

Surface reconstruction methods

Φίλιππος Ρωσσίδης

18 Νοεμβρίου 2025

1 Προγράμματα

Χρησιμοποιώντας:

- **MeshLab** για εκτίμηση normals, και screened Poisson surface reconstruction.
- **CGAL** για Kinetic, Advancing front
- **Blender** για δημιουργία και επεξεργασία δεδομένων.

2 Θεωρητικά: καλή και κακή τοπολογία στη γραφική

Μια σύντομη αναφορά στην καλή τοπολογία στο context της γραφικής, διότι συζητήσαμε για αυτό.

Ορολογία:

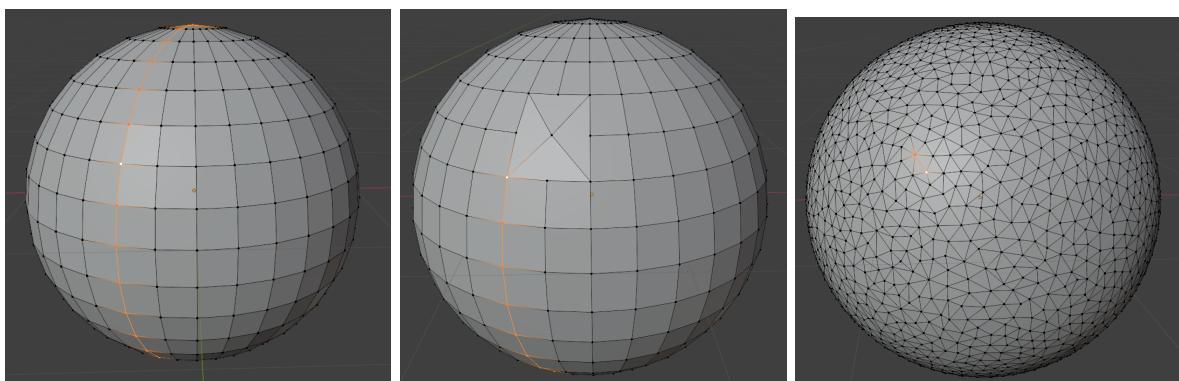
- **Quads:** τετράπλευρα
- **Ngons:** πολύγωνα με (συνήθως) παραπάνω από 4 πλευρές

Θέλουμε τα meshes να αποτελούνται από quads, και όχι τρίγωνα, ούτε ngons.

Αυτό διότι στα quads (αν τοποθετηθούν σωστά) μπορούμε να εντοπίσουμε edge flow, όπως π.χ. είναι χρωματισμένο με πορτοκαλί για 3 περιπτώσεις στο σχήμα 1. Στο αριστερά μπορούμε να βρούμε συνεχόμενες ροές ακμών σε όλο το μήκος της σφαιράς, στο μεσαίο (παρότι αποτελείται από quads) υπάρχει ένα ‘εμπόδιο’, στο αριστερό επειδή η επιφάνεια αποτελείται από τρίγωνα, δεν μπορεί πουθενά να εντοπιστεί ροή.

Ο λόγος που θέλουμε να υπάρχει ροή στις ακμές είναι γιατί το mesh θα χρησιμοποιηθεί στη συνέχεια σε διάφορα pipelines που δεν λειτουργούν καλά χωρίς αυτήν.

Παραδείγματος χάριν, αν θέλουμε να κάνουμε animate το mesh (δηλαδή να του δώσουμε κίνηση) σκεφτείτε π.χ. τα βλέφαρα ενός χαρακτήρα τα οποία θέλουμε να ανοιγοκλείνουν, θα πρέπει να υπάρχει

