βιολογία, για εύρεση υπολογιστικών προγραμμάτων από μια υψηλού επιπέδου δήλωση του προβλήματος. Οι διαδικασίες αυτές περιλαμβάνουν την αναπαραγωγή, την διασταύρωση, την μετάλλαξη, διπλασιασμό γονιδίων και διαγραφή γονιδίων. Ο γενετικός προγραμματισμός μπορεί να ψάξει στον χώρο των πιθανών προγραμμάτων για ένα συγκεκριμένο, το οποίο είναι πολύ κατάλληλο στο να λύσει ένα δεδομένο πρόβλημα.

Το γενικό μήνυμα της επιβίωσης του ικανότερου και φυσικής επιλογής που περιγράφει ο Charles Darwin είναι: Οντότητες που είναι πιο ικανές να εκτελούν εργασίες στο περιβάλλον τους, επιβιώνουν και αναπαράγονται με υψηλότερο ρυθμό. Λιγότερο ικανές οντότητες επιβιώνουν και αναπαράγονται με χαμηλότερο ρυθμό.

Η γενική ιδέα είναι απλή. Αρχικά παράγεται ένας πληθυσμός εκατοντάδων ή χιλιάδων τυχαία κατασκευασμένων προγραμμάτων υπολογιστή με διάφορα τυχαία αποφασισμένα μεγέθη και σχήματα. Μετά την αρχικοποίηση επιλέγονται οι γονείς σύμφωνα με μια συνάρτηση πιθανότητας η οποία βασίζεται στην ικανότητα των ατόμων του πληθυσμού. Όσο καλύτερη είναι η ικανότητα ενός ατόμου, τόσο αυξάνονται οι πιθανότητες να επιλεγεί περισσότερες φορές σαν γονέας για την αναπαραγωγή απογόνων. Τότε αναπαράγει γενετικά τον πληθυσμό χρησιμοποιώντας την μέθοδο, επιβίωσης και αναπαραγωγής των καταλληλότερων, του Δαρβίνου. Μετά από περίοδο πολλών γενεών, αναπαράγουμε πληθυσμούς που είναι ακόμα πιο κατάλληλοι για την επίλυση του προβλήματος.

Τα άτομα που προκύπτουν στον ενδιάμεσο πληθυσμό από το πρώτο βήμα αντικαθιστούν όλα ή ένα μέρος των ατόμων στον αρχικό πληθυσμό. Υπάρχουν δυο μοντέλα αντικατάστασης, το μοντέλο ολικής γενεαλογικής αντικατάστασης (generational replacement model) και σταθερής αντικατάστασης (state steady replacement model). Στο μοντέλο ολικής αντικατάστασης, ο ενδιάμεσος πληθυσμός έχει το ίδιο μέγεθος με τον αρχικό πληθυσμό, που ανανεώνει ολόκληρο τον πληθυσμό σε μια γενεά ενώ στο μοντέλο σταθερής αντικατάστασης ο ενδιάμεσος

πληθυσμός έχει πολύ μικρότερο μέγεθος από τον αρχικό πληθυσμό. Στο μοντέλο σταθερής αντικατάστασης, ο απόγονος δεν αντικαθιστά απαραίτητα τους γονείς του αλλά μπορεί να πάρει τη θέση οποιονδήποτε άλλων ατόμων. Η εκτέλεση ολοκληρώνει μετά από ένα προκαθορισμένο αριθμό γενεών.

2.3 Πεδίο έρευνας

Όταν θέλουμε να λύσουμε ένα πρόβλημα, συνήθως ψάχνουμε να βρούμε μια λύση η οποία θα είναι η καλύτερη ανάμεσα σε άλλες. Το πεδίο που περιλαμβάνει όλες τις εφικτές λύσεις, στο οποίο περιλαμβάνεται και η σωστή λύση, ονομάζεται πεδίο έρευνας. Κάθε σημείο στο πεδίο αυτό, αποτελεί μια πιθανή λύση. Κάθε πιθανή λύση μπορεί να σημαδευτεί με την τιμή ικανότητας της για το πρόβλημα που διερευνάται. Με τον γενετικό αλγόριθμο, ψάχνουμε για την καλύτερη λύση μέσα από ένα αριθμό πιθανών λύσεων, όπου η κάθε μια αντιπροσωπεύεται από ένα σημείο στο πεδίο έρευνας.

2.4 Γενετικοί Αλγόριθμοι

Ένας γενετικός αλγόριθμος είναι μια τεχνική έρευνας που χρησιμοποιείται στην επιστήμη της πληροφορικής για να βρει κατά προσέγγιση λύσεις σε προβλήματα βελτιστοποίησης και έρευνας. Οι γενετικοί αλγόριθμοι αποτελούν κλάση των εξελικτικών αλγόριθμων. Ένας γενετικός αλγόριθμος είναι ένας εξαιρετικά παράλληλος μαθηματικός αλγόριθμος, που μετατρέπει ένα σύνολο(πληθυσμό) από ατομικά μαθηματικά αντικείμενα, το καθένα συσχετισμένο με μια τιμή ικανότητας, σε ένα νέο πληθυσμό(την επόμενη γενιά.)

2.5 Κωδικοποίηση χρωμοσωμάτων

Η κωδικοποίηση των ατόμων που αποτελούν τον πληθυσμό και αντιπροσωπεύουν της πιθανές λύσεις, είναι το πρώτο πρόβλημα που πρέπει να αντιμετωπιστεί σε προσπάθειες επίλυσης προβλημάτων με χρήση γενετικού αλγόριθμου. Οι κωδικοποιήσεις που ακολουθούν έχουν ήδη χρησιμοποιηθεί με κάποια επιτυχία.

2.5.1 Δυαδική κωδικοποίηση (binary encoding)

Η δυαδική κωδικοποίηση είναι η πιο κοινή, γιατί είναι η πρώτη που χρησιμοποιήθηκε και λόγω της σχετικής της απλότητας. Σε αυτή την κωδικοποίηση κάθε χρωμόσωμα είναι μια σειρά των δυαδικών χαρακτήρων 0 ή 1.

Χρωμόσωμα Α	101100101100101011100101
Χρωμόσωμα Β	111111100000110000011111

Παράδειγμα χρωμοσωμάτων με δυαδική κωδικοποίηση

Η δυαδική κωδικοποίηση δίνει πολλά πιθανά χρωμοσώματα ακόμα και με μικρό αριθμό τιμών. Όμως αυτή η κωδικοποίηση δεν έχει νόημα για πολλά προβλήματα.

2.5.2 Κωδικοποίηση μεταλλαγής (permutation encoding)

Κωδικοποίηση μεταλλαγής μπορεί να χρησιμοποιηθεί σε προβλήματα καθορισμού σειράς, όπως το πρόβλημα του περιοδεύον πωλητή ή προβλήματα καθορισμού σειράς στόχων. Στη κωδικοποίηση μεταλλαγής κάθε χρωμόσωμα είναι μια σειρά από αριθμούς, που αντιπροσωπεύουν μια θέση σε μια ακολουθία.

Χρωμόσωμα Α	1 5 3 2 6 4 7 9 8
Χρωμόσωμα Β	8 5 6 7 2 3 1 4 9

Παράδειγμα χρωμοσωμάτων με κωδικοποίηση μεταλλαγής

2.5.3 Κωδικοποίηση τιμής (value encoding)

Άμεση κωδικοποίηση τιμής μπορεί να χρησιμοποιηθεί σε προβλήματα όπου χρησιμοποιούνται κάποιες πιο πολύπλοκες τιμές, όπως είναι οι πραγματικοί αριθμοί. Σε αυτή την κωδικοποίηση κάθε χρωμόσωμα είναι μια ακολουθία από τιμές. Οι τιμές αυτές μπορούν να είναι οτιδήποτε συνδέεται με το πρόβλημα, όπως πραγματικοί αριθμοί, αλφαριθμητικοί χαρακτήρες ή οποιαδήποτε αντικείμενα.

Χρωμόσωμα Α	1.2324 5.3243 0.4556 2.3293 2.4545
Χρωμόσωμα Β	ABDJEIFJDHDIERJFDLDFLFEGT
Χρωμόσωμα Γ	(back), (back), (right), (forward), (left)

Παράδειγμα χρωμοσωμάτων με κωδικοποίηση τιμής

Η κωδικοποίηση αυτή είναι μια καλή επιλογή για ειδικά προβλήματα. Είναι συνήθως απαραίτητο να δημιουργηθούν νέες λειτουργίες διασταύρωσης και μετάλλαξης συγκεκριμένες για το πρόβλημα που προσπαθεί να λύσει.

2.5.4 Κωδικοποίηση δέντρου

Η κωδικοποίηση δέντρου χρησιμοποιείται κυρίως για εξέλιξη προγραμμάτων ή εκφράσεων, όπως για παράδειγμα ο γενετικός προγραμματισμός. Στην κωδικοποίηση δέντρου κάθε χρωμόσωμα είναι ένα δέντρο από αντικείμενα, όπως οι συναρτήσεις σε ένα πρόγραμμα.

Χρωμόσωμα Α	Χρωμόσωμα Β
+	do until
* /	step wall
5 y	
(+ x (/ 5 y))	(do_until step wall)

Παράδειγμα χρωμοσωμάτων με κωδικοποίηση δέντρου

2.6 Οι αρχικές δομές

Οι αρχικές δομές αποτελούνται από τον αρχικό πληθυσμό των ατομικών εκφράσεων του προβλήματος. Για επιλογή κάθε έκφρασης, γίνεται βασικά τυχαία δημιουργία ενός ατόμου.

Στην κωδικοποίηση δέντρου γίνεται τυχαία δημιουργία ενός ριζωμένου δέντρου, με ονομαζόμενους κόμβους και διατεταγμένα κλαδιά. Ξεκινούμε επιλέγοντας μια συνάρτηση, από το σύνολο των συναρτήσεων, για όνομα της ρίζας του δέντρου. Για κάθε σημείο του δέντρου που έχει για όνομα κάποια συνάρτηση, δημιουργούνται τόσες συνδέσεις όσες οι παράμετροι που δέχεται η συνάρτηση. Για κάθε τέτοια σύνδεση, ένα στοιχείο από το σύνολο $C = F \cup T$, όπου F το σύνολο των συναρτήσεων και T το σύνολο των τερματικών, επιλέγεται τυχαία να είναι το όνομα για το στοιχείο στο οποίο οδηγεί η σύνδεση. Στις άλλες κωδικοποιήσεις η δημιουργία των αρχικών ατόμων γίνεται με bit randomizer.

Στον αρχικό πληθυσμό συνίσταται να μην υπάρχουν αντίγραφα, καθώς σπαταλούν υπολογιστικούς πόρους και μειώνουν αθέμιτα την γενετική ποικιλία του πληθυσμού. Όταν τα δέντρα είναι μικρά, τότε υπάρχει μεγάλη πιθανότητα δημιουργίας αντιγράφων. Για αυτό τον λόγο κάθε έκφραση ελέγχεται αν είναι μοναδική πριν να εισαχθεί στον πληθυσμό. Στην περίπτωση των σειρών, όταν το διάστημα που ερευνάται είναι αρκετά μεγάλο, είναι εξαιρετικά δύσκολο να εμφανιστεί αντίγραφο στον αρχικό πληθυσμό.

2.7 Τιμή ικανότητας

Η μέτρηση «ικανότητας» είναι η κινητήρια δύναμη του γενετικού αλγόριθμου. Η πιο κοινή προσέγγιση για μέτρηση της ικανότητας είναι η δημιουργία ενός συγκεκριμένου μέτρου ικανότητας για κάθε άτομο στον πληθυσμό.

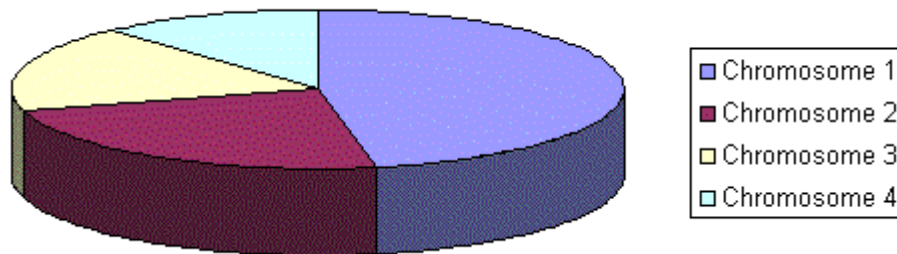
2.8 Επιλογή

Όπως είδαμε στο 2.2, χρωμοσώματα επιλέγονται από τον πληθυσμό για να γίνουν γονείς για την διασταύρωση ή κάποια άλλη λειτουργία. Το πρόβλημα που εμφανίζεται εδώ είναι πως θα γίνει αυτή η επιλογή. Σύμφωνα με την θεωρία του Darwin, τα καλύτερα επιβιώνουν για να δημιουργήσουν νέους απογόνους.

Υπάρχουν πολλές μέθοδοι για να γίνει η επιλογή και θα αναφέρω εδώ τις πιο σημαντικές από αυτές.

2.8.1 Επιλογή ρουλέτας (roulette wheel selection)

Οι γονείς επιλέγονται σύμφωνα με την ικανότητα τους. Όσο καλύτερα χρωμοσώματα είναι, τόσο περισσότερες πιθανότητες έχουν να επιλεγούν. Έχουμε για παράδειγμα μια ρουλέτα, όπου πάνω τοποθετούνται όλα τα χρωμοσώματα. Το μέγεθος των τομέων που χωρίζεται η ρουλέτα είναι ανάλογο της τιμής ικανότητας του κάθε χρωμοσώματος. Δηλαδή, όσο πιο μεγάλη είναι η τιμή τόσο πιο μεγάλος είναι ο τομέας.

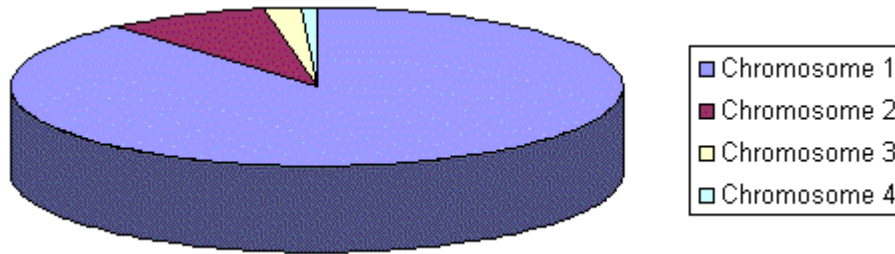


Σχήμα επιλογής ρουλέτας

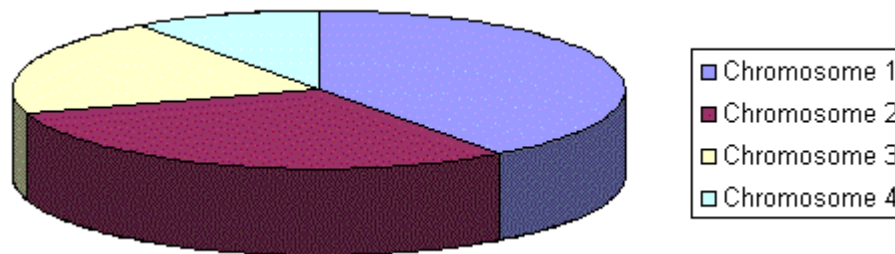
Σε αυτό το σημείο γίνεται μίμηση της ρουλέτας και σε όποιο χρωμόσωμα σταματήσει η μπίλια τότε αυτό επιλέγεται. Προφανώς τα χρωμοσώματα με μεγαλύτερες τιμές ικανότητας θα επιλεγούν περισσότερες φορές.

2.8.2 Επιλογή ταξινόμησης (rank selection)

Το προηγούμενο είδος επιλογής, θα αντιμετωπίζει προβλήματα όταν υπάρχουν μεγάλες διαφορές μεταξύ των τιμών ικανότητας των χρωμοσωμάτων. Η μέθοδος επιλογής ταξινόμησης πρώτα ταξινομεί τον πληθυσμό και μετά κάθε χρωμόσωμα λαμβάνει τιμή ικανότητας βάσει της θέσης του. Τα χειρότερα θα πάρουν τιμή ικανότητας 1, τα δεύτερα χειρότερα θα πάρουν τιμή 2 κλπ και τα καλύτερα θα έχουν τιμή ικανότητας N (όπου N είναι ο αριθμός χρωμοσωμάτων στον πληθυσμό.)



Κατάσταση πριν ταξινόμηση (γράφος ικανοτήτων)



Κατάσταση μετά την ταξινόμηση (γράφος τιμών σε σειρά)

Τώρα όλα τα χρωμοσώματα έχουν την ελπίδα να επιλεγούν. Αυτή η μέθοδος μπορεί να οδηγήσει σε πιο αργή σύγκλιση του πληθυσμού, αφού τα καλύτερα χρωμοσώματα δεν διαφέρουν τόσο πολύ από τα άλλα.

