

LAB 3 – Modellazione con Blender & Geometry Processing con Meshlab

Fondamenti di Computer Graphics M
Erika Gardini – Ingegneria Informatica A.A. 2016/2017

Parte 1 – Blender

La prima parte dell'esercitazione consiste nella familiarizzazione con l'ambiente, nello svolgimento dei primi tutorial, nella creazione di superfici spline per estrusione, skinning, rotazione e swinging e nella modellazione di un oggetto o di una scena non banale.

La prima tecnica sperimentata è stata quella di ottenere le componenti di un tavolo a partire da forme elementari ed applicando ad esse le tecniche di estrusione, scalatura e traslazione.

L'immagine di riferimento in fase di modellazione è stata la seguente:



Per prima cosa è stata realizzata la zampa, partendo da un cilindro ed applicando la tecnica di estrusione e scalatura:

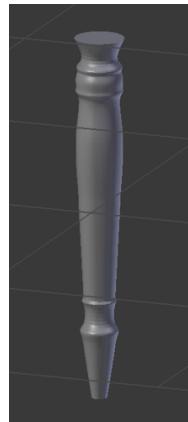


Figura 1: zampa del tavolo

L'effetto arrotondato è stato ottenuto aggiungendo il modificatore “subdivision surface” ed aumentando il valore di view a 2 ed il valore di render a 3.

A questo punto si è passati alla realizzazione degli altri componenti mediante l'utilizzo delle stesse tecniche, partendo però da cubi:

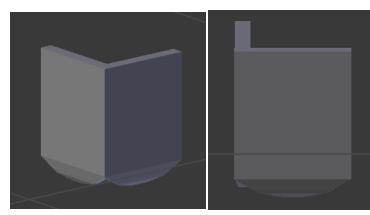


Figura 2: dettaglio angolo

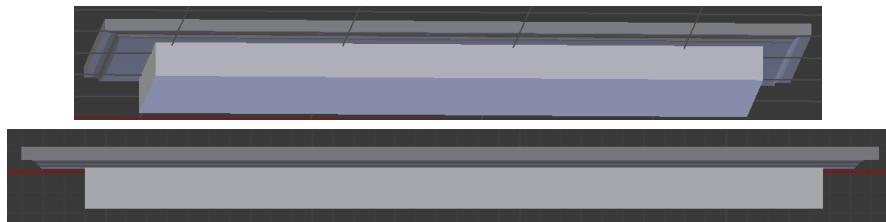


Figura 3: piano del tavolo

Unendo tutti i componenti insieme e duplicando la zampa con il comando “shift+d” è stato ottenuto il seguente risultato:

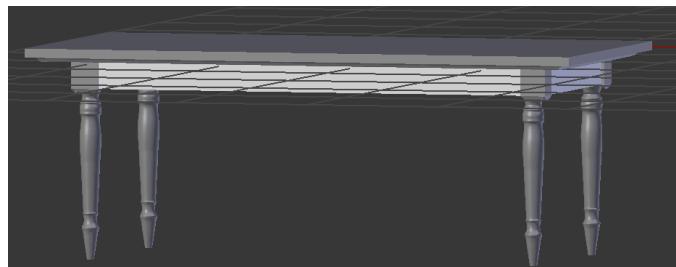


Figura 4: tavolo completo

A questo punto si è passati alla realizzazione delle sedie, partendo sempre dalla forma base del cubo ed utilizzando le tecniche indicate precedentemente.

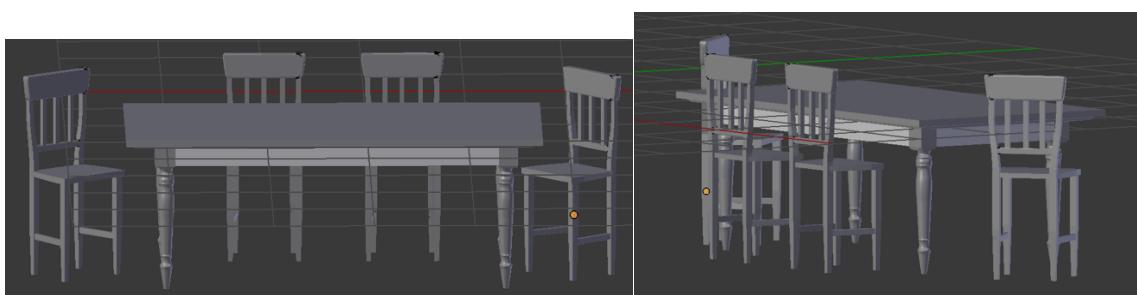


Figura 5: tavolo e sedie

Infine è stata realizzata una libreria a partire da un cubo; le tecniche utilizzate per lo svuotamento e lo spessore sono state prese dal tutorial al seguente link: https://www.youtube.com/watch?v=6uJ6kd9_CfQ. Il risultato ottenuto è il seguente:

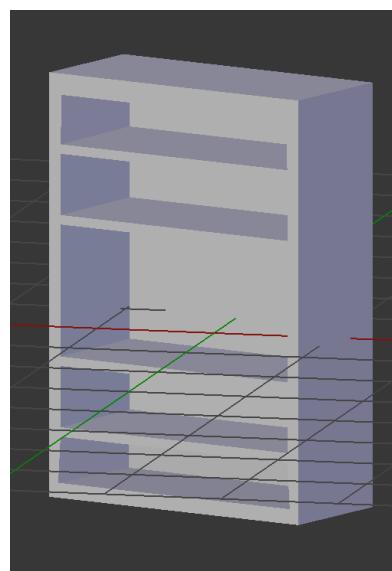


Figura 6: libreria

Per arricchire la scena sono stati aggiunti due diversi tipi di vaso, il primo ottenuto a partire da una curva NURBS fatta ruotare lungo l'asse x, il secondo ottenuto a partire da curve NURBS superficie (tecnica skinning). I risultati ottenuti sono stati i seguenti:

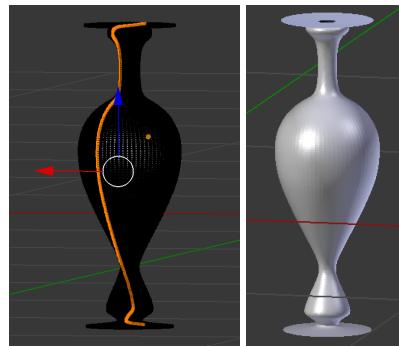


Figura 7: vaso da curva NURBS ruotata lungo x



Figura 8: centrotavola da curve NURBS superficie

Il modello finale è il seguente:

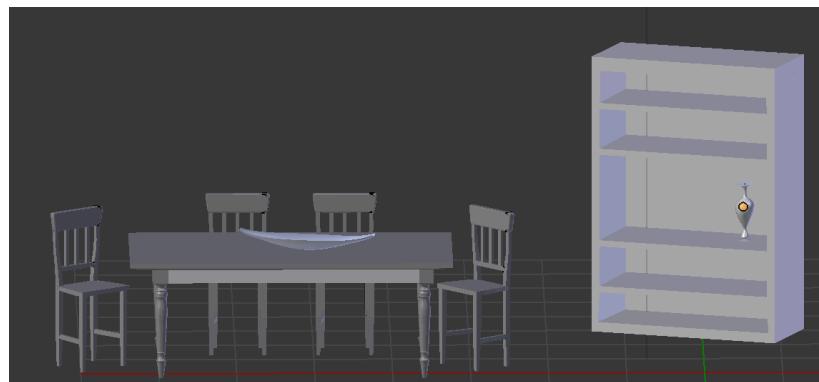


Figura 9: modello stanza

Una volta realizzato il modello sono stati assegnati i colori agli oggetti in scena. È stata utilizzata una texture solo per la libreria. Inoltre, sono stati aggiunti due piani in modo da realizzare il pavimento e la parete. Infine, sono state aggiunte delle luci in modo da illuminare la scena.

Il risultato finale è il seguente:

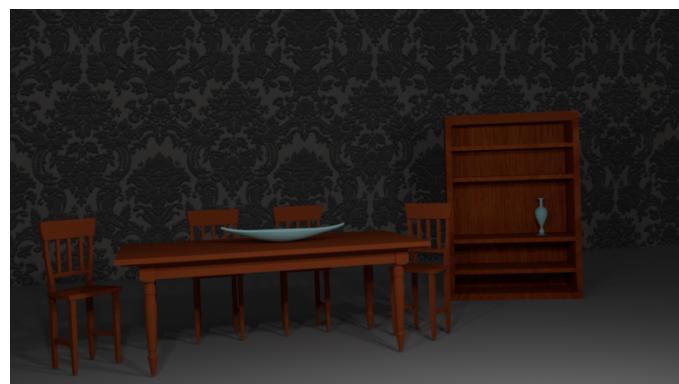


Figura 10: rendering della scena

Terminato il primo modello si è passati alla modellazione del personaggio Olaf del cartone animato Disney Frozen. Per realizzarlo sono state utilizzate le tecniche di estrusione, traslazione e scalatura su sfere e coni. Inoltre, nella realizzazione del modello è stato seguito il tutorial al seguente link: https://www.youtube.com/watch?v=xxr_MB_KDHU.