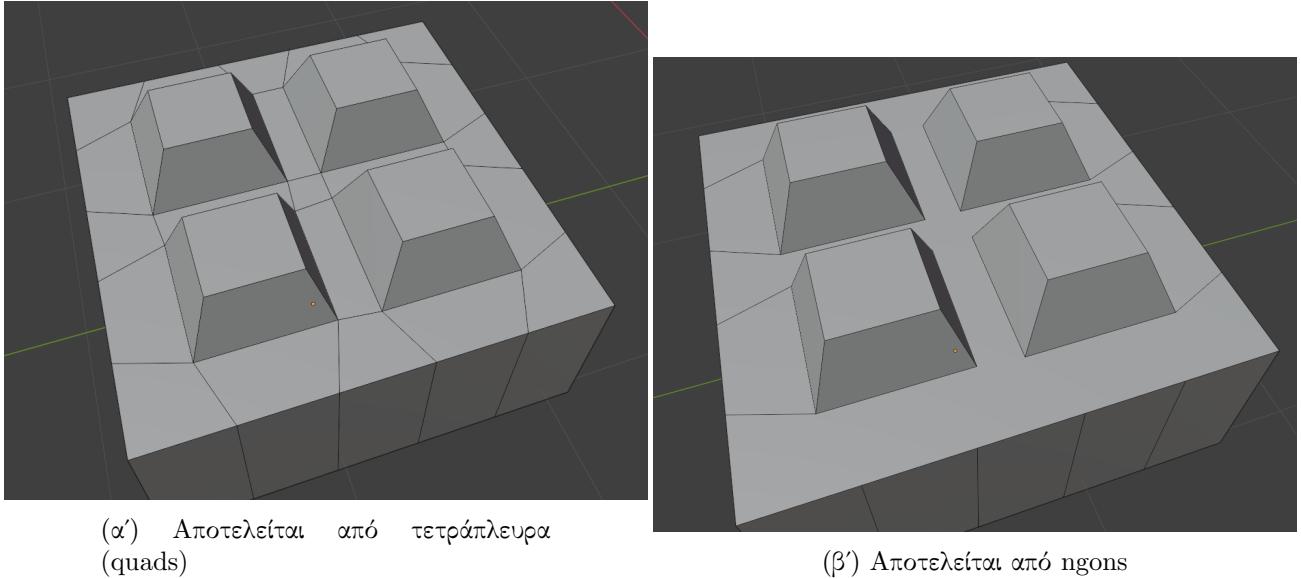


(α') Ο καλός

(β') ο κακός

(γ') και ο άσχημος

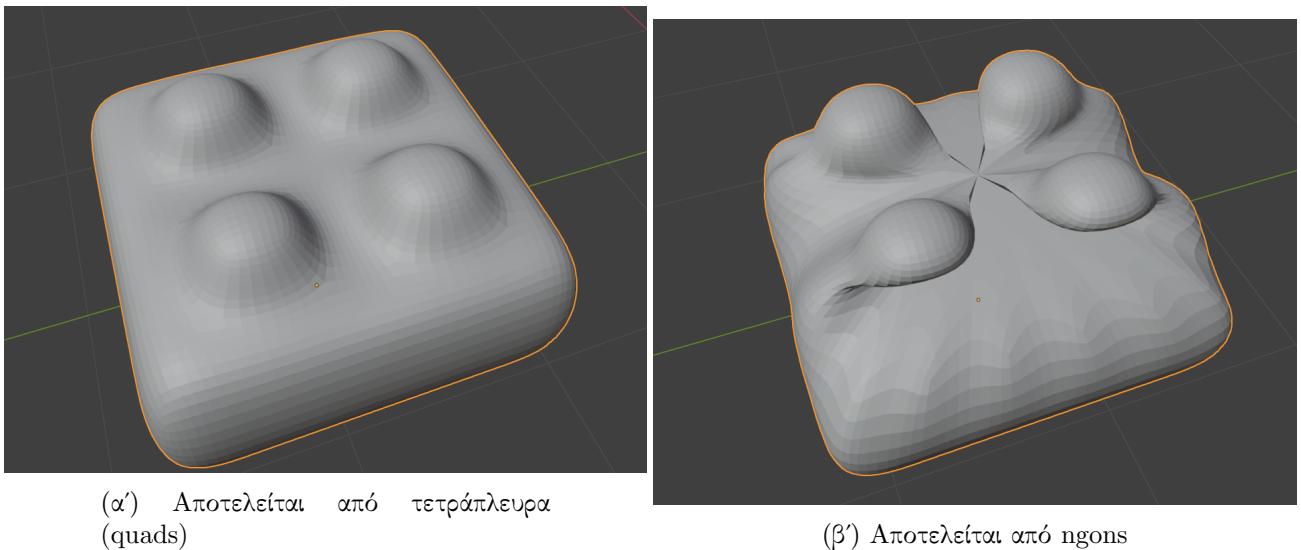
Σχήμα 1: Σύγχριση τοπολογίας



(α') Αποτελείται από τετράπλευρα
(quads)

(β') Αποτελείται από ngons

Σχήμα 2: Σύγκριση τοπολογίας πριν το subdivision



(α') Αποτελείται από τετράπλευρα
(quads)

(β') Αποτελείται από ngons

Σχήμα 3: Σύγκριση τοπολογίας μετά το subdivision

ξεκάθαρη ροή των βλεφάρων γύρω από το μάτι, ώστε να μπορούν να κινηθούν χωρίς να παραμορφωθεί η γεωμετρία τους.

Ένας άλλος λόγος φαίνεται στα σχήματα 2, 3. Εδώ χρησιμοποιείται η τεχνική subdivision surface, όπου υποδιαιρείται η επιφάνεια σε μικρότερα πολύγωνα (γίνεται σε κάθε βήμα διπλασιασμός του πλήθους τους και interpolation ανά 3 αν δεν κάνω λάθος) για να γίνει πιο smooth το σχήμα.

Στην αριστερά τοπολογία (quads), η μεθόδος λειτουργεί κανονικά, ενώ στην δεξιά (Ngons), το interpolation αδυνατεί να λειτουργήσει αφού κάποιοι κόμβοι έχουν γείτονες μόνο προς μία κατεύθυνση.

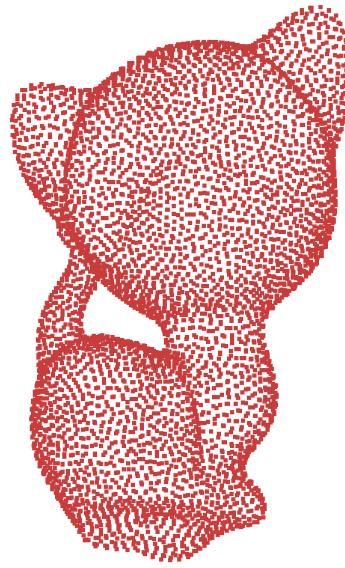
3 Δημιουργία point cloud δεδομένων

Η βιβλιοθήκη CGAL με την εγκατάστασή της παρέχει κάποια point clouds τα οποία χρησιμοποιώ για tests των μεθόδων (π.χ. σχήμα 4).

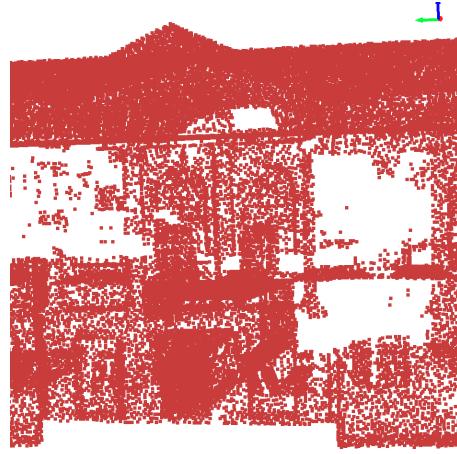
Τα συμπεριλαμβάνω στο φάκελο CGAL-6.1

Επίσης δημιουργώ δικά μου δεδομένα με χρήση του Blender:

Μοντελοποιώ ένα mesh, και με τα geometry nodes (μέθοδος procedural παραγωγής γεωμετριών) του σχήματος 5, κάνω scatter σημεία στην επιφάνεια του mesh με τυχαίο τρόπο. Μπορώ να ελέγξω την

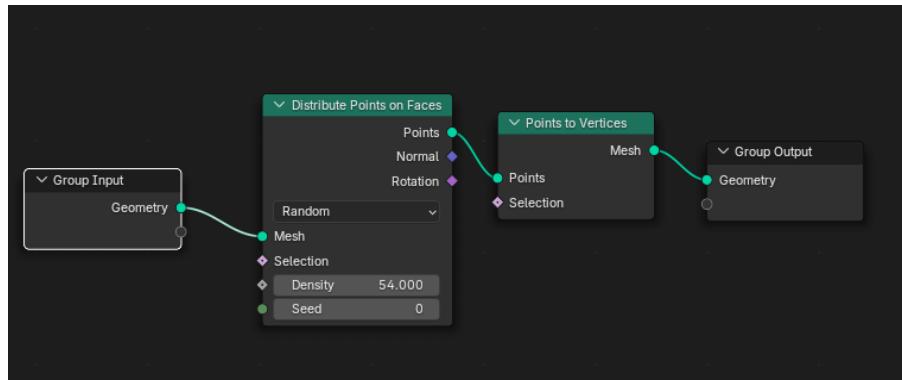


(α') Σημεία μοντέλου γάτας, χωρίς θόρυβο, με σταθερή πυκνότητα.



(β') Λεπτομερές κτήριο με θόρυβο και ελλειπή δεδομένα.

Σχήμα 4: Παραδείγματα point data της CGAL.