2.8.3 Επιλογή διαγωνισμού (tournament selection)

Αυτή η μέθοδος επιλογής είναι η πιο απλή. Επιλέγεται τυχαία ένας αριθμός χρωμοσωμάτων (συνήθως δύο) από τον πληθυσμό και το χρωμόσωμα με την ψηλότερη τιμή ικανότητας επιλέγεται.

2.8.3 Επιλογή σταθερής κατάστασης (steady state selection)

Αυτή η μέθοδος δεν είναι μέθοδος συγκεκριμένη για επιλογή γονέων που να εκτελούν κάποια διαδικασία. Η κύρια ιδέα αυτού του τύπου της επιλογής είναι ότι μια μεγάλη μερίδα χρωμοσωμάτων μπορούν να επιβιώσουν στην επόμενη γενιά. Η μέθοδος λειτουργεί ως εξής. Σε κάθε γενιά κάποια χρωμοσώματα με ψηλή τιμή ικανότητας επιλέγονται για να δημιουργήσουν απόγονους. Τότε κάποια χρωμοσώματα με χαμηλή τιμή ικανότητας αφαιρούνται και οι απόγονοι

τοποθετούνται στην θέση τους. Ο υπόλοιπος πληθυσμός επιβιώνει στην επόμενη γενιά.

2.8.4 Διατήρηση των ικανών (elitism)

Κατά την διάρκεια της δημιουργίας ενός νέου πληθυσμού με τις διάφορες βιολογικές διαδικασίες, υπάρχει μεγάλη πιθανότητα να χαθεί το χρωμόσωμα με την ψηλότερη τιμή ικανότητας. Αυτή η μέθοδος αντιγράφει το καλύτερο χρωμόσωμα (ή κάποια καλύτερα χρωμοσώματα) στο νέο πληθυσμό. Ο υπόλοιπος πληθυσμός επιλέγεται με τις άλλες μεθόδους.

2.9 Πρωτεύουσες διαδικασίες

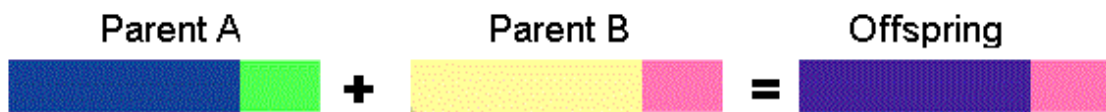
Οι πρωτεύουσες διαδικασίες για τροποποίηση των δομών είναι η αναπαραγωγή και η διασταύρωση.

Η αναπαραγωγή λειτουργεί πάνω σε ένα άτομο-έκφραση και παράγει ένα απόγονο άτομο-έκφραση με την ακόλουθη διαδικασία. Ένα άτομο-έκφραση επιλέγεται από τον πληθυσμό σε σχέση με την ικανότητα του. Το επιλεγμένο άτομο αντιγράφεται, χωρίς τροποποιήσεις, από τον τωρινό πληθυσμό στον νέο. Η επιλογή μπορεί να γίνει με τους τρόπους που αναφέρονται στο 2.7. Οι γονείς μπορούν να επιλεγθούν περισσότερες από μια φορές για αναπαραγωγή σε μια γενιά.

Η διασταύρωση δημιουργεί ποικιλία παράγοντας νέα άτομα. Παίρνει δύο γονείς και παράγει ένα απόγονο με την ακόλουθη διαδικασία. Οι δύο γονείς επιλέγονται με τον ίδιο τρόπο που επιλέξαμε τον γονέα της αναπαραγωγής.

1. Διασταύρωση ενός σημείου

Επιλέγεται τυχαία ένα σημείο σε κάθε γονέα για να είναι το σημείο διασταύρωσης, σε εκείνο το γονέα. Έτσι κάθε γονέας χωρίζεται σε δύο μέρη. Το πρώτο κομμάτι του πρώτου γονέα και το δεύτερο κομμάτι του δεύτερου γονέα αντιγράφεται και έτσι δημιουργείται ο απόγονος.



2. Διασταύρωση δύο σημείων

Επιλέγονται τυχαία δύο σημεία σε κάθε γονέα για να είναι τα σημεία διασταύρωσης, σε εκείνο το γονέα. Έτσι κάθε γονέας χωρίζεται σε τρία μέρη. Το πρώτο και τρίτο κομμάτι του πρώτου γονέα μαζί με το δεύτερο κομμάτι του δεύτερου γονέα αντιγράφεται και έτσι δημιουργείται ο απόγονος.



3. Ομοιόμορφη διασταύρωση

Επιλέγονται τυχαία κομμάτια από τους δύο γονείς και δημιουργούν τον απόγονο.



4. Αριθμητική διασταύρωση

Κάποια αριθμητική πράξη εκτελείται μεταξύ των δύο γονέων και δημιουργείται ο απόγονος.



Συνήθως οι γονείς διαφέρουν σε μέγεθος. Όταν οι δύο γονείς είναι οι ίδιοι, συνήθως οι απόγονοι είναι διαφορετικοί, γιατί επιλέγονται δύο σημεία διασταύρωσης. Αυτό δεν ισχύει στις σειρές χαρακτήρων σταθερού μήκους που επιλέγεται ένα σημείο διασταύρωσης.

2.10 Δευτερεύουσες διαδικασίες

Οι δευτερεύουσες διαδικασίες είναι η μετάλλαξη, η εναλλαγή (permutation), η σύνταξη (editing), ο ενθυλάκωση (encapsulation) και ο αποδεκατισμός (decimation).

Η μετάλλαξη εισάγει τυχαίες αλλαγές στις δομές μέσα στο πληθυσμό. Στους αλγόριθμους που δουλεύουν πάνω σε σειρές χαρακτήρων, η μετάλλαξη είναι πολύ χρήσιμη καθώς ξαναπροσφέρει ποικιλία στον πληθυσμό που κινδυνεύει συγκλίνει. Ένας πληθυσμός συγκλίνει όταν όλα του τα άτομα είναι τα ίδια. Η μετάλλαξη γίνεται ως ακολούθως. Επιλέγεται ένα τυχαίο σημείο μέσα στο άτομο-έκφραση. Μετά αφαιρείται το σημείο που επιλέχθηκε και όλα όσα βρίσκονται κάτω από αυτό (υπόδεντρο) και εισάγεται στην θέση του ένα τυχαία παραγμένο υπόδεντρο. Στις σειρές χαρακτήρων αλλάζει τυχαία ο χαρακτήρας που επιλέγηκε. Στις δυαδικές σειρές η τιμή του επιλεγμένου χαρακτήρα αντιστρέφεται.

Η εναλλαγή αλλάζει την σειρά χαρακτήρων που βρίσκονται μεταξύ δύο επιλεγμένων σημείων ενός ατόμου, μετακινώντας την σειρά όλων των χαρακτήρων μεταξύ τους. Ενεργεί πάνω σε ένα γονικό άτομο-έκφραση και παράγει ένα απόγονο.

Η σύνταξη δίνει την δυνατότητα τροποποίησης και απλοποίησης των ατόμων καθώς τρέχει ο γενετικός αλγόριθμος. Λειτουργεί πάνω σε ένα γονικό άτομο-έκφραση και παράγει ένα απόγονο. Αυτή η λειτουργία, εφαρμόζει αναδρομικά κάποιους προκαθορισμένους κανόνες σε κάθε έκφραση στον πληθυσμό.

Η ενθυλάκωση εντοπίζει αυτόματα ένα πιθανώς χρήσιμο υπόδεντρο και το βαπτίζει με ένα όνομα έτσι ώστε να μπορεί να βρεθεί και να χρησιμοποιηθεί αργότερα. Λειτουργεί πάνω σε ένα γονικό άτομο-έκφραση και παράγει ένα απόγονο και ένα νέο καθορισμό υπόδεντρου.

Αν στην επιλογή του αρχικού πληθυσμού, μεγάλο ποσοστό ατόμων έχει χαμηλή τιμή ικανότητας, υπάρχει κίνδυνος τα λίγα άτομα με κάπως καλύτερη τιμή ικανότητας να τείνουν να κυριαρχήσουν νωρίς. Παρόλο που η λειτουργία της διασταύρωσης είναι συνήθως ικανή να εισάγει σύντομα ποικιλία στον πληθυσμό, επειδή η επιλογή γίνεται βάση της ικανότητας, η λειτουργία συγκεντρώνεται στα άτομα αυτά. Η λειτουργία του αποδεκατισμού προσφέρει ένα πιο γρήγορο τρόπο για αντιμετώπιση αυτής της κατάστασης. Μια συνθήκη καθορίζει πότε θα ξεκινήσει αυτή η λειτουργία και τότε αμέσως διαγράφονται άτομα από τον πληθυσμό.

2.11 Κατάσταση του εξελικτικού συστήματος

Η κατάσταση του συστήματος αποτελείται μόνο από τον τωρινό πληθυσμό. Σε μια υλοποίηση στον ηλεκτρονικό υπολογιστή, είναι επίσης απαραίτητη η αποθήκευση των παραμέτρων ελέγχου, του συνόλου τερματικού, του συνόλου συναρτήσεων και του καλύτερου μέχρι στιγμής ατόμου (αν αποτελεί μέρος της απόφασης αποτελέσματος.)

2.12 Κριτήριο τερματισμού

Το κριτήριο τερματισμού επαληθεύεται όταν ένας προκαθορισμένος αριθμός γενεών έχει τρέξει ή όταν κάποια άλλη συνθήκη, σχετική με το πρόβλημα, ικανοποιηθεί. Η συνθήκη συνήθως περιλαμβάνει την εύρεση μιας 100% σωστής λύσης του προβλήματος ή την ανίχνευση σύγκλισης του πληθυσμού.

2.13 Προσδιορισμός του αποτελέσματος

Υπάρχουν τρεις μέθοδοι προσδιορισμού του αποτελέσματος.

- 1) Το καλύτερο άτομο που παρουσιάστηκε σε οποιαδήποτε γενιά αποτελεί το αποτέλεσμα της εκτέλεσης. Όταν χρησιμοποιείται αυτή η μέθοδος, η κατάσταση του συστήματος αποτελείται από τον τωρινό πληθυσμό και τον μέχρι στιγμής αποθηκευμένο καλύτερο άτομο.
- 2) Το καλύτερο άτομο που βρίσκεται στον πληθυσμό την στιγμή που τερμάτισε ο αλγόριθμος. Σε τέτοια περίπτωση δεν χρειάζεται αποθήκευση ατόμου. Συνήθως φέρνει το ίδιο αποτέλεσμα με την πρώτη μέθοδο, γιατί είτε δημιουργήθηκε σε προηγούμενο πληθυσμό και λόγω της υψηλής ικανότητας αντιγράφηκε στους επόμενους, είτε γιατί η εκτέλεση τερμάτισε επειδή δημιουργήθηκε αυτό το άτομο (ικανοποιεί την συνθήκη.)
- 3) Σε κάποια προβλήματα, όλος ο πληθυσμός ή μέρος του καθορίζεται ως το αποτέλεσμα.

2.14 Παράμετροι ελέγχου

Οι κύριες παράμετροι είναι:

- 1) το μέγεθος του πληθυσμού
- 2) ο μέγιστος αριθμός γενεών που θα εκτελεστούν.

Υπάρχουν όμως πολλές δευτερεύουσες αριθμητικές παράμετροι που πρέπει να ληφθούν υπόψη. Αυτές είναι:

- 1) Πιθανότητα διασταύρωσης
- 2) Πιθανότητα αναπαραγωγής
- 3) Μέγιστο βάθος δέντρου
- 4) Μέγιστο βάθος δέντρου αρχικού πληθυσμού
- 5) Πιθανότητα μετάλλαξης
- 6) Πιθανότητα εναλλαγής
- 7) Συχνότητα σύνταξης
- 8) Πιθανότητα ενθυλάκωσης
- 9) Συνθήκη αποδεκατισμού
- 10) Ποσοστό αποδεκατισμού

2.15 Βήματα Γενετικού Αλγορίθμου

2.15.1 Βήματα Προετοιμασίας του Γενετικού Αλγόριθμου

Ο χρήστης μεταφέρει την υψηλού επιπέδου δήλωση του προβλήματος, στο σύστημα γενετικού προγραμματισμού, κάνοντας κάποιες αρχικές προετοιμασίες. Τα τέσσερα κύρια βήματα για προετοιμασία χρήσης του γενετικού αλγόριθμου, για επίλυση ενός προβλήματος είναι:

- Απόφαση μεθόδου αναπαράστασης των ατόμων.
- Απόφαση μέτρου αξιολόγησης ικανότητας.
- Απόφαση των παραμέτρων και των μεταβλητών ελέγχου του αλγορίθμου.
- Απόφαση του τρόπου κρίσης του αποτελέσματος και του κριτηρίου τερματισμού της εξέλιξης.

2.15.2 Βήματα εκτέλεσης του γενετικού αλγορίθμου

- 1) Δημιούργησε τυχαία ένα αρχικό πληθυσμό από ατομικά προγράμματα υπολογιστή που αποτελούνται από τις διαθέσιμες συναρτήσεις και τα διαθέσιμα τερματικά.
- 2) Ακολούθησε τα εξής υπό-βήματα μέχρι να ικανοποιείται στο κριτήριο τερματισμού του αλγορίθμου.
 - a) Αξιολόγησε την ικανότητα του κάθε ατόμου στον πληθυσμό χρησιμοποιώντας την μέθοδο αξιολόγησης του προβλήματος.
 - b) Επέλεξε ένα ή δύο άτομα από τον πληθυσμό με πιθανότητα επιλεγμένη βασισμένη στην τιμή ικανότητας, για να συμμετάσχουν στις γενετικές λειτουργίες του c.
 - c) Δημιούργησε ένα νέο πληθυσμό από άτομα, χρησιμοποιώντας τις ακόλουθες γενετικές λειτουργίες με καθορισμένες πιθανότητες.
 - i) Αναπαραγωγή: Αντίγραψε το επιλεγμένο άτομο στον νέο πληθυσμό.
 - ii) Διασταύρωση: Δημιούργησε νέα άτομα για τον νέο πληθυσμό συνδυάζοντας γενετικά επιλεγμένα γονίδια από τα δύο επιλεγμένα άτομα.
 - iii) Μετάλλαξη: Δημιούργησε ένα νέο άτομο για τον νέο πληθυσμό αλλάζοντας τυχαία ένα κομμάτι ενός επιλεγμένου ατόμου.
 - iv) Λειτουργίες αλλαγής αρχιτεκτονικής: Επέλεξε μια από τις διαθέσιμες λειτουργίες αλλαγής αρχιτεκτονικής (δευτερεύουσες λειτουργίες) και δημιούργησε ένα νέο άτομο για τον νέο πληθυσμό.
- 3) Αφού ικανοποιηθεί το κριτήριο τερματισμού, το ικανότερο άτομο που εμφανίστηκε σε οποιοδήποτε πληθυσμό αποφασίζεται ως το αποτέλεσμα του γενετικού αλγόριθμου. Αν το τρέξιμο είναι επιτυχές, το αποτέλεσμα μπορεί να αποτελεί λύση του προβλήματος.

2.16 Γιατί Γενετικός Αλγόριθμος

Υπάρχουν διαθέσιμες πολλές ισχυρές αριθμητικές μέθοδοι βελτιστοποίησης. Γιατί κάποιος να θέλει να χρησιμοποιήσει μια εναλλακτική μέθοδο βασισμένη σε πιθανότητες, όπως είναι οι γενετικοί αλγόριθμοι; Η απάντηση εξαρτάται εν μέρει από την εφαρμογή. Αν τα προβλήματα που πρέπει να αντιμετωπιστούν

συμπεριφέρονται σχετικά με φυσιολογικό τρόπο, τότε οι συμβατικές ντετερμινιστικές τεχνικές είναι η καλύτερη λύση. Όταν ο υπολογισμός περιλαμβάνει πολλές παραμέτρους που επιδρούν με τρόπο μη γραμμικό, τότε εμφανίζονται δυσκολίες. Σε τέτοιες περιπτώσεις, ευρετικές μέθοδοι όπως είναι οι γενετικοί αλγόριθμοι αποτελούν μια ισχυρή εναλλακτική λύση και θα μπορούσαν να ενισχύσουν δυναμικά το σύνολο των διαθέσιμων εργαλείων στους ερευνητές. Επίσης οι γενετικοί αλγόριθμοι δουλεύουν πολύ καλά σε προβλήματα όπου για να βρεθεί η σωστή λύση, πρέπει να εξεταστούν όλες οι πιθανές λύσεις με την σειρά και να αξιολογηθεί η ορθότητα τους. Τέτοια προβλήματα όμως έχουν τον περιορισμό ότι το πεδίο έρευνας, στο οποίο βρίσκεται και η λύση, πρέπει να είναι γνωστό από πριν. Σε προβλήματα όπου το πεδίο έρευνας δεν είναι γνωστό ή δεν υπάρχει καν, οι γενετικοί αλγόριθμοι δεν μπορούν να βοηθήσουν.