Per prima cosa è stato modellato il corpo e la testa, unendo insieme sfere scalate in maniera differente:

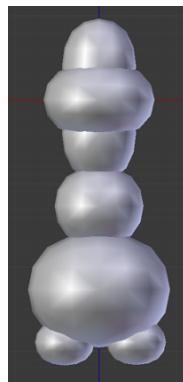


Figura 11: corpo e testa Olaf

Sono poi stati aggiunti gli occhi (ottenuti da sfere), il naso (ottenuto da un cono), i denti (ottenuti da un cubo), le sopracciglia (ottenute da un cubo) e le ombre degli occhi (ottenute da tori):

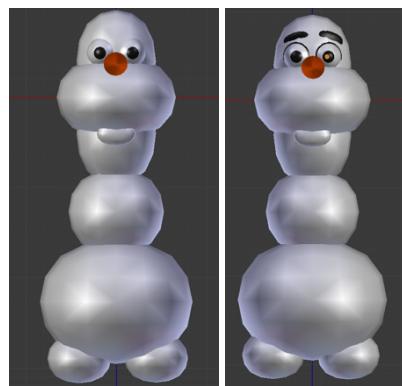


Figura 12: dettagli volto Olaf

Infine, sono state aggiunte le braccia (partendo da cilindri), i capelli (partendo da cilindri) ed i bottoni (partendo da cubi):



Figura 13: dettagli corpo Olaf

Per l'aggiunta dello sfondo e della luce è stato seguito il tutorial al seguente link: https://www.youtube.com/watch?v=y_mYwLJVs50.

Il risultato finale è il seguente:



Figura 14: rendering Olaf

Parte 2: Geometry Processing con Meshlab

Nella seconda parte dell'esercitazione veniva richiesto di sperimentare una serie di funzionalità di post-process di dati acquisiti da sessioni di digitalizzazione.

1) Ricostruzione di oggetti mesh a partire da nuvole di punti (Poisson, MLS, Marching cubes)

Per realizzare questa prima richiesta è stata caricata in Blender una mesh. Passando in “edit mode”, selezionando la mesh, premendo tasto “x” e scegliendo l'opzione “edge&faces” è possibile ottenere la nuvola di punti:

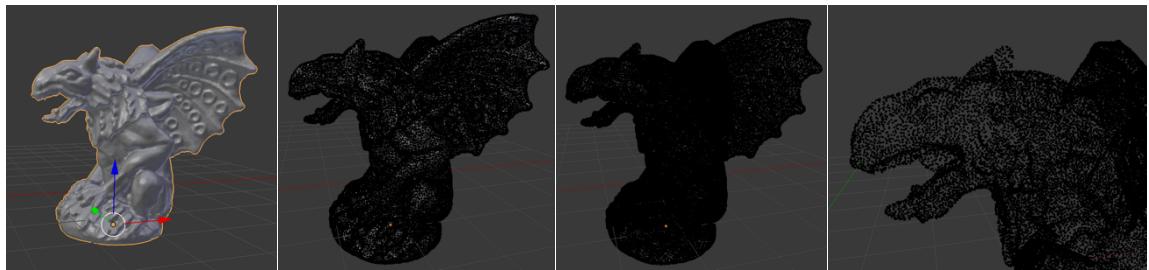


Figura 15: da mesh a nuvola di punti

Una volta esportata la nuvola di punti, è necessario importarla in meshlab:

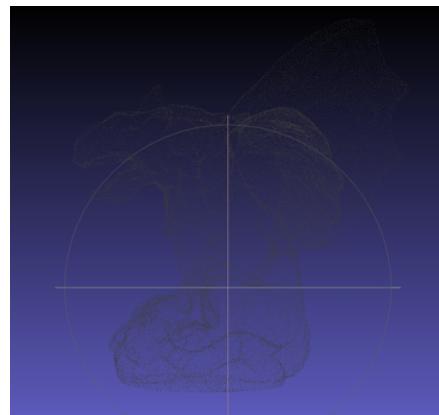


Figura 16: nuvola di punti su meshlab

A questo punto è stato applicato il filtro “surface reconstruction: ball pivoting” in modo tale da ottenere i vertici a partire dalla nuvola di punti.