Σχήμα 5: Scatter points on surface

πυκνότητα με την τιμή density.

Μπορώ να το επεξεργαστώ στην πορεία για να συμπεριλάβω θόρυβο κλπ.

4 Delaunay-like methods

Μέθοδοι που χρησιμοποιούν delaunay σε κάποιο στάδιο τους. Χρησιμοποιούν ακριβώς τα σημεία του point cloud ως κόμβους του mesh.

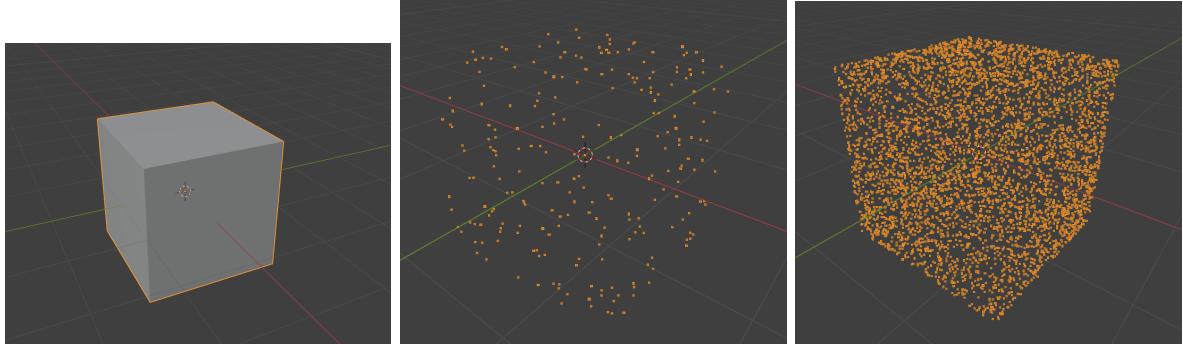
4.1 Advancing front

Προς το παρόν, δεν ασχολούμαστε με το πως λειτουργεί ο αλγόριθμος, όσο με τα αποτελέσματα που παράγει.

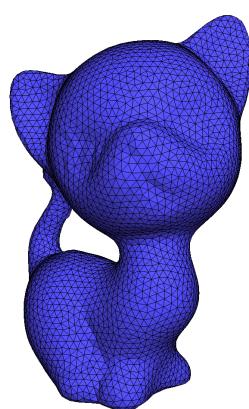
Χρησιμοποιώ τη συνάρτηση που παρέχει η βιβλιοθήκη CGAL (συμπεριλαμβάνω και τα αρχεία κώδικα). Εφαρμόζοντας στο point set 4α' παίρνουμε ένα πολύ καλό αποτέλεσμα (σχήμα 7α') επειδή τα δεδομένα είχαν σταθερή και επαρκή πυκνότητα (ούτε πολύ λεπτομερή, ούτε ελλειπή) και δεν είχαν θόρυβο.

Σε δεδομένα με μη σταθερή πυκνότητα (σχήμα 7β') το αποτέλεσμα έχει γενικά κακή τοπολογία, υπάρχουν πολύ μικρά και πολύ μεγάλα τρίγωνα, υπάρχουν μικρές γωνίες, δεν υπάρχει ομαλότητα και ροή.

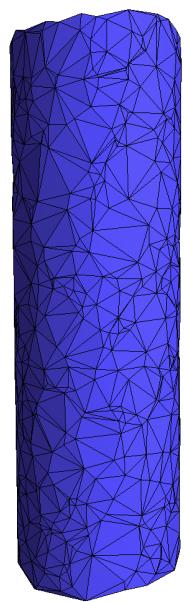
Ένα άλλο πρόβλημα αυτής της μεθόδου φαίνεται στο σχήμα 8. Στα αριστερά φαίνεται το point cloud το οποίο δεν είχε θόρυβο, στο κέντρο το mesh που δημιουργήθηκε, σε μια από τις γωνίες.



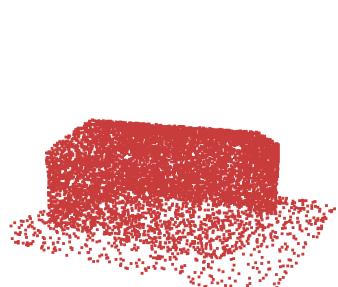
$\Sigma\chi\eta\mu\alpha 6$



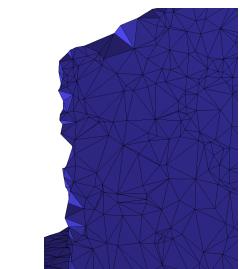
(α') kitten, Advancing front



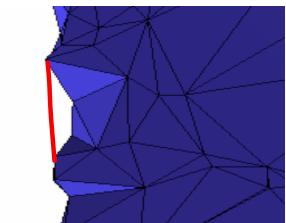
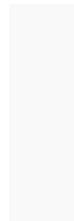
(β') cylinder, Advancing front



(α') building point set

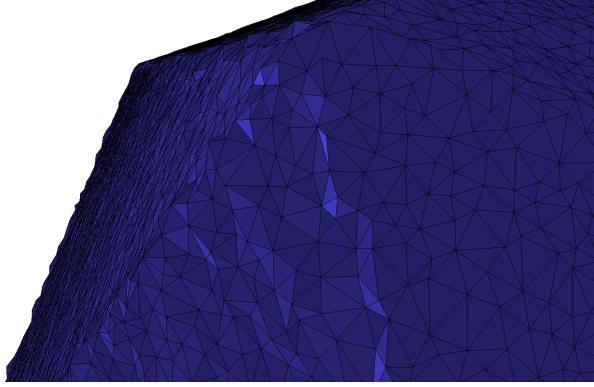


(β') Η γωνία του reconstruction από το point set.



(γ') Διόρθωση

$\Sigma\chi\eta\mu\alpha 8$



Σχήμα 9

Παρότι δεν υπήρχε θόρυβος στα δεδομένα, η γωνία εμφανίζεται ‘κομμένη’. Για να διατηρηθεί σωστά η γεωμετρία θα έπρεπε για γίνει π.χ. η σύνδεση που φαίνεται με κόκκινο στο σχήμα 8γ' και οι υπόλοιπες απαραίτητες ακμές για να την στηρίξουν.

Για μεγαλύτερη ακρίβεια, θα έπρεπε οι κόμβοι που μοντελοποιούν τη γωνία να βρίσκονται ακριβώς πάνω στην θεωρητική γωνία. Έτσι θα έπρεπε είτε να ανιχνευτούν οι κόμβοι που έχουν τις πιο εξωτερικές συντεταγμένες και να γίνουν οι ενώσεις (όπως η κόκκινη γραμμή στο σχήμα 8γ') μόνο μεταξύ τους, οπότε θα υπήρχαν δυσανάλογα μεγάλα τρίγωνα στις γωνίες, ή να δημιουργηθούν και νούργιοι κόμβοι πάνω στην γωνία.