Η απλότητα με την οποία αντιμετωπίζουν τα προβλήματα είναι ένα από πιο δυνατά χαρακτηριστικά των γενετικών αλγορίθμων. Ένας γενετικός αλγόριθμος είναι πρακτικά η επαναλαμβανόμενη εφαρμογή κάποιων απλών λειτουργιών. Στη πραγματικότητα δεν γνωρίζει τίποτα για το δεδομένο πρόβλημα. Δεν χρειάζεται να γνωρίζει πώς να λύσει ένα πρόβλημα, αλλά πώς να αναγνωρίσει μια καλή λύση όταν την δει.

Ένα ακόμα από τα δυνατά χαρακτηριστικά του γενετικού αλγορίθμου είναι η παράλληλη φύση της πιθανολογικής αναζήτησης του. Λόγω αυτής της παραλληλότητας του, η απόδοση του εξαρτάται λιγότερο από τις αρχικές παραμέτρους του και έτσι ο χρόνος σύγκλισης των ατόμων είναι προβλέψιμος. Το πρόβλημα εύρεσης ενός τοπικού βέλτιστου ελαχιστοποιείται, καθώς ο γενετικός αλγόριθμος κάνει χιλιάδες αρχικές εικασίες.

Η παράλληλη φύση της πιθανολογικής αναζήτησης έχει όμως και συνέπειες. Η σύγκλιση ενός γενετικού αλγορίθμου είναι συνήθως πιο αργή από τις παραδοσιακές τεχνικές. Αν για παράδειγμα ο γενετικός αλγόριθμος κάνει μια αρχική εικασία κοντά στο τοπικό βέλτιστο, μια αριθμητική τεχνική μάλλον θα είναι πιο γρήγορη αφού ο γενετικός αλγόριθμος θα καθυστερήσει αξιολογώντας και τις υπόλοιπες μη βέλτιστες λύσεις.

Επίσης παρόλο που η λύση θα είναι πιθανότερη να υπολογίσει το βέλτιστο, δεν θα είναι παρά μια εκτίμηση. Οι γενετικοί αλγόριθμοι μόνο κατά τύχη θα βρουν μια ακριβή βέλτιστη λύση, ενώ η παραδοσιακές μέθοδοι στην περίπτωση που θα βρουν την λύση, θα είναι ακριβής. Ο χρήστης θα πρέπει να αποφασίσει αν η λύση που βρέθηκε είναι αρκετά κοντά στο επιθυμητό αποτέλεσμα. Το πόσο κοντά στο σωστό αποτέλεσμα χρειάζεται να είναι η λύση, εξαρτάται από την εφαρμογή.

2.17 Βιβλιοθήκες Γενετικών Αλγορίθμων

Υπάρχουν πολλές βιβλιοθήκες γενετικών αλγορίθμων διαθέσιμες στο διαδίκτυο, σε διάφορες γλώσσες προγραμματισμού. Είναι σωστό να αναφερθώ σε κάποιες από αυτές τις βιβλιοθήκες, καθώς χρειάστηκε να τις μελετήσω για να αποφασίσω ποια ήταν η καλύτερη για να χρησιμοποιήσω στην δική μου περίπτωση. Θα αναφερθώ σε βιβλιοθήκες που είναι υλοποιημένες σε C ή C++, καθώς σε κάποια από τις δύο αυτές γλώσσες θα υλοποιηθεί το σύστημά μου, παρόλο που υπάρχουν διαθέσιμες βιβλιοθήκες για άλλες γλώσσες όπως Java και Perl. Οι πιο κάτω βιβλιοθήκες διατίθεται δωρεάν στο διαδίκτυο για να τις κατεβάσει κάποιος και να τις χρησιμοποιήσει.

GAGS genetic algorithm C++ class library

Είναι μια βιβλιοθήκη κλάσεων γραμμένων σε C++, που χρησιμοποιούνται για να προγραμματιστούν όλα τα στοιχεία ενός γενετικού αλγορίθμου. Θέλει gcc

Κύρια Χαρακτηριστικά

- Χρωμοσώματα, οι οποίες είναι οι βασικές δομικές μονάδες των γενετικών αλγορίθμων. Είναι bit strings και έχουν μεταβλητό μήκος.
- Γενετικοί τελεστές, οι οποίοι δεν είναι μέρος της κλάσης των χρωματοσωμάτων. Υπάρχουν εννιά προκαθορισμένοι τελεστές αλλά μπορούν να προστεθούν και νέοι και να χρησιμοποιηθούν με τον ίδιο τρόπο.
- Όψεις, οι οποίες αντιπροσωπεύουν το χρωμόσωμα και χρησιμοποιούνται για να το αξιολογήσουν και να το τυπώσουν.
- Πληθυσμός, ο οποίος περιλαμβάνει την λίστα των χρωματοσωμάτων που θα αξιολογηθούν, και την ταξινομεί.

GAlib

A C++ Library of Genetic Algorithm Components

Είναι μια βιβλιοθήκη κλάσεων γενετικών αλγορίθμων γραμμένων σε C++. Η βιβλιοθήκη περιέχει εργαλεία για χρήση των γενετικών αλγορίθμων για βελτιστοποίηση οποιουδήποτε προγράμματος C++, χρησιμοποιώντας οποιαδήποτε αναπαράσταση ή γενετικούς τελεστές. Η τεκμηρίωση περιλαμβάνει μια εκτενή επισκόπηση για το πώς να εφαρμοστεί ένας γενετικός αλγόριθμος καθώς και παραδείγματα που επεξηγούν προσαρμογές στις GAlib κλάσεις.

GALOPPS 3.2.4 - the "Genetic Algorithm Optimized for Portability and Parallelism System"

Είναι μια ευέλικτη βιβλιοθήκη γενετικών αλγορίθμων γραμμένη σε γλώσσα C.

Κύρια Χαρακτηριστικά

- Πέντε μέθοδοι επιλογής
- Τυχαία ή ομοιόμορφη αρχικοποίηση δυαδικών ή όχι χρωματοσωμάτων.
- Δυαδικά ή όχι χρωματοσώματα.
- Τρία είδη διασταύρωσης για αναπαραστάσεις βασισμένες στην τιμή.
- Τέσσερα είδη διασταύρωσης για αναπαραστάσεις βασισμένες στην σειρά.
- Τέσσερα είδη μετάλλαξης.
- Αξιολόγηση ικανότητας
- Επιλογές αντικατάστασης χρωματοσωμάτων.
- Όλα τα runs μπορούν να ξαναξεκινήσουν από αυτόματα αρχεία σημείων ελέγχου.

Gaul: Genetic Algorithm Utility Library

Είναι μια ευέλικτη βιβλιοθήκη γενετικού προγραμματισμού, σχεδιασμένη για να ενίσχυση την ανάπτυξη εφαρμογών που χρησιμοποιούν γενετικούς ή εξελικτικούς αλγόριθμους.

Κύρια Χαρακτηριστικά

- Εξελικτικές μέθοδοι Darwin, Lamarck ή Baldwin.

- Περιλαμβάνονται γενετικοί αλγόριθμοι καταστάσεων(steady state) και βασισμένων σε γενιές(generation based).
- Μια επιλογή κοινών τύπων χρωμοσωμάτων είναι προκαθορισμένοι.
- Περιλαμβάνει όλους τους συνηθισμένους αλγόριθμους επιλογής, διασταύρωσης και μετάλλαξης.
- Επιτρέπει στον χρήστη να καθορίσει νέους τελεστές για διασταύρωση, μετάλλαξη, επιλογή, προσαρμογή και αντικατάσταση.
- Πλήρης υποστήριξη για πολλαπλούς πληθυσμούς που να εξελίσσονται ταυτόχρονα.
- Επιπρόσθετοι προκαθορισμένοι αλγόριθμοι βελτιστοποίησης.
- Εύκολο να επεκταθεί με εξωτερικό κώδικα, χρησιμοποιώντας τα προκαθορισμένα code hooks.
- Υποστήριξη υπολογισμών σε πολυεπεξεργαστές διάφορων αρχιτεκτονικών.
 - MPI
 - OpenMP 2.0
 - Pthreads
 - Forked-process model

Από τις πιο πάνω βιβλιοθήκες επέλεξα να χρησιμοποιήσω την Gaul: Genetic algorithm utility library, γιατί παρέχει τις περισσότερες δυνατότητες. Είναι υλοποιημένη σε απλή φορητή γλώσσα C, με την οποία είμαι πολύ πιο εξοικειωμένη, δίνει πολλή ελευθερία στον χρήστη και η χρήση της είναι πολύ απλή.

Κεφάλαιο 3:

Εισαγωγή στην έρευνα

3.1 Στόχος έρευνας

3.2 Ιδέα για έναρξη έρευνας

3.3 Δημιουργία βιβλιοθήκης

3.4 Αναπαράσταση χρωμοσωμάτων

3.5 Φωτογραφίες και τύποι κτηρίων

3.1 Στόχος έρευνας

Ο τελικός στόχος αυτής της έρευνας είναι η δημιουργία ενός τρισδιάστατου μοντέλου κτηρίου, το οποίο θα μπορεί να χρησιμοποιηθεί σε διάφορες εφαρμογές γραφικών. Αυτό το μοντέλο θα πρέπει να μοιάζει όσο το δυνατόν πιο πολύ με το πραγματικό κτήριο και να είναι ρεαλιστικό. Το πρόγραμμα που θα δημιουργεί αυτό το μοντέλο, θα παίρνει σαν είσοδο κάποιες φωτογραφίες του κτηρίου από διάφορες πλευρές του και βάση αυτές θα πρέπει να το δημιουργεί.

Στον πραγματικό κόσμο τα κτήρια δεν είναι μόνα τους αλλά συνοδεύονται από διάφορα άλλα αντικείμενα, όπως δέντρα, λουλούδια, κάγκελα κλπ. Το πρόγραμμα αυτό θα πρέπει να μπορεί να δημιουργεί μια τρισδιάστατη σκηνή, μέσα στην οποία θα βρίσκεται το κτήριο και τα διάφορα άλλα αντικείμενα τα οποία το πλαισιώνουν. Εναλλακτικά, το πρόγραμμα θα πρέπει να μπορεί να αφαιρεί τα αντικείμενα τα οποία δεν αποτελούν μέρος του κτηρίου.

3.2 Ιδέα για έναρξη έρευνας

Βασικό μέρος της έρευνας αυτής αποτελεί η χρήση γενετικού αλγόριθμου. Όπως είδαμε η χρήση γενετικών αλγόριθμων είναι πολύ καλή σε προβλήματα όπου πρέπει να εξεταστούν όλες οι πιθανές λύσεις και να βρεθεί η σωστή. Στο πρόβλημα αυτό, υπάρχουν πολλές πιθανές παραλλαγές κτηρίων ανάμεσα στις οποίες υπάρχει και το κτήριο της φωτογραφίας που θα αναζητήσουμε.

Η δημιουργία του μοντέλου του κτηρίου πρέπει να γίνει με τρόπο με τον οποίο θα μπορούμε να αποθηκεύουμε το μοντέλο σε κάποια μορφή, για να μπορούμε να την χρησιμοποιήσουμε σε άλλες εφαρμογές. Αν δεν το καταφέρουμε αυτό, τότε δεν θα πετύχουμε τον στόχο μας. Έτσι οδηγηθήκαμε στο συμπέρασμα ότι η εικόνα που θα δημιουργείται θα πρέπει να αποτελείται από αντικείμενα τα οποία τοποθετημένα στην κατάλληλη θέση και με τον κατάλληλο τρόπο στην σκηνή, θα σχηματίζουν το κτήριο της φωτογραφίας. Τα αντικείμενα αυτά θα είναι χαρακτηριστικά στοιχεία που έχουν τα κτήρια όπως πόρτες, παράθυρα, στέγες, κάγκελα, μπαλκόνια κλπ. Έτσι ο γενετικός αλγόριθμος θα τοποθετεί τυχαία διάφορα αντικείμενα στην σκηνή και θα προσπαθεί να δημιουργήσει μια σκηνή που μοιάζει με αυτή που αναζητεί.

3.3 Δημιουργία βιβλιοθήκης

Ερχόμαστε στο πρόβλημα του πως θα δημιουργείται η βιβλιοθήκη με τα αντικείμενα. Τα αντικείμενα αυτά, όπως και τα κτήρια, έχουν πολλές διαφορετικές αρχιτεκτονικές, χρώματα και σχήματα. Είναι πολύ δύσκολο να δημιουργηθεί μια βιβλιοθήκη που να έχει όλα τα πιθανά αντικείμενα τα οποία θα μπορούσαν να έχουν τα διάφορα κτήρια. Για αυτό τον λόγο μπορούμε να ακολουθήσουμε δύο πιθανές οδούς.

Η πρώτη ιδέα είναι να περιοριστούμε σε ένα συγκεκριμένο είδος αρχιτεκτονικής, στο οποίο τα αντικείμενα των κτηρίων έχουν περιορισμένο αριθμό πιθανών σχημάτων, και να δημιουργήσουμε μια βιβλιοθήκη από αυτά. Αυτή η βιβλιοθήκη θα δημιουργηθεί πριν την χρήση του προγράμματος για δημιουργία ενός μοντέλου. Τα μοντέλα στην βιβλιοθήκη δεν θα έχουν χρώμα ή textures πάνω τους και θα είναι κοινωνικοποιημένα, έτσι ώστε ο γενετικός αλγόριθμος να μπορεί να δοκιμάζει διάφορα χρώματα και μεγέθη του ίδιου αντικειμένου. Σε αυτή την

περίπτωση η συνάρτηση αξιολόγησης θα πρέπει να επεκταθεί για να μπορεί να αναγνωρίζει αν ένα αντικείμενο είναι στην σωστή θέση, αλλά έχει διαφορετικό χρώμα. Η επέκταση αυτή θα γίνει για να βοηθήσει τον γενετικό αλγόριθμο να δώσει καλά αποτελέσματα πιο γρήγορα, καθώς θεωρητικά και χωρίς επέκταση ο γενετικός αλγόριθμος μπορεί να βρει μια καλή λύση.

Η δεύτερη ιδέα είναι η είσοδος επιπλέον φωτογραφιών στο πρόγραμμα, οι οποίες θα απεικονίζουν τα αντικείμενα του κτηρίου που αναζητείται το μοντέλο του. Οι φωτογραφίες αυτές είτε θα πρέπει να φωτογραφηθούν επιπλέον, είτε να γίνει επεξεργασία των αρχικών για να απομονωθούν τα αντικείμενα. Με αυτό τον τρόπο θα μπορεί να εκτελεστεί ο ίδιος γενετικός αλγόριθμος που θα εκτελείται για την δημιουργία του μοντέλου κτηρίου, για δημιουργία μοντέλου ενός αντικειμένου που παρουσιάζεται στην φωτογραφία. Ένα αντικείμενο της φωτογραφίας, αποτελείται κι εκείνο από άλλα αντικείμενα. Για παράδειγμα σε ένα παράθυρο μπορεί να υπάρχουν κάποια τετράγωνα κομμάτια γυαλιό όπως στη φωτογραφία 3.1. Αυτά τα κομμάτια είναι έξι ορθογώνια. Τα περισσότερα αντικείμενα κτηρίων μπορούν να δημιουργηθούν από ένα αριθμό τέτοιων σχημάτων.



Φωτογραφία 3.1

Τα αντικείμενα που θα αναγνωρίζονται, θα εισάγονται στην βιβλιοθήκη η οποία θα χρησιμοποιηθεί για την εύρεση του μοντέλου του κτηρίου αργότερα. Η ιδέα αυτή θα μπορούσε να επεκταθεί, δίνοντας την δυνατότητα στο χρήστη να επιλέξει δυναμικά τα αντικείμενα μέσα από τις φωτογραφίες του κτηρίου, μέσω του προγράμματος.

Είναι ευκολότερο να γίνεται εισαγωγή επιπλέον φωτογραφιών ή να επιλέγει ο χρήστης τα αντικείμενα από τις φωτογραφίες σε σχέση με την δημιουργία μιας βιβλιοθήκης μοντέλων τα οποία ίσως κανείς να μην χρειαστεί. Επιπλέον η δεύτερη ιδέα δίνει την δυνατότητα δημιουργίας μοντέλων για κτήρια διάφορων αρχιτεκτονικών, ενώ η πρώτη περιορίζει στην αρχιτεκτονική που θα επιλεγεί. Η δεύτερη ιδέα έχει το μειονέκτημα ότι θα χρειάζεται περισσότερο χρόνο να εκτελεστεί, αφού θα πρέπει να εκτελεί γενετικούς αλγορίθμους για όλα τα διαφορετικά αντικείμενα του κτηρίου πριν να εκτελέσει για το ίδιο το κτήριο.