Παρόμοιο πρόβλημα βλέπουμε στα αυτιά της γάτας (σχήμα 7α').

- Αυτό παρουσιάζει ενδιαφέρον για περαιτέρω μελέτη.

Στο σχήμα 9 είναι το αποτέλεσμα σε δεδομένα με θόρυβο. Όπως φαίνεται, δεν καταπολεμείται χωρίς προεπεξεργασία.

5 Poisson

Η μέθοδος Poisson προσπαθεί να λύσει το πρόβλημα παρουσία θορύβου, επομένως είναι απαραίτητο να δημιουργηθούν νέοι κόμβοι για το mesh, έναντι των σημείων του point cloud. Λειτουργεί για watertight κλειστά σχήματα.

Παίρνει το όνομά της διότι προσεγγίζει τη λύση μιας εξίσωσης Poisson ως εξής:

Έστω ένα αντικείμενο που χωρίζει τον χώρο σε εσωτερικό και εξωτερικό και έστω ότι γνωρίζουμε την κλίση του σε ορισμένα σημεία της επιφάνειας (έχουμε normal vectors). Ορίζουμε την συνάρτηση x ως $\frac{1}{2}$ στο εσωτερικό και $-\frac{1}{2}$ στο εξωτερικό. Τότε η κλίση της συνάρτησης είναι 0 παντού εκτός από την επιφάνεια που θέλουμε να βρούμε, όπου είναι άπειρη και δείχνει προς το εσωτερικό. Για να αποκτήσει πραγματική τιμή (ίση με την τιμή των normals) εφαρμόζουμε συνέλιξη με κάποια smoothness συνάρτηση. Θέλουμε να εκτιμήσουμε την συνάρτηση x από τα δεδομένα.

Αν \mathbf{V} είναι η πραγματική κλίση, τότε η συναρτησιακή:

$$E(x) = \int ||\nabla x(p) - \mathbf{V}(p)||^2 dp$$

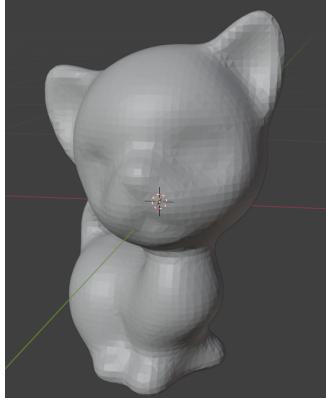
δηλώνει την απόσταση της εκτίμησης x από την επιλυμητή συνάρτηση.

Προσεγγίζουμε την συνάρτηση x ελαχιστοποιώντας τη συναρτησιακή, ή ισοδύναμα (Euler-Lagrange) λύνοντας την εξίσωση Poisson:

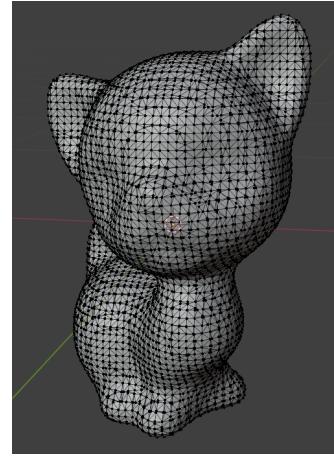
$$\Delta x = \nabla \cdot \mathbf{V}$$

Γίνεται διακριτοποίηση του χώρου και λύνεται το σύστημα που προκύπτει από τη διατύπωση Galerkin της παραπάνω διαφορικής για να εκτιμηθεί η x . Έπειτα χρησιμοποιείται η x για τη δημιουργία του mesh.

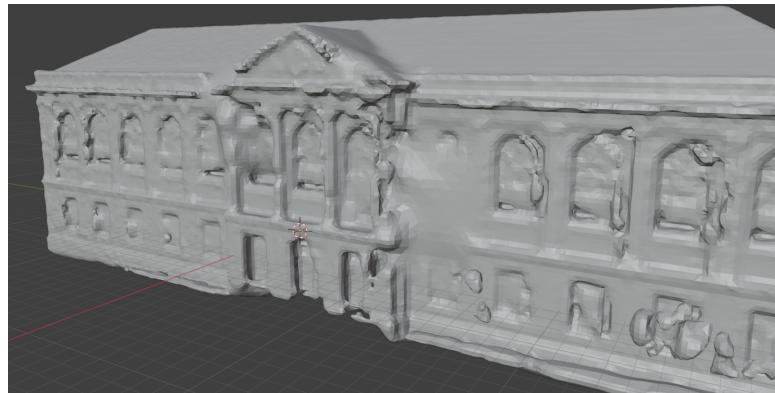
Περισσότερες πληροφορίες στο φάκελο με τα papers (Screened Poisson).



(α') γάτα, Poisson



(β') γάτα Poisson, edges



Σχήμα 11

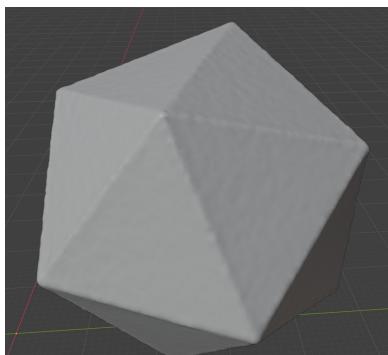
5.1 Κάποια αποτελέσματα

Εδώ χρησιμοποιώ το MeshLab για να παράγω αποτελέσματα. Τα normals δημιουργούνται από το point set με τη μέθοδο του MeshLab.

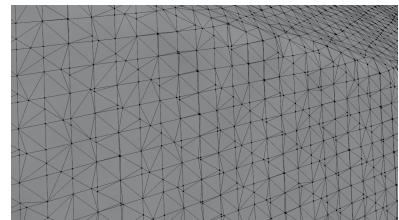
Για smooth επιφάνειες, όπως η γάτα στο σχήμα 10β' τα αποτελέσματα είναι πολύ καλά. Εδώ δεν υπάρχει το πρόβλημα που είχε η μέθοδος advancing front με τα αυτιά της γάτας. Η τοπολογία όμως του παραγόμενου mesh δεν είναι καλή (τρίγωνα με πολύ μικρές γωνίες)

Σε επιφάνειες με επίπεδα και γωνίες, όπως κτήρια (σχήμα 11) τα αποτελέσματα δεν είναι τόσο καλά. Στο κτήριο του σχήματος υπάρχουν προβλήματα λόγω ‘κακών’ δεδομένων (κενά, έντονος θόρυβος).

Σε δεδομένα με λίγο θόρυβο (σχήμα 12) η μέθοδος Poisson εξομαλύνει τον θόρυβο, αλλά κάνει επίσης smooth στις γωνίες όπου χάνεται η ακμή.

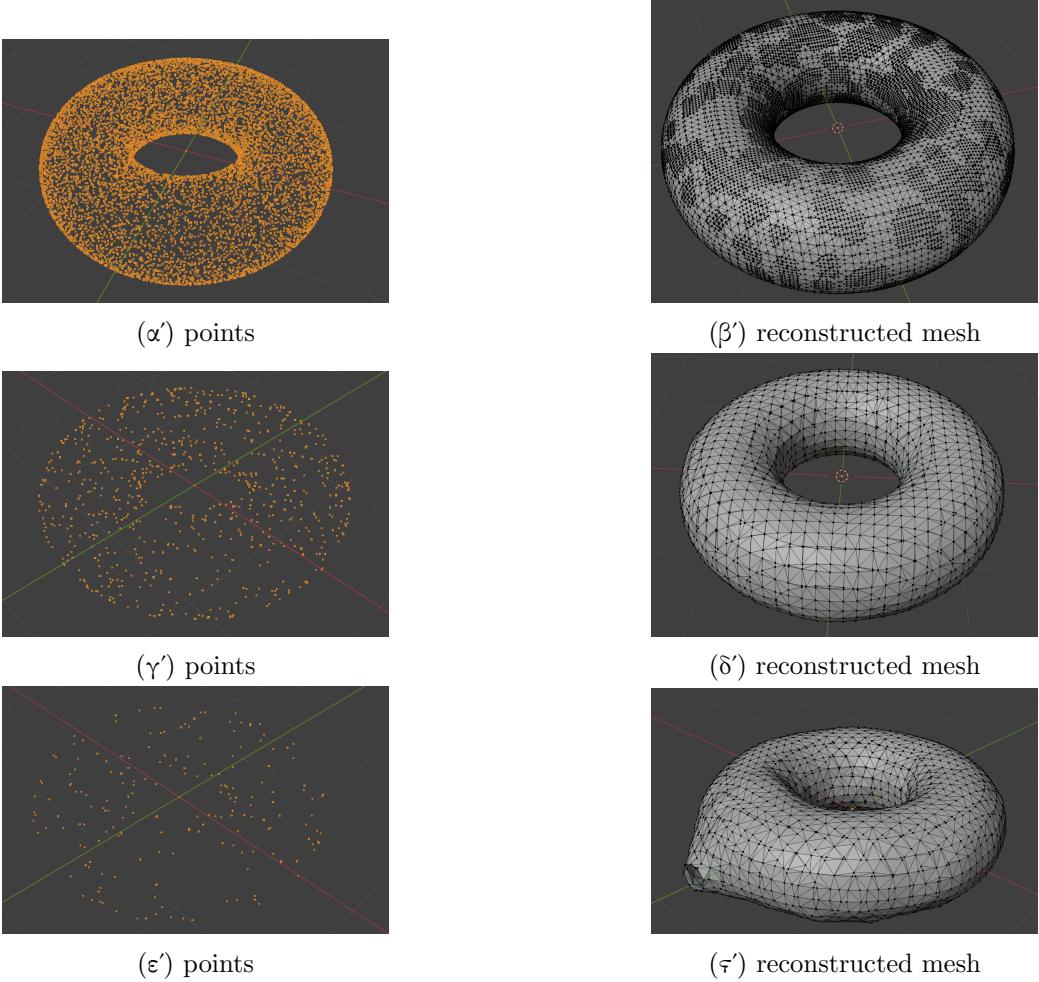


(α') reconstructed mesh



(β') Τοπολογία

Σχήμα 12



Σχήμα 13: Μέθοδος Poisson σε αντικείμενο με τρύπα (torus) για διαφορετικές πυκνότητες σημείων στα δεδομένα

Ένα πρόβλημα, επομένως, της μεθόδου είναι το πρόβλημα της διατήρησης της ακμής η οποία χάνεται λόγω της συνέλιξης που προαναφέρθηκε (περιγράφεται στο paper ...).

Στο σχήμα 14 βλέπουμε το reconstruction για το point cloud ενός δράκου. Στα scales του δράκου έχει χαθεί η ακμή που τα διαχωρίζει μεταξύ τους.

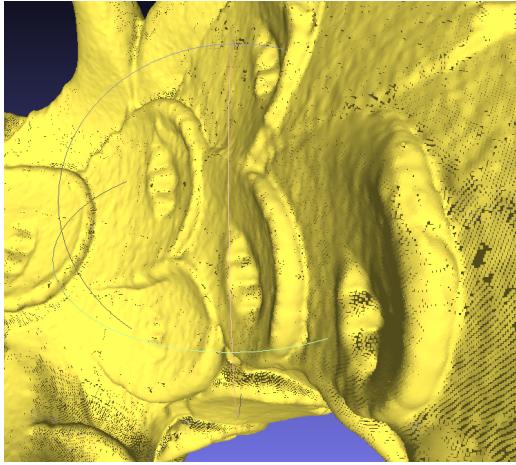
6 Kinetic

Η μέθοδος αυτή δέχεται ως είσοδο point sets, με normals που δείχνουν προς τα ίξω και ένα σύνολο επιπέδων τα οποία έχουν ανιχνευθεί από τα σημεία. Ως μηδενικό βήμα απλοποιεί κάθε ένα από τα επίπεδα ως το κυρτό περίβλημα των σημείων που βρίσκονται εντός του. Έπειτα κάνει τις απαραίτητες τομές ώστε τα επίπεδα να μην τέμνονται μεταξύ τους.