Παρόλα αυτά επιλέξαμε να προχωρήσουμε με την δεύτερη μέθοδο αφού αυτή επιτρέπει περισσότερη ποικιλία κτηρίων τα οποία θα μπορεί να αναζητήσει ο χρήστης. Στην περίπτωση όμως που κάποια ή όλα τα αντικείμενα ενός κτηρίου είναι απλά, για παράδειγμα μια ορθογώνια πόρτα χωρίς παράθυρα, η βιβλιοθήκη θα περιέχει τα αντικείμενα που δημιουργηθήκαν αλλά και πρωτεύοντα σχήματα τα οποία θα χρησιμοποιεί για να βρει τα απλά αντικείμενα. Δεδομένου ότι η δημιουργία ενός κτηρίου με την χρήση λογισμικού μοντελοποίησης είναι χρονοβόρα και χρειάζεται εμπειρία, είμαστε διατεθειμένοι να ανεχτούμε, ως ένα σημείο, λίγη καθυστέρηση.

3.4 Αναπαράσταση χρωμοσωμάτων

Για να χρησιμοποιήσω ένα γενετικό αλγόριθμο για να βρω λύση σε αυτό το πρόβλημα, πρέπει να αποφασίσω τι κωδικοποίηση θα χρησιμοποιήσω για τα χρωμοσώματα που θα αναπαριστούν τα κτήρια. Δηλαδή, πρέπει να βρεθεί ένας κοινός τρόπος αναπαράστασης των πιθανών λύσεων αυτού του προβλήματος.

Η διαδικασία για την δημιουργία του κτηρίου συμπεριλαμβάνει πλέον και δημιουργία των αντικειμένων του. Αφού θέλουμε να χρησιμοποιήσουμε τον ίδιο γενετικό αλγόριθμο και για τα δύο αυτά μέρη της εργασίας, πρέπει να έχουν την ίδια αναπαράσταση χρωμοσωμάτων. Ένα χρωμόσωμα θα αναπαριστάται από μια σειρά δυαδικών χαρακτήρων. Έτσι προκύπτει ότι ένα κτήριο θα αναπαρίσταται ως ακολούθως.

Αντικείμενο 1	Αντικείμενο 2	Αντικείμενο 3	...	Αντικείμενο N
---------------	---------------	---------------	-----	---------------

Κάθε αντικείμενο θα έχει κάποια χαρακτηριστικά σε σχέση με το κτηρίο. Αυτά τα χαρακτηριστικά θα αποφασίζονται με την χρήση του γενετικού αλγόριθμου και θα είναι σε σχέση με την ευκρίνεια της εικόνας που θα προκύψει. Αυτά τα χαρακτηριστικά θα είναι:

- Θέση χ
- Θέση ψ
- Μεγέθυνση χ
- Μεγέθυνση ψ
- Απόχρωση
- Αντικείμενο βιβλιοθήκης

Έχουμε αποφασίσει ότι για κάθε αντικείμενο του κτηρίου θα εκτελείται ένας γενετικός αλγόριθμος, για εύρεση των σχημάτων που το αποτελούν. Το αντικείμενο βιβλιοθήκης, θα είναι αυτό το αντικείμενο που θα βρίσκει σε αυτή τη φάση ο γενετικός αλγόριθμος και θα το βάζει στην βιβλιοθήκη. Τα χαρακτηριστικά των σχημάτων θα είναι:

- Θέση χ
- Θέση ψ
- Μεγέθυνση χ
- Μεγέθυνση ψ
- Πρωτεύον σχήμα
- Χρώμα ή texture

Ένα αντικείμενο αυτού του τύπου θα αναπαρίσταται ως ακολούθως.

Θέση χ	Θέση ψ	Μεγέθυνση χ	Μεγέθυνση ψ	Πρωτεύον σχήμα	Χρώμα
--------	--------	-------------	-------------	----------------	-------

3.5 Φωτογραφίες και τύποι κτηρίων

Η προσέγγιση που ακολουθήθηκε για επίλυση αυτού του προβλήματος δεν μπορεί παρά να περιορίσει το είδος των φωτογραφιών που θα πρέπει να εισάγονται στο πρόγραμμα. Οι φωτογραφίες θα πρέπει να παίρνονται από όλες τις πλευρές του σπιτιού. Η φωτογραφική μηχανή πρέπει να είναι παράλληλη σε μια όψη του

σπιτιού και να βλέπει ίσια προς αυτή καλύπτοντας όλη την πλευρά που θα φωτογραφηθεί. Για παράδειγμα μια φωτογραφία μπορεί να δείχνει την πρόσοψη του σπιτιού. Η πιο κάτω φωτογραφία αποτελεί παράδειγμα καλής φωτογραφίας για το πρόγραμμα.



Τα κτήρια τα οποία θα πρέπει να φωτογραφίζονται πρέπει να είναι σχετικά απλά, δηλαδή να μην έχουν περίεργα σχήματα τα οποία να είναι δύσκολο να εντοπιστούν. Το κτήριο της φωτογραφίας είναι παράδειγμα τέτοιου τύπου κτηρίου.

Κεφάλαιο 4:

Υλοποίηση γενετικού αλγόριθμου και μέθοδος αξιολόγησης

- 4.1 Ρόλος Gaul
 - 4.2 Αρχική ιδέα
 - 4.3 Αναπαράσταση λύσης
 - 4.4 Βιβλιοθήκη αντικειμένων
 - 4.5 Σταθερός αριθμός αντικειμένων
 - 4.6 Υλοποίηση γενετικού αλγορίθμου
 - 4.7 Γραφική αναπαράσταση
 - 4.8 Αξιολόγηση Ικανότητας
 - 4.9 Προσομοιώσεις
 - 4.10 Συμπεράσματα Αποτελεσμάτων
 - 4.11 Αλλαγή κατεύθυνσης
-

4.1 Ρόλος Gaul

Θα δούμε τώρα με ποιο τρόπο μπορεί να γίνει η υλοποίηση ενός πολύ απλού γενετικού αλγόριθμου με την χρήση του Gaul, χωρίς να χρησιμοποιήσουμε οποιαδήποτε προηγμένα χαρακτηριστικά της βιβλιοθήκης.

Αρχικά, στον κώδικα που θα χρησιμοποιήσει την Gaul πρέπει να συμπεριληφθεί και η βιβλιοθήκη (`#include <gaul.h>`).

Το πρώτο πράγμα που είναι απαραίτητο, είναι η συνάρτηση αξιολόγησης ικανότητας των ατόμων του πληθυσμού. Την συνάρτηση αυτή την υλοποιεί ο προγραμματιστής του γενετικού αλγόριθμου. Για παράδειγμα, αν τα άτομα ενός προβλήματος είναι σειρές χαρακτήρων τότε ο στόχος ενός προγράμματος θα

μπορούσε να είναι ο τίτλος ενός βιβλίου. Τότε η συνάρτηση αυτή, θα σύγκρινε την σειρά χαρακτήρων του κάθε ατόμου με τον τίτλο του βιβλίου και θα μετρούσε τους κοινούς χαρακτήρες τους. Το αποτέλεσμα θα είναι η τιμή ικανότητας του ατόμου.

Για την δημιουργία του αρχικού πληθυσμού, η βιβλιοθήκη χρησιμοποιεί την δική της γεννήτρια τυχαίων αριθμών, οπότε πρέπει ο προγραμματιστής να καλέσει μια από τις συναρτήσεις `random_init()` ή `random_seed()`. Επίσης πρέπει να ορίσει μια αρχική δομή πληθυσμού, όπου θα αποθηκεύεται ο πληθυσμός της κάθε γενιάς (`population *pop=NULL`).

Για τον καθορισμό των χαρακτηριστικών του πληθυσμού, η `Gaul` χρησιμοποιεί κάποιες προκαθορισμένες συναρτήσεις, για δημιουργία πληθυσμών με προεπιλεγμένες παραμέτρους. Αυτές οι συναρτήσεις ονομάζονται `ga_genesis_XXX()`, όπου το `XXX` αντιπροσωπεύει το είδος της δομής του ατόμου. Στη περίπτωση που τα άτομα είναι σειρές χαρακτήρων, θα χρησιμοποιήσουμε την `ga_genesis_char()`. Οι περισσότερες από τις παραμέτρους της συνάρτησης αυτής, είναι δείκτες σε συναρτήσεις που καλούνται την κατάλληλη στιγμή, για να εκτελέσουν κομμάτια του γενετικού αλγόριθμου. Με αυτό τον τρόπο είναι εύκολο να υλοποιήσει κάποιος δικές του συναρτήσεις για οποιαδήποτε διαδικασία του γενετικού προγραμματισμού, όπως είναι η διασταύρωση, η επιλογή ή η μετάλλαξη.

Παράμετροι

1. Μέγεθος πληθυσμού.
2. Αριθμός χρωμοσωμάτων σε κάθε άτομο.
3. Μέγεθος χρωμοσωμάτων. Εξαρτάται από το είδος του χρωμοσώματος, το οποίο μπορεί και να αγνοεί αυτή την τιμή.
4. Δείκτης σε συνάρτηση η οποία καλείται στην αρχή κάθε γενιάς για εξελικτικές διαδικασίες. Μπορεί να είναι `NULL`.
5. Δείκτης σε συνάρτηση η οποία καλείται στην αρχή κάθε επανάληψης για ντετερμινιστικές διαδικασίες βελτιστοποίησης. Μπορεί να είναι `NULL`.
6. Δείκτης σε συνάρτηση η οποία χειρίζεται κατανομή μνήμης για δεδομένα τα οποία καθορίζει ο χρήστης. Μπορεί να είναι `NULL`.
7. Δείκτης σε συνάρτηση η οποία χειρίζεται κατανομή μνήμης για δεδομένα τα οποία καθορίζει ο χρήστης. Μπορεί να είναι `NULL`.

8. Δείκτης στη συνάρτηση αξιολόγησης ικανότητας.
9. Δείκτης στη συνάρτηση δημιουργίας αρχικού πληθυσμού.
10. Δείκτης στη συνάρτηση προσαρμογής.
11. Δείκτης στη συνάρτηση επιλογής ενός ατόμου.
12. Δείκτης στη συνάρτηση επιλογής δύο ατόμων.
13. Δείκτης στη συνάρτηση μετάλλαξης.
14. Δείκτης στη συνάρτηση διασταύρωσης.
15. Δείκτης στη συνάρτηση αντικατάστασης. Χρησιμοποιείται μόνο στην εξέλιξη σταθερής κατάστασης (steady state evolution). Μπορεί να είναι NULL.
16. Δείκτης σε δεδομένα του χρήστη. Μπορεί να είναι NULL.

Για τις συναρτήσεις 8 μέχρι 15 ισχύουν τα ακόλουθα. Μπορεί να χρησιμοποιηθεί προκαθορισμένη συνάρτηση ή συνάρτηση που καθόρισε ο προγραμματιστής. Κατά την εκτέλεση του αλγόριθμου, η συνάρτηση μπορεί να τροποποιηθεί. Επίσης μπορεί να αλλάξει δυναμικά η συνάρτηση που καλείται για εκτέλεση της λειτουργίας που καθορίζει, κατά την εκτέλεση του αλγόριθμου.

Πριν την εκτέλεση του αλγορίθμου πρέπει να οριστούν κάποιες παράμετροι, που καθορίζουν το είδος της εξέλιξης που θα ακολουθήσει ο γενετικός αλγόριθμος. Για αυτή την περίπτωση καλείται η συνάρτηση `ga_population_set_parameters`.

Παράμετροι:

1. Δείκτης στον πληθυσμό που δηλώθηκε.
2. Το είδος της εξελικτικής μεθόδου που θα ακολουθηθεί. Υπάρχουν διαθέσιμες οι μέθοδοι Darwin, Lamarck ή Baldwin.
3. Καθορίζει το πέρασμα γονέων στις επόμενες γενιές. Είναι ασφαλές να γίνει αλλαγή αυτού του κανόνα και κατά την διάρκεια της εκτέλεσης του αλγόριθμου, καλώντας την συνάρτηση `ga_population_set_elitism()`.
4. Ποσοστό πιθανότητας για διασταύρωση.
5. Ποσοστό πιθανότητας για μετάλλαξη.
6. Ποσοστό πιθανότητας για μετανάστευση. (Ισχύει μόνο για αλγόριθμους που ακολουθούν το πρότυπο νησιού)

Υπάρχουν κάποιες προκαθορισμένες παράμετροι και γι αυτό τον λόγο η κλήση αυτής της συνάρτησης μπορεί να παραληφθεί.

Αφού καθορίστηκαν όλες οι απαραίτητες παράμετροι, τώρα μπορούμε να καλέσουμε την συνάρτηση εξέλιξης. Αυτή είναι η `ga_evolution()` για τους αλγόριθμους βασισμένους σε γενεές και `ga_evolution_steady_state()` για τους αλγόριθμους βασισμένους σε καταστάσεις. Η `ga_evolution()` παίρνει σαν παραμέτρους την δομή του πληθυσμού (`pop`) και τον μέγιστο αριθμό γενεών που θα εκτελεστεί ο αλγόριθμος.

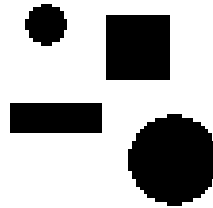
Η Gaul προσφέρει διάφορες παράλληλες υλοποιήσεις του γενετικού αλγόριθμου και κυρίως είναι εναλλακτικές μορφές τις διαδικασίας εξέλιξης. Εξαιρεση αποτελεί η χρήση του OpenMP για παράλληλο κώδικα. Το OpenMP είναι προδιαγραφή για ένα σύνολο οδηγιών για μεταγλωττιστές, βιβλιοθηκών μεταβλητών περιβάλλοντος που μπορεί να χρησιμοποιηθεί για καθορισμό κοινής μνήμης για προγράμματα σε Fortran και C/C++. Αν το πρόγραμμα Gaul μεταγλωττιστεί με υποστήριξη για OpenMP (πρέπει να το υποστηρίζει ο μεταγλωττιστής), τότε ο κώδικας του Gaul μπορεί να ωφεληθεί χωρίς περαιτέρω ενέργειες από τον προγραμματιστή. Οι εναλλακτικές υλοποιήσεις της `ga_evolution()` περιλαμβάνουν τις `ga_evolution_mpi()`, `ga_evolution_forked()` και `ga_evolution_threaded()`. Η `ga_evolution_mpi()` απαιτεί την εγκατάσταση μιας παράλληλης βιβλιοθήκης MPI, η `ga_evolution_forked()` την χρήση `forks` για δημιουργία `forked` διεργασιών και η `ga_evolution_threaded()` το Pthread API το οποίο είναι διαθέσιμο ευρέως στα Linux και Unix.

Τελειώνοντας αποδεσμεύουμε την δομή του πληθυσμού καλώντας `ga_extinction(pop)` και `exit(EXIT_SUCCESS)`.

4.2 Αρχική ιδέα

Το πρόγραμμα που θα υλοποιηθεί θα έχει σαν είσοδο του μια εικόνα ενός κτηρίου. Παρόλο που ο τελικός στόχος είναι να δημιουργηθεί μια τρισδιάστατη αναπαράσταση του κτηρίου που αντιπροσωπεύει, για αρχή θα περιορίσουμε το πρόβλημα στην δημιουργία μιας δυσδιάστατης εικόνας. Αυτή η εικόνα θα δημιουργείται με την μέθοδο του γενετικού προγραμματισμού.

Αποφάσισα να απλοποιήσω περεταίρω το πρόβλημα μου για να μπορέσω να το λύσω πιο εύκολα και επιπλέον να ελέγξω και να επιβεβαιώσω ότι όντως η μέθοδος του γενετικού προγραμματισμού έχει την δυνατότητα να επιλύσει αυτό το πρόβλημα. Οι έγχρωμες εικόνες είναι πιο πολύπλοκές έτσι αποφάσισα για αρχή να χρησιμοποιήσω απλές μαυρόασπρες εικόνες που να έχουν λευκό φόντο και περιέχουν ένα αριθμό από πρωτεύοντα σχήματα (κύκλος, τετράγωνο κλπ). Στόχος πλέον του προγράμματος είναι να δημιουργήσει μια εικόνα σαν την πιο κάτω, σε ένα τρισδιάστατο χώρο.



Παράδειγμα εικόνας που πρέπει να δημιουργεί το πρόγραμμα

4.3 Αναπαράσταση λύσης

Έχω αποφασίσει ότι τα άτομα του πληθυσμού θα αντιπροσωπεύονται από σειρές δυοδιάστατων χαρακτήρων. Όμως αυτό από μόνο του, δεν μου αντιπροσωπεύει μια εικόνα. Πρέπει με κάποιο τρόπο να μπορώ να μεταφράσω αυτή την σειρά σε ένα τρισδιάστατο κτήριο.