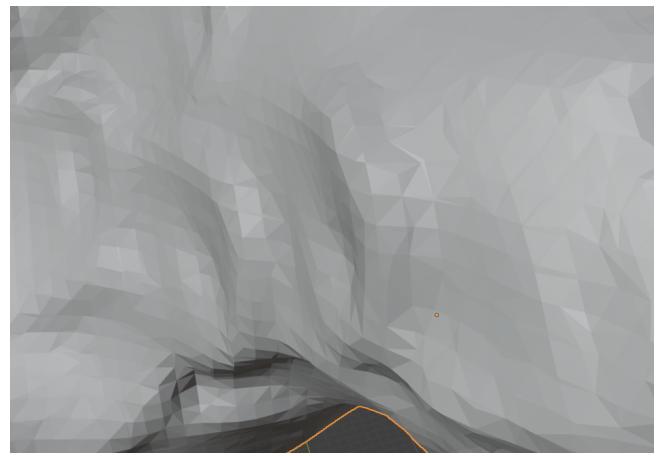
Το κύριο βήμα του αλγορίθμου (kinetic partitioning) διαμερίζει τον υπολογιστικό χώρο (bounding box των δεδομένων) σε (κυρτά) πολύεδρα μεγαλώνοντας σε κάθε βήμα κάθε (κυρτό πλέον) επίπεδο μέχρι να ανιχνεύσει σύγκρουση, όπου μπορεί είτε να σταθεροποιήσει τους κόμβους στο σημείο της σύγκρουσης, είτε να αλλάξει την κατεύθυνση της κίνησής τους, είτε να διαμερίσει τα επίπεδα. Όταν έχουν σταθεροποιηθεί πλέον όλοι οι κόμβοι, έχουμε το διαμερισμό του χώρου σε πολύεδρα.

Έπειτα ο αλγόριθμος αποφασίζει ποια πολύεδρα ανήκουν στο εσωτερικό και ποια στο εξωτερικό του αντικειμένου: Θέτει για κάθε πολύεδρο μία τιμή $x_i = \{in, out\}$ ελαχιστοποιώντας την:

$$U(\mathbf{x}) = D(\mathbf{x}) + \lambda V(\mathbf{x})$$



(α') Δεδομένα



(β') Αποτέλεσμα

Σχήμα 14: Reconstruction δράκου.

με

$$D(\mathbf{x}) = \frac{1}{|I|} \sum_{i \in C} \sum_{p \in I_i} d_i(p, x_i)$$

όπου C το σύνολο των πολυέδρων, I_i τα σημεία που βρίσκονται (inlier) στο πολύεδρο i και $d(p, x_i)$ είναι 1 όταν το normal vector του σημείου p δείχνει προς το εσωτερικό σύμφωνα με το x_i , και

$$V(\mathbf{x}) = \frac{1}{A} \sum_{i \sim j} a_{ij} \cdot 1_{\{x_i \neq x_j\}}$$

όπου με \sim συμβολίζουμε γειτονικά πολύεδρα, a_{ij} το εμβαδόν της κοινής πλευράς και A ένα normalization factor ίσο με το άνθροισμα όλων των εμβαδών της διαμέρισης.

Ο όρος $U(\mathbf{x})$ είναι ελάχιστος για την καλύτερη προσέγγιση εσωτερικού και εξωτερικού, και ο όρος $V(\mathbf{x})$ υπάρχει για να απλοποιεί το σχήμα και να αποφευχθούν τα zig-zag.

Στην περίπτωση του CGAL είναι ορισμένα ως

$$U(\mathbf{x}) = (1 - \lambda)D(\mathbf{x}) + \lambda V(\mathbf{x})$$

με

$$D(\mathbf{x}) = \sum_{i \in C} \sum_{p \in I_i} d_i(p, x_i)$$

και

$$V(\mathbf{x}) = \frac{1}{A} \sum_{i \sim j} a_{ij} \cdot 1_{\{x_i \neq x_j\}}$$

Η παραπάνω ελαχιστοποιείται με min-cut.

Αφού γίνει αυτό, επιστρέφεται το σύνολο των επιπέδων που έχουν μία πλευρά στο εσωτερικό και μια στο εξωτερικό του αντικειμένου ως το reconstructed mesh.

Η ανίχνευση των επιπέδων γίνεται με region growing, αλλά μπορεί να γίνει και με RANSAC.

Περισσότερες λεπτομέρειες μπορείτε να βρείτε στο paper (το έχω συμπεριλάβει στα αρχεία).

6.1 Παράμετροι

Shape Detection

- maximum_distance : The maximum distance of a point to a plane.
- maximum_angle : The maximum angle in degrees between the normal associated with a point and the normal of a plane.

- `minimum_region_size` : The minimum number of points a region must have.
- `k_neighbors` : defines the number K of nearest neighbors of the query point.

Shape Regularization

- `regularize_parallelism` : indicates whether parallelism should be regularized or not
- `regularize_coplanarity` : indicates whether coplanarity should be regularized or not
- `regularize_orthogonality` : indicates whether orthogonality should be regularized or not
- `regularize_axis_symmetry` : indicates whether axis symmetry should be regularized or not
- `maximum_offset` : maximum allowed orthogonal distance between two parallel planes such that they are considered to be coplanar
- `angle_tolerance` : maximum allowed angle in degrees between plane normals used for parallelism, orthogonality, and axis symmetry
- `symmetry_direction` : an axis for symmetry regularization

Kinetic Space Partition

- `k` : the maximum number of intersections that can occur for a polygon before its expansion stops
- `reorient_bbox` : Setting `reorient_bbox` to true aligns the x-axis of the bounding box with the direction of the largest variation in horizontal direction of the input data while maintaining the z-axis
- `bbox_dilation_ratio` : By default the size bounding box of the input data is increased by 10% to avoid that input polygons are coplanar with the sides of the bounding box.
- `max_octree_node_size` : A kinetic partition is split into 8 subpartitions using an octree if the number of intersecting polygons is larger than specified. The default value is 40 polygons.
- `max_octree_depth` : Limits the maximum depth of the octree decomposition. A limitation is necessary as arbitrary dense polygon configurations exist, e.g., a star. The default value is set to 3.

6.2 Αποτελέσματα

Η μέθοδος είναι ευαίσθητη στις παραμέτρους και χρειάζεται αρκετές δοκιμές για να παρθεί καλό αποτέλεσμα. Για μεγάλα datasets αυτό αποτελεί πρόβλημα κανώς μπορεί κάθε δοκιμή να τρέχει για ώρες.

Δοκίμασα κάποια δεδομένα που δόθηκαν από τον συγγραφέα που πρότεινε τη μέθοδο με τις παραμέτρους που προτείνουν στη CGAL:

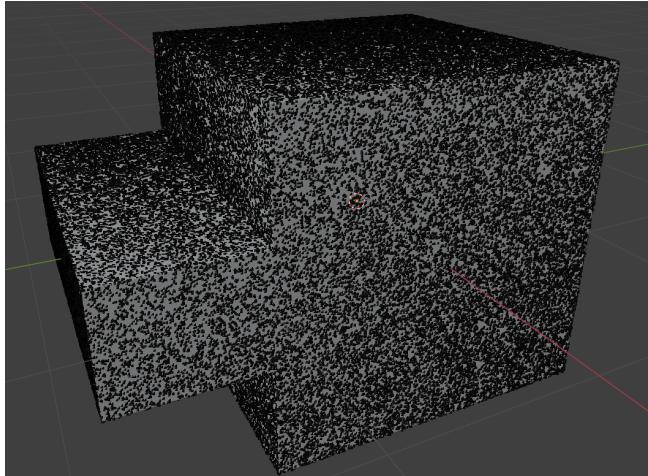
Για το παράδειγμα Hilbert_Cube (σχήμα 16) πήρα τέλεια αποτελέσματα (με μόνη εξαίρεση κάποιους αχρείαστους κόμβους σε σενευθειακά τμήματα).

Με δικά μου συνθετικά δεδομένα που παρήγαγα από τη σφαίρα του σχήματος 17 (αριστερά) παρήγαγα (στο κέντρο) με Poisson και (δεξιά) με Kinetic. Η Poisson παράγει πυκνή γεωμετρία με smoothed ακμές, ενώ η Kinetic καταφέρνει να ‘πιάσει’ κάθε πλευρά σωστά, με εξαίρεση τις πολύ μικρές που βρίσκονται στους πόλους. Παρουσιάζει πολύ καλό αποτέλεσμα, αν και είναι σημαντικά πιο αργή από την Poisson.

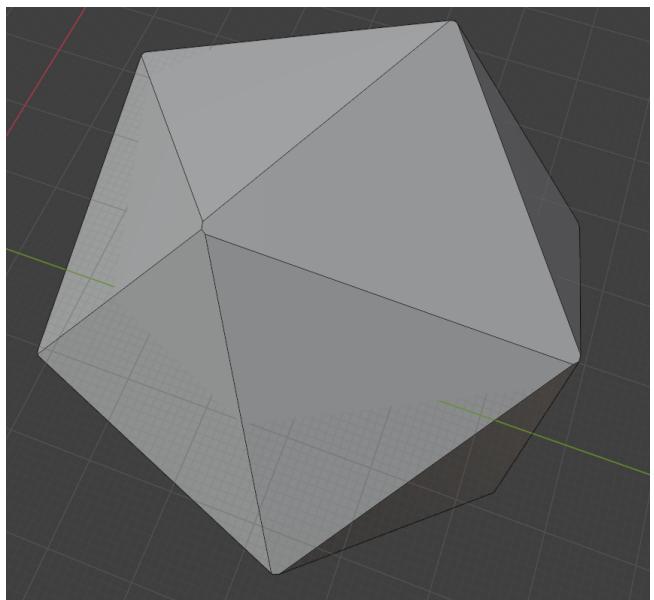
Στο σχήμα 18 φαίνεται ένα παράδειγμα με δεδομένα που περιέχουν outliers και κενά. Η μέθοδος καταφέρει να ανιχνεύσει τη γεωμετρία, αλλά με αρκετά προβλήματα. Δύο πλευρές είναι degenerate (είναι το warning που μου δίνει το meshLab, μπορεί να σημαίνει πολλά, γενικά είναι ότι δεν είναι καλώς ορισμένη), είναι οι δύο πλευρές που δεν εμφανίζονται στην αναπαράσταση (στο κέντρο).

Στο manual της cgal παίρνουν καλύτερα αποτελέσματα όλα δεν καταφέρνω να τα αναπαράγω.

Η μέθοδος φαίνεται αδύνατη να ανακατασκευάσει έναν απλό κύβο!

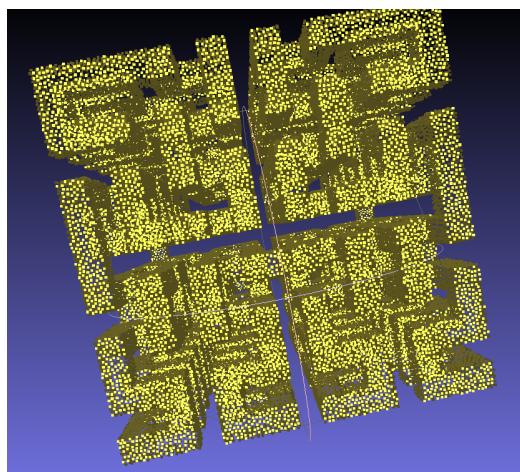


(α') Άρτιο αποτέλεσμα

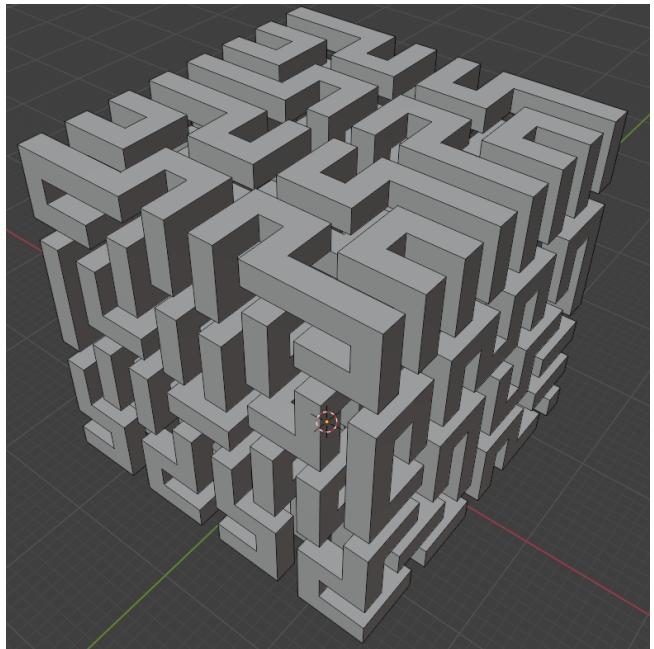


(β') Εικοσάεδρο. Καλό αποτέλεσμα ε-κτός από τις γωνίες.