Για την συγκεκριμένη υλοποίηση θα ακολουθήσω την λογική που αναφέρθηκε στο κεφάλαιο 3, αφαιρώντας όμως κάποιες παραμέτρους οι οποίες προς το παρόν δεν θα χρησιμοποιούνται. Κάθε αντικείμενο της εικόνας (πχ παράθυρο), θα αντιπροσωπεύεται από μια σειρά δυαδικών χαρακτήρων, το οποίο θα αποτελείται από ένα αριθμό bit για την θέση του αντικειμένου στον x άξονα, ένα αριθμό bit για την θέση του αντικειμένου στον y άξονα, ένα αριθμό bit για την μεγέθυνση του αντικειμένου στον x άξονα, ένα αριθμό bit για την μεγέθυνση του αντικειμένου στον y άξονα και ένα αριθμό bit για κάποιο αντικείμενο της βιβλιοθήκης. Τα πρώτα τέσσερα θα είναι ανάλογα με το μέγεθος της εικόνας και το τελευταίο είναι ανάλογο με το μέγεθος της βιβλιοθήκης των αντικειμένων. Για παράδειγμα αν η εικόνα μου έχει μέγεθος 64 x 64 pixel τότε χρειάζομαι 6 bit για να

αναπαραστήσω την θέση x και ακόμα 6 bit για να αναπαραστήσω την θέση y . Το ίδιο ισχύει και για την μεγέθυνση. Το τελευταίο κομμάτι της αναπαράστασης του αντικειμένου θα καθορίζει το είδος του. Δηλαδή θα αποφασίζει αν θα είναι κύκλος, τετράγωνο, παράθυρο ή πόρτα κλπ.

Θέση x Θέση y Μεγέθυνση x Μεγέθυνση y Είδος αντικ.

Σχήμα αναπαράστασης ενός αντικειμένου

Κάθε χρωμόσωμα θα αποτελείται από ένα αριθμό bit, ο οποίος θα είναι πολλαπλάσιος του συνολικού αριθμού bit για ένα αντικείμενο. Αυτό σημαίνει ότι αν ο αριθμός bit του χρωμοσώματος είναι ίσος με ενός αντικειμένου, τότε το κτήριο που αντιπροσωπεύεται από αυτό το χρωμόσωμα έχει μόνο ένα αντικείμενο.

4.4 Βιβλιοθήκη αντικειμένων

Στο πρόγραμμα μου θα διατηρώ μια βιβλιοθήκη με αντικείμενα, που σε αρχικά στάδια θα αποτελείται από τετράγωνα και κύκλους και μετά από παράθυρα, πόρτες κλπ στοιχεία ενός κτηρίου. Αυτή η βιβλιοθήκη θα είναι ένας πίνακας και κάθε θέση του πίνακα θα έχει ένα αντικείμενο. Τα αντικείμενα στον πίνακα θα είναι κανονικοποιημένα, δηλαδή όλες οι πλευρές τους θα έχουν μήκος 1 έτσι ώστε το μέγεθος τους να αποφασίζεται από τον γενετικό αλγόριθμο, δυναμικά στην κάθε λύση. Κάθε αντικείμενο θα υπάρχει μόνο μια φορά στην βιβλιοθήκη και το χρώμα θα αποφασίζεται από τον γενετικό αλγόριθμο. Προς το παρόν το χρώμα των αντικειμένων θα είναι μαύρο.

4.5 Σταθερός αριθμός αντικειμένων

Το πεδίο έρευνας αυτού του προβλήματος είναι άπειρο. Θέλοντας να βοηθήσω τον γενετικό αλγόριθμο να βρει πιο γρήγορα την λύση που ψάχνω, αποφάσισα ότι θα χρησιμοποιώ σταθερό αριθμό αντικειμένων στα άτομα που θα αντιπροσωπεύουν την λύση. Με αυτό τον τρόπο όλα τα άτομα στον πληθυσμό θα έχουν ίσο μέγεθος

με το άτομο που αντιπροσωπεύει την σωστή λύση κι έτσι περιορίζεται το πεδίο έρευνας.

4.6 Υλοποίηση γενετικού αλγορίθμου

Θα μιλήσω τώρα για τις αποφάσεις που πήρα για την υλοποίηση του γενετικού αλγορίθμου που θα χρησιμοποιήσω στην εργασία.

4.6.1 Μοντέλο αντικατάστασης

Ο γενετικός αλγόριθμος θα χρησιμοποιεί το μοντέλο ολικής γενεαλογικής αντικατάστασης, όπου το μέγεθος του πληθυσμού που ορίζεται στην αρχή παραμένει το ίδιο σε όλες τις γενιές.

4.6.2 Κωδικοποίηση

Αποφάσισα να χρησιμοποιήσω δυαδική κωδικοποίηση για να αντιπροσωπεύσω τα άτομα στον πληθυσμό. Ο λόγος που επέλεξα αυτή την μέθοδο για κωδικοποίηση είναι γιατί είναι απλή και μπόρεσα να βρω πολλές πληροφορίες γύρω από αυτή. Επίσης η βιβλιοθήκη γενετικών αλγορίθμων που χρησιμοποιώ παρέχει όλες τις απαραίτητες λειτουργίες γενετικού αλγορίθμου που λειτουργούν πάνω σε χρωμοσώματα με αυτή την κωδικοποίηση.

4.6.3 Επιλογή

Για την επιλογή των χρωματοσωμάτων που θα λαμβάνουν μέρος στις γενετικές λειτουργίες αποφάσισα να χρησιμοποιώ επιλογή διαγωνισμού. Είναι ένας αρκετά χρησιμοποιημένος τρόπος επιλογής σε γενετικούς αλγόριθμους και πιστεύω στην περίπτωση του προβλήματος που αντιμετωπίζω είναι ο πιο κατάλληλος.

4.6.4 Γενετικές διαδικασίες

Οι γενετικές διαδικασίες που θα χρησιμοποιηθούν είναι η διασταύρωση και η μετάλλαξη. Η διαδικασία της διασταύρωσης θα γίνεται με επιλογή δύο σημείων στα χρωμοσώματα και θα δημιουργούνται δύο απόγονοι. Η διαδικασία της

μετάλλαξης θα γίνεται με επιλογή ενός σημείου στο χρωμόσωμα και αντιστροφή του δυαδικού χαρακτήρα σε αυτό το σημείο.

4.6.5 Κριτήριο τερματισμού

Η εκτέλεση του γενετικού αλγορίθμου θα τερματίζει όταν γίνει εξέλιξη του μέγιστου αριθμός γενιών που θα δηλωθεί και αν βρεθεί η σωστή λύση. Αν ο πληθυσμός συγκλίνει πριν εκπληρωθεί κάποιο από τα άλλα κριτήρια τερματισμού, η εκτέλεση θα τερματίζει.

4.6.6 Προσδιορισμός αποτελέσματος

Όταν η εκτέλεση του αλγορίθμου τερματίσει τότε θα καθορίζεται το αποτέλεσμα. Θα ταξινομούνται τα χρωμοσώματα με βάση την τιμή ικανότητας τους και αυτό που θα είναι πρώτο στην σειρά, δηλαδή αυτό που θα έχει την πιο ψηλή τιμή, θα παρουσιάζεται ως λύση της εκτέλεσης.

4.6.7 Παράμετροι ελέγχου

Το μέγεθος του πληθυσμού αλλά και ο μέγιστος αριθμός γενεών που θα εξελιχτούν εξαρτάται από την φωτογραφία που αναζητείται και τον αριθμό αντικειμένων μέσα σε αυτή. Επίσης εξαρτάται από το μέγεθος της βιβλιοθήκης αντικειμένων.

Η πιθανότητα διασταύρωσης θα τοποθετηθεί στο 90% και η πιθανότητα μετάλλαξης στο 20%.

4.7 Γραφική αναπαράσταση

Η τρισδιάστατη αναπαράσταση της λύσης θα παρουσιάζεται σε γραφικό περιβάλλον. Την δυνατότητα αυτή μας την δίνει η OpenGL. Η OpenGL είναι μια διεπαφή λογισμικού στο υλικό γραφικής παράστασης. Αυτή η διεπαφή περιλαμβάνει πολλές εντολές, που καθορίζουν αντικείμενα και απαραίτητες λειτουργίες για να παραγάγουν διαλογικές τρισδιάστατες εφαρμογές.

Η γραφική αναπαράσταση είναι απαραίτητη προς το παρόν μόνο για την παρουσίαση του τελικού αποτελέσματος. Σε μελλοντικό στάδιο θα είναι δυνατή η

μετακίνηση μέσα στο τρισδιάστατο χώρο που δημιουργείται για να παρατηρηθούν καλύτερα τα κτήρια που θα δημιουργούνται.

4.8 Αξιολόγηση ικανότητας

Η αξιολόγηση της ικανότητας του κάθε ατόμου γίνεται κατά την διάρκεια της εξέλιξης και αφού δημιουργηθεί ο πληθυσμός. Στη περίπτωση αυτού του προβλήματος, η αξιολόγηση θα συμπεριλαμβάνει με κάποιο τρόπο την εικόνα που θα δίνεται σαν είσοδος. Σε αυτό το σημείο, αποφάσισα να περνώ σαν είσοδο του συστήματος ένα χάρτη των ιχνοστοιχείων της εικόνας αντί για την ίδια την εικόνα. Όταν θα έρθει η στιγμή να γίνει αξιολόγηση ενός χρωμοσώματος, θα δημιουργώ ένα χάρτη της εικόνας που αντιπροσωπεύει το χρωμόσωμα και θα τον συγκρίνω με τον χάρτη που θα περνώ σαν είσοδο.

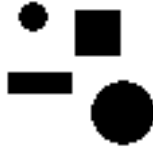
Για την δημιουργία του χάρτη θα χρησιμοποιήσω τον τρόπο που τοποθετεί τα αντικείμενα η OpenGL και θα υπολογίζω ποια ιχνοστοιχεία καλύπτονται. Στην αρχική αυτή ιδέα, όπου τα αντικείμενα είναι όλα μαύρα και το φόντο άσπρο, αυτός ο υπολογισμός είναι αρκετά εύκολος.

4.9 Προσομοιώσεις

Οι ακόλουθες προσομοιώσεις έγιναν με στόχο να ελεγχθεί κατά πόσο είναι ικανός ο γενετικός αλγόριθμος να μας δώσει αποτελέσματα, τα οποία να είναι αρκετά ικανοποιητικά για να προχωρήσουμε με αυτή την ιδέα. Επίσης θα παρατηρήσουμε αν η εξίσωση αξιολόγησης είναι σωστή και θα μετρήσουμε την απόδοση του προγράμματος με την μέχρι τώρα υλοποίηση. Έχουν γίνει αρκετές προσομοιώσεις για εικόνες με ευκρίνεια 64x64, 128x128 και 256x256. Για κάθε προσομοίωση θα αναφέρονται οι παράμετροι πληθυσμός, γενιές αλλά και ο χρόνος που χρειάστηκε για να εκτελεστεί. Οι γενιές που αναφέρονται είναι ο μέγιστος αριθμός γενεών που μπορούν να εκτελεστούν αλλά όπως αναφέρθηκε και πιο πριν, το πρόγραμμα μπορεί να βρει αποτέλεσμα πιο γρήγορα, με άλλους τρόπους.

4.9.1 Ευκρίνεια 64x64

Εικόνα εισόδου



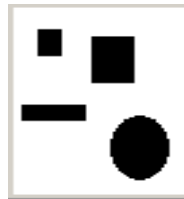
Πρώτη προσομοίωση

Πληθυσμός: 1000

Γενιές: 200

Χρόνος εκτέλεσης: 19 λεπτά και 42 δευτερόλεπτα

Η εικόνα αποτέλεσμα είναι:



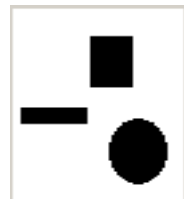
Δεύτερη προσομοίωση

Πληθυσμός: 2000

Γενιές: 150

Χρόνος εκτέλεσης: 45 λεπτά και 48 δευτερόλεπτα

Η εικόνα αποτέλεσμα είναι:



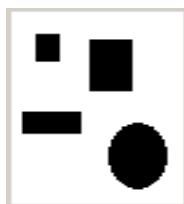
Τρίτη προσομοίωση

Πληθυσμός: 3000

Γενιές: 150

Χρόνος εκτέλεσης: 58 λεπτά και 34 δευτερόλεπτα

Η εικόνα αποτέλεσμα είναι:



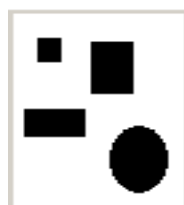
Τέταρτη προσομοίωση

Πληθυσμός: 5000

Γενιές: 150

Χρόνος εκτέλεσης: 3 ώρες, 50 λεπτά και 40 δευτερόλεπτα

Η εικόνα αποτέλεσμα είναι:



4.9.2 Ευκρίνεια 128x128

Εικόνα εισόδου



Πρώτη προσομοίωση

Πληθυσμός: 1000

Γενιές: 200

Χρόνος εκτέλεσης: 16 λεπτά και 27 δευτερόλεπτα

Η εικόνα αποτέλεσμα είναι:

Δεύτερη προσομοίωση

Πληθυσμός: 2000

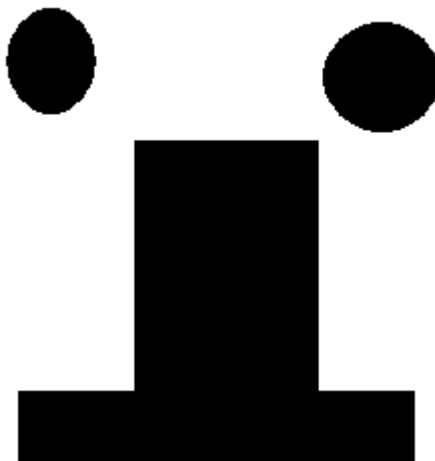
Γενιές: 150

Χρόνος εκτέλεσης:

Η εικόνα αποτέλεσμα είναι: 1 ώρα, 51 λεπτά και 36 δευτερόλεπτα

4.9.3 Ευκρίνεια 256x256

Εικόνα εισόδου



Πρώτη προσομοίωση

Πληθυσμός: 1000

Γενιές: 200

Χρόνος εκτέλεσης: 24 λεπτά και 41 δευτερόλεπτα

Η εικόνα αποτέλεσμα είναι:

Δεύτερη προσομοίωση

Πληθυσμός: 2000

Γενιές: 150

Χρόνος εκτέλεσης: 50 λεπτά και 44 δευτερόλεπτα

Η εικόνα αποτέλεσμα είναι:

Τρίτη προσομοίωση

Πληθυσμός: 3000

Γενιές: 150

Χρόνος εκτέλεσης: 1 ώρα και 22 λεπτά

Η εικόνα αποτέλεσμα είναι:

Τέταρτη προσομοίωση

Πληθυσμός: 5000

Γενιές: 150

Χρόνος εκτέλεσης: 3 ώρες και 15 λεπτά

Η εικόνα αποτέλεσμα είναι:

4.10 Συμπεράσματα αποτελεσμάτων

Παρατηρούμε ότι για όλες τις ευκρίνειες ο γενετικός αλγόριθμος βρίσκει αποτελέσματα κοντά στην λύση που ψάχνουμε. Σε καμιά από τις προσομοιώσεις όμως, δεν βρήκε ακριβώς την σωστή λύση. Στην ευκρίνεια 128x128, βρίσκει την πιο καλή λύση και σε πιο λίγο χρόνο από τις άλλες περιπτώσεις. Στη ευκρίνεια 256x256, βρίσκει την καλύτερη λύση της με τον πιο αργό χρόνο. Είναι λογικό εικόνες με μεγαλύτερη ευκρίνεια να χρειάζονται περισσότερο χρόνο για να βρουν κάποιο σωστό αποτέλεσμα. Επίσης χρειάζεται να αυξηθεί το μέγεθος του πληθυσμού αφού μεγαλύτερη ευκρίνεια σημαίνει μεγαλύτερο μέγεθος των δυναδικών αναπαραστάσεων(bitstring) των εικόνων.

Αξίζει να σημειωθεί ότι σε όλα τα αποτελέσματα, ο γενετικός αλγόριθμος τερματίζει γιατί ο πληθυσμός συγκλίνει. Σε καμία περίπτωση δεν βρίσκει το σωστό αποτέλεσμα και δεν χρειάζεται να εκτελέσει όλες τις γενιές για να τερματίσει.

Το συμπέρασμα που μπορεί να βγάλουμε από αυτά τα αποτελέσματα είναι ότι ο γενετικός αλγόριθμος μπορεί να βοηθήσει στην επίλυση του προβλήματος που προσπαθούμε να λύσουμε. Όμως αντιμετωπίζουμε πρόβλημα λόγω του χρόνου που θέλει για να βρει μια ικανοποιητική λύση.

4.11 Αλλαγή κατεύθυνσης

Ο γενετικός αλγόριθμος λόγω της φύσης του θέλει αρκετό χρόνο για να αποφέρει αποτελέσματα, όταν το πεδίο έρευνας είναι μεγάλο. Λόγω της φύσης του προγράμματος, που έχει να κάνει με εικόνες, είναι δύσκολο μια λύση για το πρόβλημα της αξιολόγησης να είναι απλή και γρήγορη.

Η μέθοδος που επέλεξα για να κάνω την αξιολόγηση της ικανότητας των ατόμων του πληθυσμού, παρόλο που μου δίνει κάποια αρκετά ενθαρρυντικά αποτελέσματα, είναι πολύ χρονοβόρα. Για την απλή ιδέα που υλοποιεί μέχρι τώρα, η δημιουργία της λειτουργίας αξιολόγησης ήταν απλή, όμως όταν θα προχωρήσουμε σε πειράματα με έγχρωμες εικόνες που είναι πιο πολύπλοκες, η λειτουργία θα χρειαστεί τροποποίηση. Παρόλα τα μειονεκτήματα της, η μέθοδος αυτή, απόδειξε ότι μπορούμε να χρησιμοποιήσουμε τον γενετικό αλγόριθμο για να προσπαθήσουμε να βρούμε λύση σε αυτό το πρόβλημα.

Στόχος μας είναι πλέον η δημιουργία μιας λειτουργίας αξιολόγησης που παρόλο που η φύση της είναι πολύπλοκη, αυτή να είναι όσο πιο απλή γίνεται. Αυτή η λειτουργία θα πρέπει να μπορεί να διατηρεί την ακεραιότητα της σε όλες τις πιθανές επεκτάσεις της ιδέας, χωρίς να χρειάζεται τροποποίηση. Στο επόμενο κεφάλαιο, θα μιλήσω για την νέα μέθοδο αξιολόγησης η οποία κατέχει αυτά τα χαρακτηριστικά.

Κεφάλαιο 5:

Βελτιωμένη μέθοδος αξιολόγησης

- 5.1 Δεύτερη μέθοδος αξιολόγησης
 - 5.2 Η γλώσσα cg και τα Shaders
 - 5.3 OpenGL extensions
 - 5.4 Frame buffer objects
 - 5.5 Occlusion queries
 - 5.6 Υλοποίηση αξιολόγησης
 - 5.7 Προσομοιώσεις
 - 5.8 Συμπεράσματα αποτελεσμάτων
-

5.1 Δεύτερη μέθοδος αξιολόγησης

Για την νέα μέθοδο αξιολόγησης, θα περνώ σαν είσοδο του συστήματος την ίδια την φωτογραφία σε μορφή bitmap και θα την αποθηκεύω σε μορφή σύστασης (texture).

Όταν θα γίνεται η αξιολόγηση, θα μετατρέπω σε μορφή σύστασης (texture) την γραφική αναπαράσταση του χρωμοσώματος που είναι η σειρά του να αξιολογηθεί. Έπειτα θα συγκρίνω το texture που θα αντιπροσωπεύει την εικόνα εισόδου με το texture που αντιπροσωπεύει το συγκεκριμένο άτομο. Ο αριθμός των κοινών ιχνοστοιχείων θα αποτελεί την τιμή ικανότητας του ατόμου.

Αυτή η μέθοδος μοιάζει πολύ με την προηγούμενη, γιατί πάλι στην ουσία συγκρίνω τα ιχνοστοιχεία των εικόνων. Σε αυτή την περίπτωση δεν δημιουργώ εγώ τον χάρτη με τα ιχνοστοιχεία, αλλά εκμεταλλεύομαι κάποιες δυνατότητες που μου

δίνουν οι μοντέρνες κάρτες γραφικών για να μετατρέψω την γραφική αναπαράσταση του χρωμοσώματος σε texture και να συγκρίνω δύο texture.

5.2 Η γλώσσα cg και τα Shaders

Στη νέα μέθοδο αξιολόγησης αποφάσισα να εκμεταλλευτώ κάποιες δυνατότητες που προσφέρουν οι κάρτες γραφικών, έτσι ώστε η αξιολόγηση να γίνεται πιο γρήγορα. Οι κάρτες γραφικών διαθέτουν GPU. Το GPU, λόγω της εξειδικευμένης του σχεδίασης, είναι πολύ πιο γρήγορο σε τομείς γραφικών, όπως για παράδειγμα το σχεδιάσμα τρισδιάστατων σκηνών, παρά ένα CPU γενικού σκοπού. Οι καινούριοι GPU επεξεργάζονται δεκάδες εκατομμύρια κορυφές το δευτερόλεπτο και δισεκατομμύρια εικονοστοιχεία. Η υψηλή απόδοση και η εξειδικευμένη φύση του GPU, οδήγησε στην δημιουργία της γλώσσας προγραμματισμού cg, που είναι εξειδικευμένη για προγραμματισμό GPU. Ένα πρόγραμμα που εκτελείται πάνω στο GPU ονομάζεται Shader.

Η Cg είναι μια γλώσσα προγραμματισμού η οποία δίνει την δυνατότητα να ελέγξει κάποιος το σχήμα, την εμφάνιση και την κίνηση των αντικειμένων που σχεδιάζονται, χρησιμοποιώντας εξοπλισμό γραφικών που μπορεί να προγραμματιστεί. Δίνει στους προγραμματιστές απόλυτο έλεγχο στις εικόνες πραγματικού χρόνου που δημιουργούν.

Τα προγράμματα σε cg χωρίζονται σε αυτά που εκτελούνται σε κορυφές (vertex Shaders) και αυτά που εκτελούνται σε εικονοστοιχεία(pixel Shaders). Σε αυτή την εργασία θα χρησιμοποιήσουμε προγράμματα για εικονοστοιχεία, αφού για την μέθοδο αξιολόγησης θα συγκρίνουμε δύο textures, τα οποία αποτελούνται από εικονοστοιχεία. Η cg είναι παρόμοια γλώσσα με την C, καθώς οι δομές και τα αναγνωριστικά(ονόματα μεταβλητών, συναρτήσεων κλπ) καθορίζονται με την ίδια σύνταξη. Είναι στην ουσία η C για γραφικά.

Στις γλώσσες προγραμματισμού γενικού σκοπού, το λειτουργικό σύστημα καλεί την κύρια ρουτίνα (main) και το πρόγραμμα εκτελεί τον κώδικα που βρίσκεται σε αυτή. Όταν η κύρια ρουτίνα τερματίσει, τότε το πρόγραμμα τερματίζει. Αντίθετα στην cg, ο μεταγλωττιστής μεταφράζει το πρόγραμμα σε μια μορφή, που η διεπαφή προγραμματισμού γραφικών μπορεί να μεταφέρει στο

υλικό. Την κλήση των απαραίτητων ρουτινών για να διαμόρφωση του προγράμματος για να το χρησιμοποιήσει το GPU, αναλαμβάνει η εφαρμογή που χρησιμοποιεί το cg πρόγραμμα.

Όταν φορτωθεί το πρόγραμμα στον επεξεργαστή εικονοστοιχείων του GPU, το εκτελεί κάθε φορά που η εφαρμογή στείλει ένα εικονοστοιχείο στο GPU. Έτσι το πρόγραμμα εκτελείται μια φορά για κάθε εικονοστοιχείο.

5.3 Επεκτάσεις OpenGL

Η έκδοση της OpenGL, για την οποία υπάρχουν επικεφαλίδες και βιβλιοθήκες για το λειτουργικό σύστημα Windows, είναι η 1.1. Αυτή η έκδοση είναι αρκετά παλιά και δεν μπορεί να εκμεταλλευτεί μοντέρνο εξοπλισμό γραφικών. Παρόλο που οι διαθέσιμες επικεφαλίδες και βιβλιοθήκες δεν επιτρέπουν την άμεση πρόσβαση σε πιο καινούριες λειτουργικότητες, οι κάρτες γραφικών τις διαθέτουν. Για να μπορέσουμε να έχουμε πρόσβαση σε αυτές, χρησιμοποιείται ο μηχανισμός των επεκτάσεων της OpenGL (OpenGL extensions.)

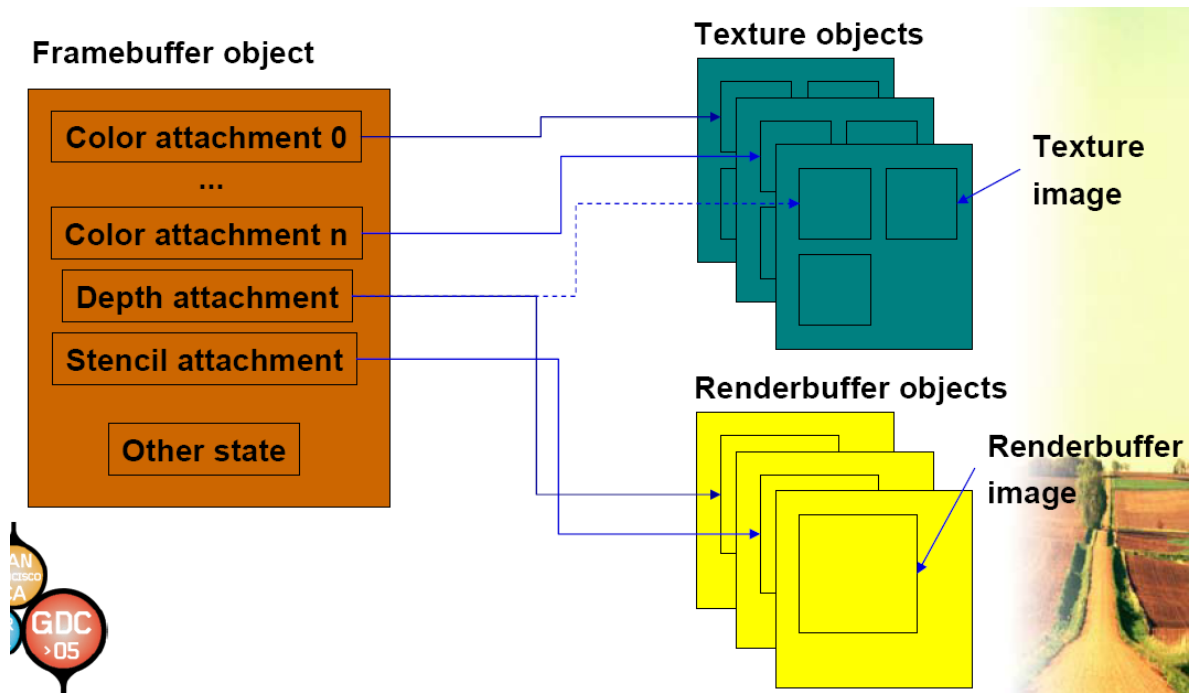
Οι κατασκευαστές εξοπλισμού γραφικών, δημιουργούν αρκετά συχνά τέτοιες επεκτάσεις. Μπορεί να έχουν παρόμοια λειτουργικότητα αλλά πολύ διαφορετική σημασιολογία και τρόπο χρήσης. Για αυτό το λόγο κάθε επέκταση OpenGL έχει μοναδικό όνομα. Οι περισσότερες επεκτάσεις προσφέρουν νέες λειτουργίες/συναρτήσεις.

Για να χρησιμοποιηθεί μια επέκταση πρέπει πρώτα να είναι διαθέσιμη. Έτσι πρέπει πρώτα να ελεγχθεί αν η κάρτα γραφικών την υποστηρίζει. Ακολούθως, για την χρήση μιας λειτουργίας που προσφέρει η επέκταση, πρέπει να γίνει αίτηση του δείκτη στην συνάρτηση από το ICD(Installable Client Driver) της κάρτας γραφικών. Το ICD προσφέρεται μέσω των οδηγών της κάρτας γραφικών, για να επιτρέψει επιτάχυνση των λειτουργικοτήτων της OpenGL, μέσω της κάρτας.

5.4 Frame buffer objects

Το Frame buffer object είναι μια επέκταση της OpenGL η οποία καθορίζει μια απλή διεπαφή για ζωγράφισμα σε προορισμούς εκτός από τα buffers που προσφέρει το σύστημα παραθύρων στην OpenGL.

Για την επέκταση αυτή, οι νέοι προορισμοί είναι γνωστοί ως εικόνες που προσκολλούνται στο Frame buffer(Frame buffer-attachable images.) Προσφέρει ένα μηχανισμό για προσκόλληση εικόνων στο Frame buffer ως ένα από τα καθορισμένα buffers της OpenGL: color, depth, stencil, accumulation. Όταν ένα Frame buffer object είναι ενεργό, οι προσκολλημένες εικόνες πάνω του είναι η πηγή και ο προορισμός των λειτουργιών σε εικονοστοιχεία(pixel operations) που αλλιώς γίνονται στο Frame buffer του παραθύρου. Ο προγραμματιστής έχει την δυνατότητα να αλλάζει από το ένα Frame buffer object στο άλλο. Το κύριο πλεονέκτημα αυτής της επέκτασης είναι ότι προσφέρει τη δυνατότητα ζωγραφίσματος κατευθείαν σε texture.



Για την χρήση ενός Frame buffer object, πρέπει πρώτα να δημιουργηθεί, να προσκολληθεί κάποιο texture πάνω του και έπειτα να γίνει ενεργό για να εκτελεστούν κάποιες λειτουργίες πάνω του. Μπορεί για παράδειγμα να ζωγραφίσει μια σκηνή, η οποία θα ζωγραφιστεί κατευθείαν στο texture που προσκολλήθηκε πάνω του.

5.5 ARB Occlusion query

Η ARB_occlusion_query είναι ακόμη μια επέκταση της OpenGL. Αυτή η επέκταση καθορίζει ένα μηχανισμό, με τον οποίο μπορεί μια εφαρμογή να ζητήσει τον αριθμό των εικονοστοιχείων που καλύπτει ένα πρωτεύον σχήμα, ή σύνολο πρωτευόντων σχημάτων, όταν ζωγραφιστεί.

Ο κυριότερος σκοπός μιας τέτοιας αναζήτησης, είναι για να ορίσει αν ένα αντικείμενο είναι ορατό. Συνήθως, η εφαρμογή θα ζωγραφίσει τα αντικείμενα στη σκηνή, μετά θα εκτελέσει ένα occlusion query(αναζήτηση επικάλυψης) για τον περιβάλλον όγκο του κάθε αντικειμένου στην σκηνή. Αν έστω και ένα εικονοστοιχείο, το οποίο βρίσκεται στον περιβάλλον όγκο του αντικειμένου, ζωγραφιστεί, τότε το αντικείμενο θα ζωγραφιστεί.

5.6 Υλοποίηση αξιολόγησης

Αρχικά το πρόγραμμα παίρνει σαν είσοδο την εικόνα την οποία θέλουμε να δημιουργήσουμε. Αυτή την εικόνα την μετατρέπουμε σε texture, το οποίο βρίσκεται πλέον στην μνήμη του προγράμματος. Ακολούθως, την στιγμή που θα γίνεται η σύγκριση, η αναπαράσταση του χρωμοσώματος που θα αξιολογηθεί πρέπει να μετατραπεί σε texture και εδώ χρησιμοποιούμε ένα Frame buffer object.

Για να ζωγραφιστεί η αναπαράσταση του χρωμοσώματος στο texture, ενεργοποιούμε ένα Frame buffer object, το οποίο έχουμε αρχικοποιήσει στις αρχικοποιήσεις του προγράμματος και στο οποίο έχουμε προσκολλήσει ένα δείκτη σε texture. Ακολούθως, ζωγραφίζουμε την σκηνή του χρωμοσώματος και μετά απενεργοποιούμε το Frame buffer object, ενεργοποιώντας κάποιο άλλο (συνήθως το προκαθορισμένο Frame buffer.)

Για την σύγκριση των δύο textures έχω υλοποιήσει ένα πρόγραμμα Shader σε cg και χρησιμοποιώ την επέκταση occlusion query για να μετρήσω τον αριθμό των κοινών εικονοστοιχείων των δύο textures. Πριν να καλέσω το cg πρόγραμμα αρχικοποιώ το occlusion query και θέτω σαν τιμή ορίου το 0.5. Έτσι θεωρούνται ορατά στοιχεία όσα έχουν alpha μεγαλύτερο του 0.5 και μη ορατά όσα έχουν alpha μικρότερο του 0.5. Το alpha καθορίζει την ορατότητα ενός εικονοστοιχείου. Το πρόγραμμα παίρνει σαν είσοδο δύο textures (την εικόνα εισόδου και την

αναπαράσταση του χρωμοσώματος προς αξιολόγηση) και έχει σαν έξοδο ένα άλλο texture. Το πρόγραμμα συγκρίνει τα εικονοστοιχεία των δύο textures και όταν τα δύο αντίστοιχα εικονοστοιχεία είναι ίσα (έχουν ίση τιμή κόκκινου, πράσινου και μπλε) θέτει στο alpha του εικονοστοιχείου εξόδου την τιμή 1. Σε αντίθετη περίπτωση, δίνει την τιμή μηδέν. Η τιμή των χρωμάτων του εικονοστοιχείου εξόδου δεν μας ενδιαφέρει.