$\Sigma\chi\eta\mu\alpha$ 15

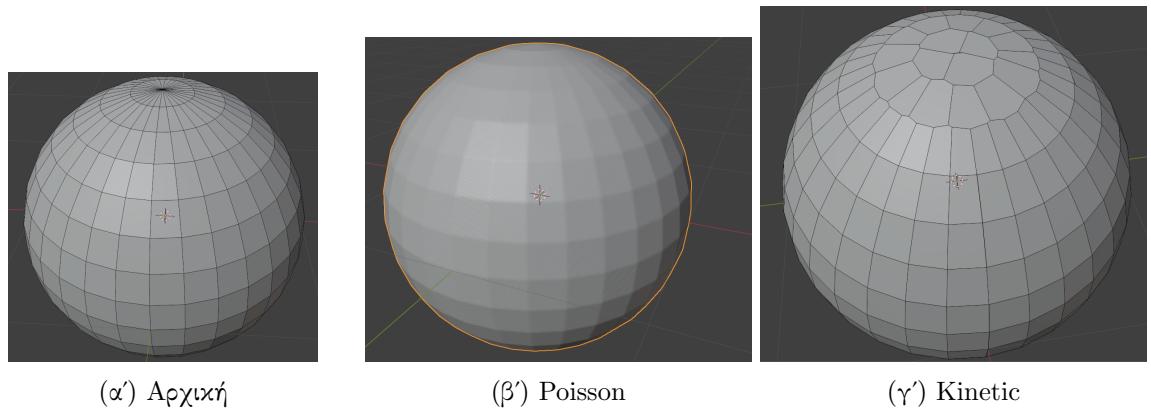


(α') Δεδομένα

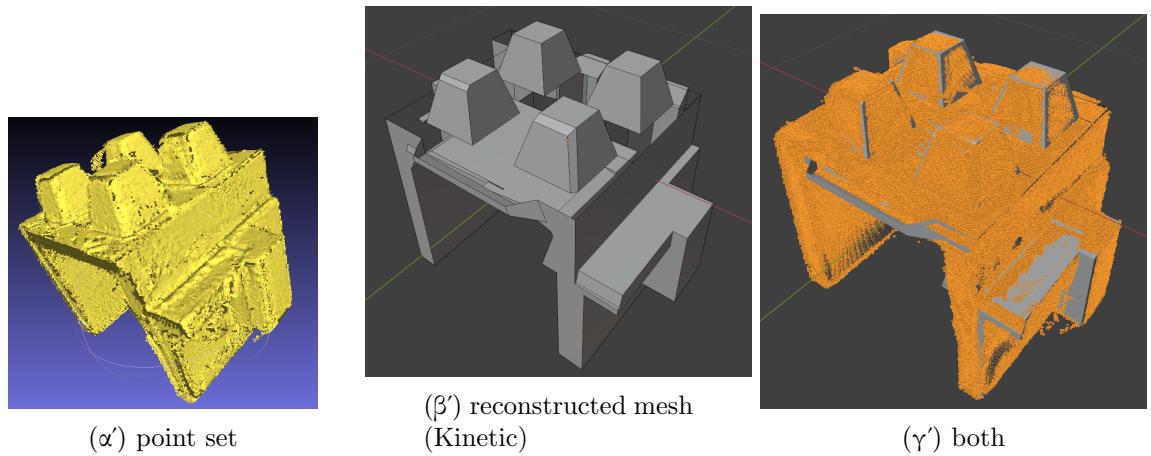


(β') Αποτέλεσμα

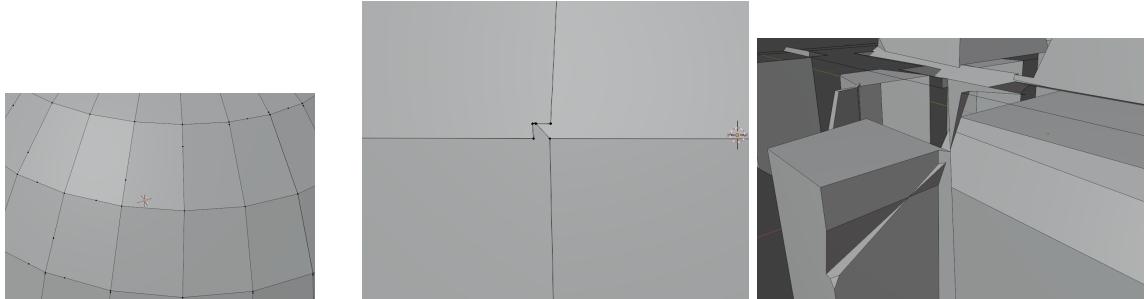
$\Sigma\chi\eta\mu\alpha$ 16: Hilbert cube



$\Sigma\chi\dot{\eta}\mu\alpha$ 17: Sphere consisting of planar shapes



$\Sigma\chi\dot{\eta}\mu\alpha$ 18: Foam Box

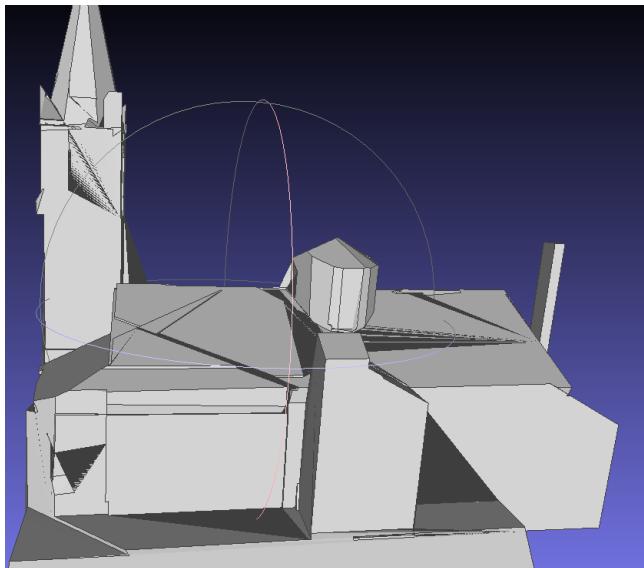


(α') Άσκοποι κόμβοι σε συνευθειακά τμήματα

(β') Περίεργες συνδέσεις σε ορθές γωνίες

(γ') Πλευρές που δεν οπτικοποιήθηκαν

Σχήμα 19: Προβλήματα



(α') Σε ένα κατά τα άλλα καλό αποτέλεσμα, υπάρχουν επικαλυπτόμενες πλευρές (σκιασμένες με γκρι)

Σχήμα 20: Degeneracies

6.3 Τεχνικά Προβλήματα

Στα περισσότερα αποτελέσματα της μεθόδους αυτής υπάρχουν degeneracies, π.χ. duplicate vertices, δηλαδή δύο κόμβοι που βρίσκονται στο ίδιο ακριβώς σημείο, κ.α.

Κάποια από αυτά λύνονται εύκολα με post processing, (π.χ. τα duplicates, ή οι αχρείαστοι κόμβοι). Άλλα δεν γνωρίζω (όπως τα degenerate faces που δε ξέρω ακριβώς γιατί προκύπτουν).

Τέλος, σε κάποια datasets ο αλγόριθμος κρασάρει και δεν παράγει αποτέλεσμα. Στο manual της CGAL για τα ίδια datasets παράγουν αποτελέσματα, όποτε μάλλον κάτι γίνεται λάθος από μεριάς μου.

7 Pre-processing

Δεν χρησιμοποίησα σε καμία από τις παραπάνω μεθόδους pre-processing των δεδομένων, το οποίο είναι αρκετά σημαντικό για τις περισσότερες. Τα αποτελέσματα επομένως δεν είναι πλήρη.