Μόλις τελειώσει το cg πρόγραμμα την εκτέλεση του, καλούμε μια αναζήτηση occlusion query για να μετρήσουμε τον αριθμό των στοιχείων που έχουν τιμή alpha μεγαλύτερο του 0.5, τα οποία είναι κοινά στα δύο textures εισόδου. Αυτός ο αριθμός αποτελεί την τιμή ικανότητας για το χρωμόσωμα που αξιολογείται.

5.7 Προσομοιώσεις

Έχουν γίνει αρκετές προσομοιώσεις για εικόνες με ευκρίνεια 64x64, 128x128 και 256x256. Για κάθε προσομοίωση θα αναφέρονται οι παράμετροι πληθυσμός, γενιές αλλά και ο χρόνος που χρειάστηκε για να εκτελεστεί. Οι γενιές που αναφέρονται είναι ο μέγιστος αριθμός γενεών που μπορούν να εκτελεστούν αλλά όπως αναφέρθηκε και πιο πριν, το πρόγραμμα μπορεί να βρει αποτέλεσμα πιο γρήγορα, με άλλους τρόπους.

5.7.1 Ευκρίνεια 64x64

Εικόνα εισόδου



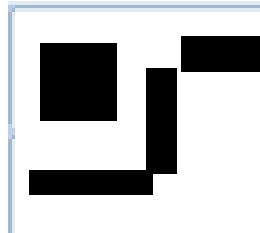
Πρώτη προσομοίωση

Πληθυσμός: 1001

Γενιές: 200

Χρόνος εκτέλεσης: 42 λεπτά

Η εικόνα αποτέλεσμα είναι:



5.7.2 Ευκρίνεια 128x128

Εικόνα εισόδου



Πρώτη προσομοίωση

Πληθυσμός: 1000

Γενιές: 200

Χρόνος εκτέλεσης: 41 λεπτά και 31 δευτερόλεπτα

Η εικόνα αποτέλεσμα είναι:



5.7.3 Ευκρίνεια 256x256

Εικόνα εισόδου



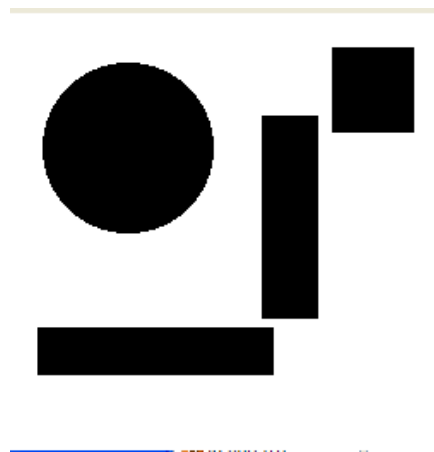
Πρώτη προσομοίωση

Πληθυσμός: 1000

Γενιές: 200

Χρόνος εκτέλεσης: 32 λεπτά

Η εικόνα αποτέλεσμα είναι:



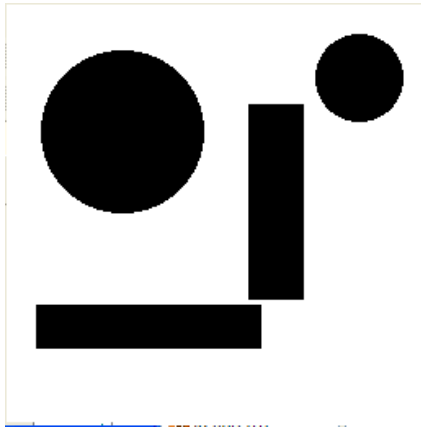
Δεύτερη προσομοίωση

Πληθυσμός: 2000

Γενιές: 200

Χρόνος εκτέλεσης: 1 ώρα και 2 λεπτά

Η εικόνα αποτέλεσμα είναι:



5.7.4 Ευκρίνεια 500x500

Εικόνα εισόδου



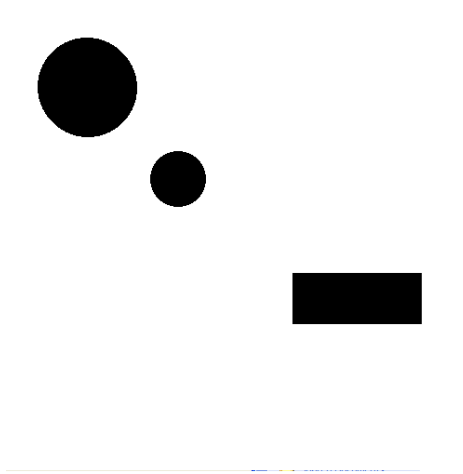
Πρώτη προσομοίωση

Πληθυσμός: 2000

Γενιές: 200

Χρόνος εκτέλεσης: 1 ώρα και 22 λεπτά

Η εικόνα αποτέλεσμα είναι:



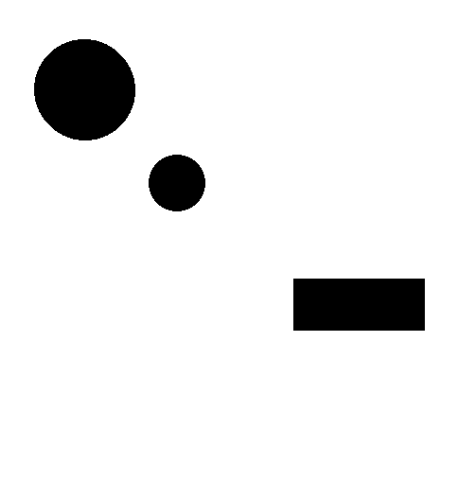
Δεύτερη προσομοίωση

Πληθυσμός: 3000

Γενιές: 100

Χρόνος εκτέλεσης: 2 ώρα και 16 λεπτά

Η εικόνα αποτέλεσμα είναι:



5.8 Συμπεράσματα αποτελεσμάτων

Οι πρώτες παρατηρήσεις που μπορούν να γίνουν είναι οι ίδιες με τις παρατηρήσεις για τις προσομοιώσεις της πρώτης μεθόδου αξιολόγησης. Ο γενετικός αλγόριθμος βρίσκει λύσεις η οποίες είναι κοντά σε αυτή που ψάχνει να βρει. Επίσης για ψηλότερες ευκρίνειες χρειάζεται μεγαλύτερο μέγεθος πληθυσμού και περισσότερο

χρόνο για να βρει σωστό αποτέλεσμα, σε σχέση με τις πιο μικρές ευκρίνειες. Από τα αποτελέσματα, βλέπουμε ότι το καλύτερο (ικανοποιητικό αποτέλεσμα και λίγος χρόνος) πείραμα είναι στην ευκρίνεια 128x128.

Όπως και στην προηγούμενη μέθοδο, οι προσομοιώσεις τερματίζουν γιατί ο πληθυσμός συγκλίνει και όχι γιατί έφτασε στη σωστή λύση ή στο όριο των γενεών που ορίστηκε.

Η νέα συνάρτηση αξιολόγησης μας δίνει ικανοποιητικά αποτελέσματα για να προχωρήσουμε σε κανονικές φωτογραφίες.

Κεφάλαιο 6:

Εστίαση στην εύρεση αντικειμένων

- 6.1 Προσέγγιση
 - 6.2 Αλλαγές στο υφιστάμενο πρόγραμμα
 - 6.3 Πρόγραμμα μετατροπής
 - 6.4 Δημιουργία μοντέλου
 - 6.5 Ιδανικές φωτογραφίες
 - 6.6 Προσομοιώσεις
 - 6.7 Αποτελέσματα προσομοιώσεων
-

6.1 Προσέγγιση

Έχουμε αποφασίσει να δημιουργούμε πρώτα μοντέλα για τα αντικείμενα που αποτελούν το κτήριο και έπειτα να χρησιμοποιούμε αυτά που βρήκαμε, για να δημιουργήσουμε το κτήριο. Σε αυτό το σημείο της έρευνας θα περνούμε σαν είσοδο του προγράμματος ένα αντικείμενο κτηρίου (παράθυρο, πόρτα κλπ) και θα προσπαθήσουμε να δημιουργήσουμε ένα μοντέλο του.

Η ιδέα προήλθε από το γεγονός ότι, για παράδειγμα τα παράθυρα αποτελούνται κυρίως από κομμάτια γυαλιού τα οποία έχουν ένα συγκεκριμένο σχήμα (πχ. ορθογώνιο). Το πρόβλημα που παρουσιάζεται είναι πώς θα εντοπίζει ο αλγόριθμος αυτά τα σχήματα από μια έγχρωμη φωτογραφία ενός παραθύρου. Η ιδέα είναι ότι το γυαλί έχει συνήθως μια απόχρωση. Είναι ανοιχτόχρωμο, σκουρόχρωμο ή έχει αποχρώσεις κοντά σε ένα χρώμα (πχ βυσσίσι). Αποφασίσαμε να προσπαθήσουμε να εντοπίσουμε τα σχήματα του παραθύρου με βάση αυτή την λογική. Για παράδειγμα η πόρτα στη φωτογραφία 6.1 είναι κόκκινη και έχει ένα άσπρο ορθογώνιο παράθυρο. Δίνοντας σαν είσοδο του προγράμματος ένα

εικονοστοιχείο από το κόκκινο κομμάτι της πόρτας, θα μπορεί να βρει το κομμάτι της πόρτας που είναι κόκκινο και να το κάνει άσπρο, και να κάνει το υπόλοιπο μαύρο.



Φωτογραφία 6.1

Έτσι δημιουργείται μια εικόνα που μοιάζει με τις εικόνες που χρησιμοποιούσαμε ως τώρα στον γενετικό αλγόριθμο (ένα μαύρο ορθογώνιο σε λευκό φόντο).

6.2 Αλλαγές στο υφιστάμενο πρόγραμμα

Για την μετατροπή μιας εικόνας αντικειμένου στο είδος των εικόνων που χρησιμοποιούμε ως τώρα, υλοποίησα ένα νέο pixel Shader. Αυτό το pixel Shader στην ουσία μετατρέπει την έγχρωμη εικόνα σε μαυρόασπρη, όπου το άσπρο μέρος της εικόνας θα είναι το φόντο και τα μαύρα μέρη θα αντιπροσωπεύουν τα αντικείμενα του κτηρίου της εικόνας.

Στο υφιστάμενο πρόγραμμα προστέθηκε ακόμα ένα frame buffer object, το οποίο χρησιμοποιείται για να ζωγραφιστεί το αποτέλεσμα του νέου pixel Shader σε texture. Επίσης έχει τροποποιηθεί η συνάρτηση που διαβάζει την εικόνα και την μετατρέπει σε texture, έτσι ώστε να της προσθέτει το alpha. Τέλος, έχει υλοποιηθεί ο κώδικας για δημιουργία ενός μοντέλου βάση του αποτελέσματος του γενετικού αλγόριθμου.

6.3 Πρόγραμμα μετατροπής

Αυτό το πρόγραμμα παίρνει σαν είσοδο ένα texture που περιέχει την έγχρωμη εικόνα του αντικειμένου και ένα εικονοστοιχείο το οποίο επέλεξε ο χρήστης ότι θα

είναι το χρώμα του φόντου. Στο παράδειγμα της φωτογραφίας 6.1, το εικονοστοιχείο επιλέχτηκε από την περιοχή του κόκκινου.

Το πρόγραμμα ελέγχει την τιμή του εικονοστοιχείου που για το οποίο θα αποφασίσει και αν είναι μια τιμή κοντά στο χρώμα του εικονοστοιχείου που επιλέχτηκε, θέτει το εικονοστοιχείο εξόδου σε λευκό. Σε αντίθετη περίπτωση, το θέτει σε μαύρο. Η τιμή ελέγχεται λαμβάνοντας υπόψη και τα τρία χρώματα που αποτελούν ένα εικονοστοιχείο (κόκκινο, πράσινο, μπλε). Το πόσο κοντά είναι το χρώμα αποφασίζεται από ένα όριο το οποίο τίθεται, με το οποίο δοκιμάζεται η φωτογραφία για να βρεθεί το καλύτερο αποτέλεσμα. Ο υπολογισμός της τιμής που θα συγκριθεί με αυτό το όριο, αν θεωρήσουμε ότι `redP`, `greenP`, `blueP` είναι οι τιμές του επιλεγμένου εικονοστοιχείου και `texColorR`, `texColorG`, `texColorB` τιμές του εικονοστοιχείου προς σύγκριση, γίνεται ως εξής:

$$(\text{redP} - \text{texColorR})^2 + (\text{greenP} - \text{texColorG})^2 + (\text{blueP} - \text{texColorB})^2$$

Το αποτέλεσμα αυτού του υπολογισμού συγκρίνεται με την τιμή ορίου (`threshold`), για να αποφασιστεί τι χρώμα θα πάρει το εικονοστοιχείο.

6.4 Δημιουργία μοντέλου

Αφού ο γενετικός αλγόριθμος εντοπίσει τα μαύρα αντικείμενα, πρέπει να βάση αυτών να δημιουργήσει το τρισδιάστατο μοντέλο. Τα αντικείμενα των κτηρίων, είναι συνήθως πιο μέσα από τον εξωτερικό τοίχο. Για παράδειγμα ένα παράθυρο σε μια πόρτα είναι ελάχιστα πιο μέσα από το επίπεδο της πόρτας. Με αυτή την λογική, όταν θα ζωγραφιστεί το παράθυρο θα πρέπει να ζωγραφιστεί λίγο πιο μέσα από την πόρτα.

Ξέροντας το μέγεθος του αντικειμένου που θέλουμε να δημιουργήσουμε (πχ πόρτα), ζωγραφίζουμε την πόρτα χαρτογραφώντας το `texture` της φωτογραφίας πάνω της. Όταν το αντικείμενο της πόρτας ζωγραφιστεί λίγο πιο πίσω της, δεν θα φαίνεται αφού θα καλύπτεται από αυτή. Με αυτή την λογική πριν να ζωγραφιστεί η πόρτα, μετατρέπουμε το μέρος του `texture` της πόρτας το οποίο θα καλύπτει το παράθυρο σε διαφανές (κάνουμε το `alpha` 0). Έτσι όταν ζωγραφιστεί το παράθυρο φαίνεται μέσα από το διαφανές κομμάτι της πόρτας. Όταν ζωγραφιστεί το παράθυρο, χαρτογραφείται πάνω του το κομμάτι της πόρτας που του αντιστοιχεί.

6.5 Ιδανικές φωτογραφίες

Οι φωτογραφίες αντικειμένων που θα είναι ευκολότερο να λειτουργήσει το πρόγραμμα, είναι φωτογραφίες που έχουν πρωτεύοντα σχήματα όπου το χρώμα είναι κατά το πλείστο ομοιόμορφο σε όλη την επιφάνεια τους. Οι φωτογραφίες πρέπει να είναι παράλληλες προς το αντικείμενο και να βλέπουν κάθετα προς αυτό.

6.6 Προσομοιώσεις

Σε αυτές τις προσομοιώσεις άλλοι παράγοντες που επηρεάζουν το αποτέλεσμα είναι το βάθος στο οποίο θα τοποθετηθεί το παράθυρο και η τιμή ορίου, σύμφωνα με την οποία θα καθορίζεται το πόσο κοντά στο επιλεγμένο χρώμα είναι κάθε εικονοστοιχείο.

Το μαύρο πλαίσιο το οποίο φαίνεται στα αποτελέσματα των πιο κάτω προσομοιώσεων αντιπροσωπεύει το βάθος στο οποίο βρίσκεται το αντικείμενο (παράθυρο). Κανονικά έπρεπε να χαρτογραφηθεί πάνω του το κόκκινο πλαίσιο του παραθύρου αλλά λόγω χρονικών περιορισμών δεν έγινε.

6.6.1 Ευκρίνεια 128x300

Εικόνα εισόδου



Ενδιάμεση μαυρόασπρη εικόνα για τιμή ορίου 0,3:



Ενδιάμεση μαυρόασπρη εικόνα για τιμή ορίου 0,25:



Ενδιάμεση μαυρόασπρη εικόνα για τιμή ορίου 0,2:



Πρώτη προσομοίωση

Πληθυσμός: 500

Γενιές: 100

Χρόνος εκτέλεσης: 5 λεπτά και 53 δευτερόλεπτα

Βάθος: 0,05

Τιμή ορίου: 0,3

Ενδιάμεση μαυρόασπρη εικόνα:

Η εικόνα αποτέλεσμα είναι:



Δεύτερη προσομοίωση

Πληθυσμός: 1000

Γενιές: 200

Χρόνος εκτέλεσης: 11 λεπτά και 48 δευτερόλεπτα

Βάθος: 0,05

Τιμή ορίου: 0,3

Η εικόνα αποτέλεσμα είναι:



Τρίτη προσομοίωση

Πληθυσμός: 500

Γενιές: 100

Χρόνος εκτέλεσης: 7 λεπτά και 13 δευτερόλεπτα

Βάθος: 0,05

Τιμή ορίου: 0,25

Η εικόνα αποτέλεσμα είναι:



Τέταρτη προσομοίωση

Πληθυσμός: 500

Γενιές: 100

Χρόνος εκτέλεσης: 7 λεπτά και 20 δευτερόλεπτα

Βάθος: 0,07

Τιμή ορίου: 0,25

Η εικόνα αποτέλεσμα είναι:



Πέμπτη προσομοίωση

Πληθυσμός: 500

Γενιές: 100

Χρόνος εκτέλεσης: 5 λεπτά και 13 δευτερόλεπτα

Βάθος: 0,05

Τιμή ορίου: 0,2

Η εικόνα αποτέλεσμα είναι:



Έκτη προσομοίωση

Πληθυσμός: 1000

Γενιές: 100

Χρόνος εκτέλεσης: 10 λεπτά και 55 δευτερόλεπτα

Βάθος: 0,05

Τιμή ορίου: 0,2

Η εικόνα αποτέλεσμα είναι:



6.7 Αποτελέσματα προσομοιώσεων

Παρατηρούμε ότι στις δύο πρώτες προσομοιώσεις έχουμε το ίδιο αποτέλεσμα παρόλο που δίνουμε μεγαλύτερο μέγεθος πληθυσμού και αριθμό γενιών. Αυτό συμβαίνει γιατί η πόρτα έχει μόνο ένα αντικείμενο και αφού θέτουμε τον αριθμό των αντικειμένων πριν να εκτελεστεί το πρόγραμμα, το μέγεθος των χρωμοσωμάτων είναι μικρό. Έτσι δεν χρειάζεται μεγάλο μέγεθος πληθυσμού για να βρει ένα αρκετά σωστό αποτέλεσμα.

Στις επόμενες προσομοιώσεις, ελέγχουμε πόσο επηρεάζουν το αποτέλεσμα οι τιμές ορίου και βάθους. Όσον αφορά το όριο, η διαφορά που βλέπουμε στην ενδιάμεση μαυρόασπρη εικόνα, είναι ότι όσο πιο μικρό είναι το όριο τόσο πιο μικρό είναι και το σύνολο των εικονοστοιχείων τα οποία αποδέχεται ότι είναι κοντά στο κόκκινο και τους δίνει λευκό χρώμα. Στο όριο 0,3 το σχήμα είναι μεν πιο ξεκάθαρο, όμως είναι λίγο πιο μικρό στο περιθώριο και κάποιο μέρος του παραθύρου γίνεται λευκό. Στο 0,2 το σχήμα του παραθύρου είναι πιο σωστό (πιο πολλά μαύρα εικονοστοιχεία σε αυτό), υπάρχουν όμως και άλλες μαύρες λάθος περιοχές.

Στην τρίτη προσομοίωση το παράθυρο εμφανίζεται να είναι ελάχιστα πιο ορθό. Αυτό συμβαίνει λόγω της αλλαγής της τιμής ορίου. Στην τέταρτη

προσομοίωση παρατηρούμε το μαύρο πλαίσιο να είναι πιο μεγάλο, λόγω του ότι βρίσκεται σε μεγαλύτερο βάθος. Η οπτική διαφορά είναι εμφανής.

Στις τελευταίες δύο προσομοιώσεις η τιμή ορίου έχει τεθεί στο 0,2. Τα αποτελέσματα είναι ελάχιστα πιο κοντά στην λύση σε σχέση με τις προηγούμενες προσομοιώσεις με καλύτερο αποτέλεσμα αυτό της έκτης προσομοίωσης.

Σημαντική παρατήρηση είναι οι μικροί σχετικά χρόνοι για δημιουργία του τρισδιάστατου μοντέλου της πόρτας. Σε σχέση με τα προηγούμενα πειράματα το μέγεθος των χρωματοσωμάτων είναι πολύ μικρότερο, καθώς εδώ μόνο ένα αντικείμενο αναζητείται σε εκείνα ήταν τέσσερα.

Κεφάλαιο 7

Πηγές και Εργαλεία

7.1 Πηγές πληροφορίας

7.2 Εργαλεία που χρησιμοποιήθηκαν

7.1 Πηγές πληροφορίας

Γύρω από το θέμα των γενετικών αλγορίθμων υπάρχει σήμερα ένας μεγάλος αριθμός βιβλίων και ιστοσελίδων με χρήσιμο υλικό για το θεωρητικό αλλά και το πρακτικό μέρος τους. Το βιβλίο το οποίο μελέτησα περισσότερο για τον γενετικό προγραμματισμό είναι το “Genetic Programming: On the Programming of Computers by Means of Natural Selection” του John R. Koza [1]. Άλλες πηγές που με βοήθησαν στην μελέτη των γενετικών αλγορίθμων είναι οι ιστοσελίδες www.genetic-programming.org [7], www.genetic-programming.com [8], Introduction to genetic programming with Java applets [9] και το άρθρο Genetic Algorithms in Search and Optimization του Richard Baker [3]. Για την έρευνα στις διάφορες βιβλιοθήκες γενετικών αλγορίθμων, που είναι οι GAGS, GALib, GALLOPS και GAUL[4,5,6,2], βασίστηκα στις πληροφορίες που πήρα από τις ιστοσελίδες τους.

Πληροφορίες για άλλα προϊόντα τα οποία δημιουργούν τρισδιάστατα μοντέλα κτηρίων με χρήση φωτογραφιών βρήκα στις ιστοσελίδες των photo3D και photoModeler [14,15].

Το δεύτερο κομμάτι της εργασίας αυτής απαιτούσε κάποια έρευνα σε νέους τομείς των γραφικών πραγματικού χρόνου και σε αυτό με βοήθησε το βιβλίο “The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics” των

Randima Fernando και Mark J. Kilgard [8]. Άλλη χρήσιμη βοήθεια ήταν η ιστοσελίδα της OpenGL www.opengl.org [9], το βιβλίο OpenGL Programming Guide: The Official Guide to Learning OpenGL [10], το Gg Toolkit User Guide [11] και η ιστοσελίδα <http://www.codesampler.com/> [12]. Για τις επεκτάσεις OpenGL που χρησιμοποιήσα, δηλαδή τα frame buffer objects και τα occlusion queries χρήσιμη βοήθεια βρήκα στο www.opengl.org [9] και στο άρθρο Moving Beyond OpenGL 1.1 for Windows της ιστοσελίδας <http://www.gamedev.net/reference/articles/article1929.asp> [13]. Επίσης στα κείμενα προδιαγραφών τους [16,17].

7.2 Εργαλεία που χρησιμοποιήθηκαν

Για την υλοποίηση του προγράμματος χρησιμοποιήθηκε το περιβάλλον Microsoft Visual C++ .net 2003 και η υλοποίηση έγινε σε γλώσσα C.

Η υλοποίηση του γενετικού αλγόριθμου έγινε με την βοήθεια της βιβλιοθήκης Gaul : Genetic Algorithm Utility Library.

Για το γραφικό κομμάτι της υλοποίησης χρησιμοποιήθηκε η γλώσσα OpenGL και η γλώσσα Cg με την χρήση του NVIDIA Cg Toolkit. Σε κάποια σημεία του προγράμματος χρησιμοποιήθηκαν οι επεκτάσεις της OpenGL Frame Buffer Object και ARB occlusion query.

Άλλα εργαλεία τα οποία χρησιμοποιήθηκαν σε διάφορες φάσεις αυτής της εργασίας είναι το IrfanView και Microsoft Paint. Για την συγγραφή της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκε το εργαλείο Microsoft Word.

Κεφάλαιο 8

Συμπεράσματα

8.1 Ανασκόπηση των στόχων

8.2 Μελλοντική έρευνα

8.3 Τελικά Συμπεράσματα

8.1 Ανασκόπηση των στόχων

Ξεκινώντας αυτή την εργασία, έχουν τεθεί κάποιοι στόχοι τους οποίους επιδιώξα να πετύχω στην περίοδο εκπόνησης της διπλωματικής μου εργασίας. Αυτοί οι στόχοι έχουν ολοκληρωθεί με κάποιο βαθμό επιτυχίας και θα εξεταστούν πιο κάτω:

Έγινε μελέτη γενετικών αλγορίθμων για κατανόηση τους έτσι ώστε να δημιουργηθεί ένας για ανάπτυξη του προγράμματος. Έχει γίνει η υλοποίηση του γενετικού αλγόριθμου και δύο συναρτήσεων αξιολόγησης από τις οποίες επιλέχτηκε μια, έτσι αυτός ο στόχος έχει επιτευχθεί.

Έχει καθοριστεί ένας τύπος κτηρίων και φωτογραφιών με τις οποίες θα επεξεργάζεται το πρόγραμμα για να δημιουργήσει το μοντέλο.

Η δημιουργία μιας τρισδιάστατης πλευράς κτηρίου με χρήση μιας φωτογραφίας δεν έχει επιτευχθεί. Έχει όμως δημιουργηθεί τρισδιάστατο μοντέλο αντικειμένου ενός κτηρίου, του οποίου εισάγεται η φωτογραφία στο πρόγραμμα.

8.2 Μελλοντική έρευνα

Η ανάπτυξη του προγράμματος αυτού δεν έχει ολοκληρωθεί. Οι κύριοι λόγοι της μη ολοκλήρωσης του είναι ότι ο χρόνος ολοκλήρωσης της διπλωματικής εργασίας δεν ήταν αρκετός και ο στόχος του προγράμματος δεν είναι απλός. Παρ' όλα αυτά

το στάδιο στο οποίο έφθασε, μας δίνει ενθαρρυντικά αποτελέσματα για την συνέχεια. Για επέκταση του υφιστάμενου προγράμματος προτείνονται:

1. Διεξαγωγή περισσότερων πειραμάτων για δημιουργία τρισδιάστατων μοντέλων αντικειμένων κτηρίων, όπως εργαστήκαμε στο κεφάλαιο 6.
2. Επέκταση για δημιουργία πιο πολύπλοκων αντικειμένων.
3. Είσοδος περισσότερων φωτογραφιών αντικειμένων και δημιουργία περισσότερων μοντέλων σε μια εκτέλεση του προγράμματος.
4. Δημιουργία τρισδιάστατης πλευράς κτηρίου με απλά αντικείμενα όπως την περίπτωση της πόρτας που δημιουργήθηκε.
5. Δημιουργία βιβλιοθήκης με τα αντικείμενα που δημιουργήθηκαν.
6. Δημιουργία τρισδιάστατης πλευράς κτηρίου με αντικείμενα που δημιουργήθηκαν από άλλες φωτογραφίες.
7. Συνδυασμός των 4 και 6.
8. Δημιουργία μοντέλου κτηρίου χρησιμοποιώντας περισσότερες από μια φωτογραφίες του, συνδυάζοντας τις τρισδιάστατες πλευρές του που θα δημιουργηθούν.
9. Δυνατότητα αποθήκευσης του παραγόμενου μοντέλου για χρήση σε άλλες εφαρμογές.

8.3 Τελικά Συμπεράσματα

Η συγκεκριμένη έρευνα έχει αρκετό ενδιαφέρον. Η χρήση ενός γενετικού αλγόριθμου για δημιουργία ενός τρισδιάστατου μοντέλου δεν έχει μελετηθεί στο παρελθόν. Με διάφορες μεθόδους φτάσαμε στο σημείο να μπορούμε να δημιουργήσουμε το μοντέλο ενός απλού αντικειμένου κτηρίου. Τα αποτελέσματα είναι αρκετά ενθαρρυντικά όμως θα πρέπει να γίνουν και άλλες δοκιμές για να επιτευχθούν καλύτερα αποτελέσματα. Η συνέχεια της ανάπτυξης αυτού του προγράμματος θα καθορίσει μπορεί να επεκταθεί στην δημιουργία ενός μοντέλου ενός ολόκληρου κτηρίου και αν αυτό θα είναι ικανοποιητικό για χρήση σε σημερινές εφαρμογές γραφικών.

Βιβλιογραφία

[1] John R. Koza, "Genetic Programming: On the Programming of Computers by Means of Natural Selection", December 11, 1992

[2] Gaul: Genetic Algorithm Utility Library
<http://gaul.sourceforge.net/>

[3] Genetic Algorithms in Search and Optimization
By Richard Baker
<http://www.fenews.com/fen5/ga.html>

[4] GAGS genetic algorithm C++ class library
<http://kal-el.ugr.es/GAGS/newGAGS.html>

[5] GALib A C++ Library of Genetic Algorithm Components
<http://lancet.mit.edu/ga/>

[6] GALOPPS 3.2.4 - the "Genetic ALgorithm Optimized for Portability and Parallelism System"

[7] <http://www.genetic-programming.org/>

[8] <http://www.genetic-programming.com/>

[9] Inroduction to genetic programming with Java applets
<http://cs.felk.cvut.cz/~xobitko/ga/>

[8] Randima Fernando and Mark J. Kilgard, "The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics", 2003

[9] www.opengl.org

[10] OpenGL Architecture Review Board, “OpenGL Programming Guide: The Official Guide to Learning OpenGL”, 2003

[11] Gg Toolkit User Guide

[12] <http://www.codesampler.com/>

[13] Dave Astle, “Moving Beyond OpenGL 1.1 for windows”, Premier Press
<http://www.gamedev.net/reference/articles/article1929.asp>

[14] <http://www.photo3d.com/eindex.html>

[15] <http://www.photomodeler.com/index.html>

[16] Framebuffer object specification

[17] ARB occlusion query specification

Παράρτημα

Cg Pixel Shader πρόγραμμα για σύγκριση δύο εικόνων. Το αποτέλεσμα είναι μια μαυρόασπρη εικόνα στην οποία το λευκό αντιπροσωπεύει τα κοινά εικονοστοιχεία και το μαύρο εκείνα που διαφέρουν.

```
struct fragment
{
    float4 position    : POSITION;
    float4 color0      : COLOR0;
    float2 texcoord0   : TEXCOORD0;
};

struct pixel
{
    float4 color : COLOR;
};

//-----
// IN          - incoming fragment to be processed
// tex1        - a texture for the pixel shader to use
// tex2        - a second texture for the pixel shader to use
//-----

pixel compTex( fragment IN,

// Do this if you're going to use Cg's method of passing the multi-
// texture ID.
    uniform sampler2D tex1,
    uniform sampler2D tex2 )
{
    pixel OUT;

    float4 texColor0 = tex2D( tex1, IN.texcoord0 );
    float4 texColor1 = tex2D( tex2, IN.texcoord0 );

    if( texColor0[1] == texColor1[1] && texColor0[2] == texColor1[2] &&
        texColor0[3] == texColor1[3]) //&& texColor0[3] == texColor1[3] )
        OUT.color=float4(1,1,1,1);
    else
        OUT.color=float4(0,0,0,0);

    return OUT;
}
```

Cg Pixel Shader για μετατροπή μιας έγχρωμης εικόνας σε μαυρόασπρη. Τα RED_PIXEL, GREEN_PIXEL και BLUE_PIXEL αντιπροσωπεύουν τα στοιχεία RGB του εικονοστοιχείου της εικόνας που επιλέχτηκε για φόντο.

```
/red door
#define RED_PIXEL 0.79
#define GREEN_PIXEL 0.31
#define BLUE_PIXEL 0.4

struct fragment
{
    float4 position : POSITION;
    float4 color0 : COLOR0;
    float2 texcoord0 : TEXCOORD0;
};

struct pixel
{
    float4 color : COLOR;
};

//-----
// IN - incoming fragment to be processed
// tex1 - a texture for the pixel shader to use
//-----
pixel convertTex( fragment IN,
// Do this if you're going to use Cg's method of passing the multi-
texture ID.
    uniform sampler2D tex1)//, float4 rgbColor)
{
    pixel OUT;

    float4 texColor0 = tex2D( tex1, IN.texcoord0 );
    float threshold=0.2;
    float value=(RED_PIXEL-texColor0[0])*(RED_PIXEL-
texColor0[0])+(GREEN_PIXEL-texColor0[1])*(GREEN_PIXEL-
texColor0[1])+(BLUE_PIXEL-texColor0[2])*(BLUE_PIXEL-texColor0[2]);
    if(value<=threshold)
    {
        OUT.color=float4(1,1,1,1);//white
    }else
    {
        OUT.color=float4(0,0,0,1);
    }
    return OUT;
}
```