In particolare, l'algoritmo, dato un insieme di punti e le loro normali, ricostruisce iterativamente una superficie a partire da un “triangolo seme” e basandosi soltanto sui punti esistenti, senza creare di nuovi. Pertanto, le prestazioni sono massimizzate con nuvole di punti uniformi.

Il risultato ottenuto è il seguente:

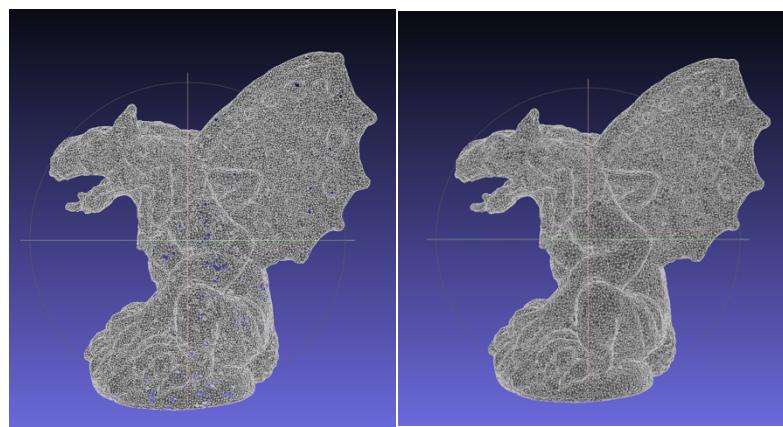


Figura 17: costruzione della mesh a partire dalla nuvola di punti (parametro 0, poi 30%)

Impostando nel menù di destra shading=face e color=mesh è possibile ottenere la seguente visuale:

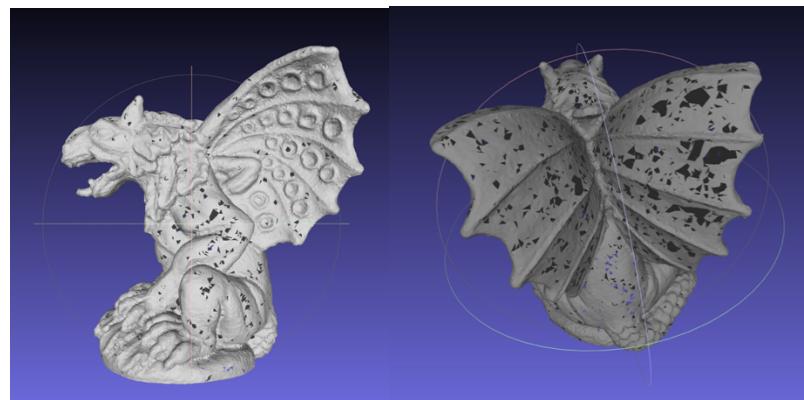


Figura 18: mesh parzialmente corrotta

Problema: la mesh risulta essere parzialmente corrotta.

2) Utilizzo dei tool **Fill Hole /Mesh Repair** per la chiusura di mesh parzialmente corrotte

Per la realizzazione della seguente funzionalità è stata utilizzata la mesh parzialmente corrotta ottenuta dalla scansione di una palma:

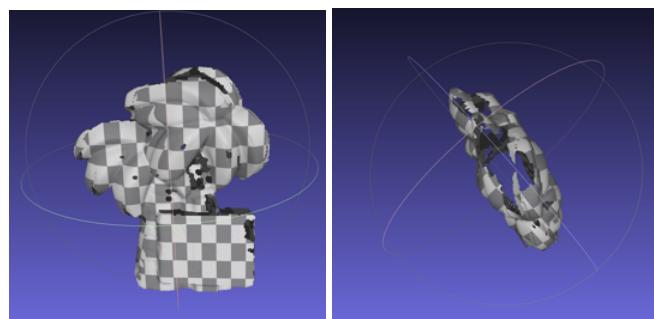


Figura 19: palma corrotta

Settando dal menù di destra “boundary edges = on” ed applicando il filtro “close holes” con diversi valori del parametro “max size to be closed” è possibile ottenere il seguente risultato:

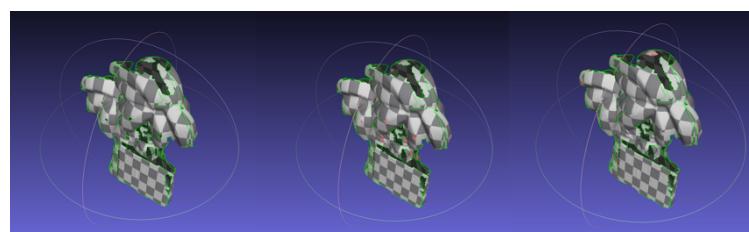
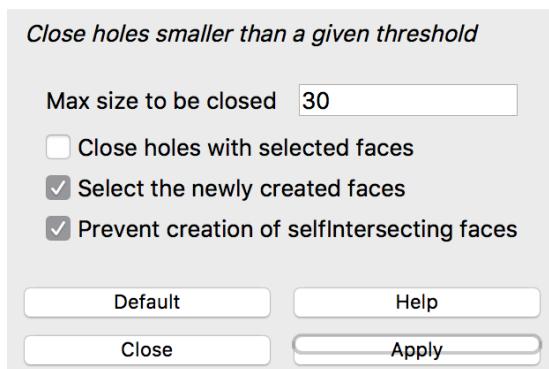


Figura 20: filtro "close holes" con parametri 30, 50, 100

Problema: la mesh corrotta non viene completamente chiusa; inoltre, aumentando il valore del parametro sopra indicato la correzione effettuata risulta essere approssimata, provocando quindi delle “deformazioni”.

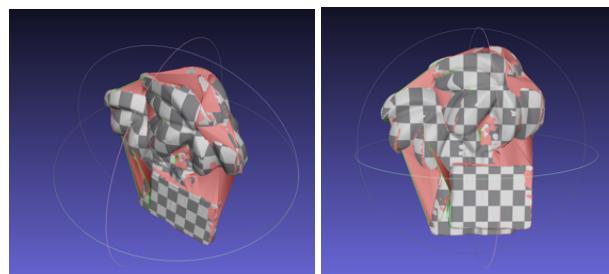
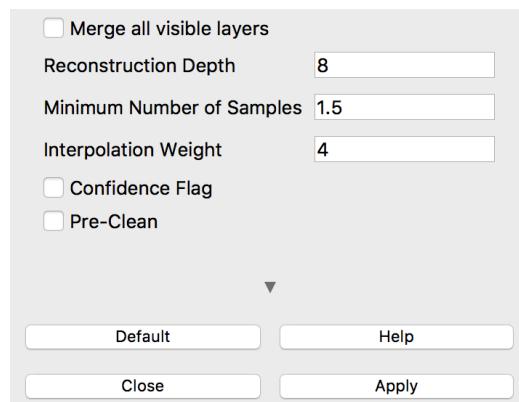


Figura 21: filtro "close holes" con parametro 1000

Per correggere la mesh è stato quindi utilizzato un altro filtro: “screened poisson surface reconstruction” con parametro “minimum number of samples” incrementato da 1,5 a 5 e selezionando l’opzione “pre-clean”.



Il risultato ottenuto è il seguente:



Figura 22: filtro "screened poisson surface reconstruction" con parametro da 1,5 a 5

Lo stesso procedimento è stato ripetuto sulla mesh al punto 1, ottenendo il seguente risultato:

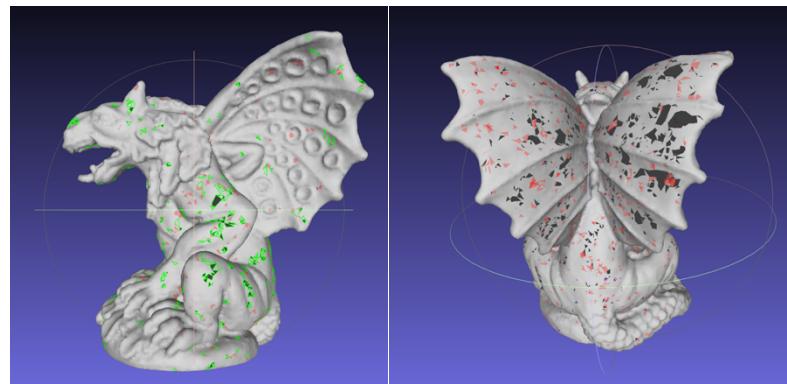


Figura 23: filtro "close holes" con parametro 50

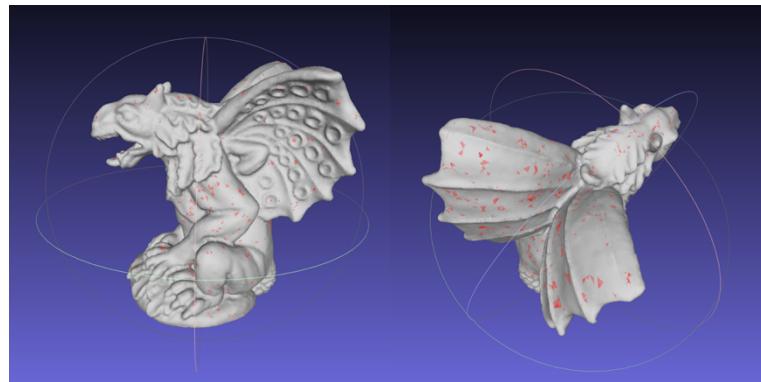
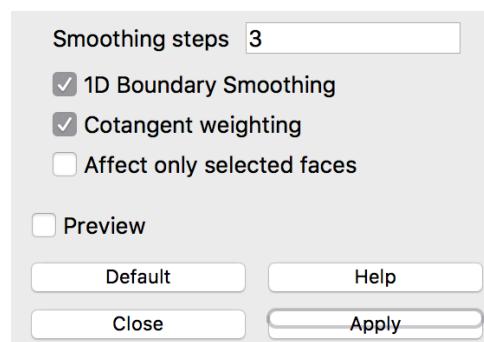


Figura 24: filtro "screened poisson surface reconstruction"

3) FAIRING: applicare un filtro di denoising (fairing) ad una mesh perturbata

L'operazione di fairing ha come obiettivo quello di minimizzare la variazione di curvatura, la quale genera la granularità della mesh, in modo tale da rendere la superficie più liscia possibile. Questa operazione non modifica la topologia della mesh.

Per realizzare la seguente funzionalità in Meshlab, è stato applicato il filtro "laplational smooth" ad una mesh perturbata. Il filtro calcola per ogni vertice la posizione media con gli altri vertici vicini.



Variando il parametro "smoothing steps" sono stati ottenuti i seguenti risultati:

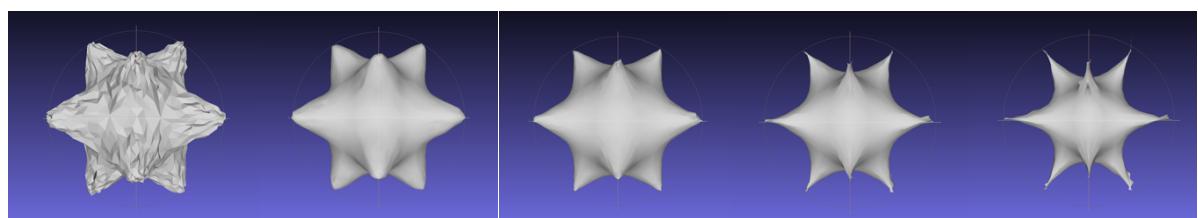
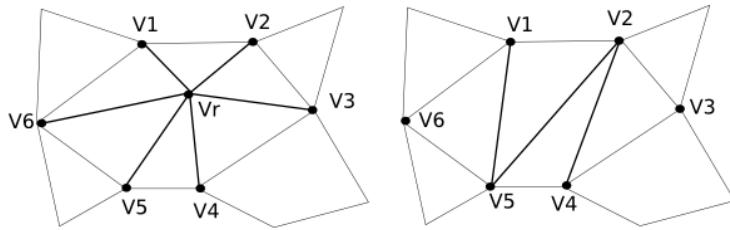


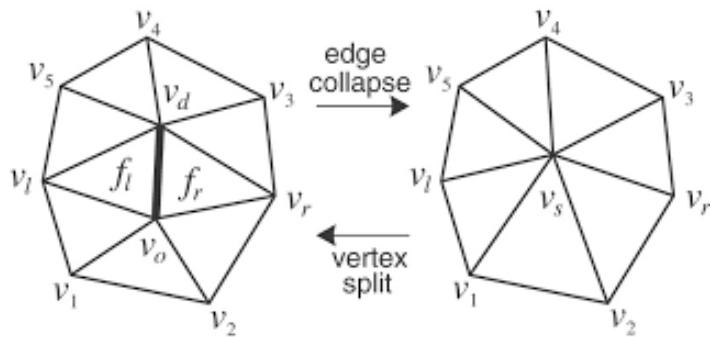
Figura 25: laplational smooth con parametro 3, 10, 50, 100

4) DECIMATION: semplificare a più livelli una mesh con un numero elevato di elementi

La decimazione viene effettuata per diminuire il numero di vertici di una mesh.



Gli algoritmi che effettuano una edge decimation collassano due vertici in uno unico. In questo modo è possibile ridurre il numero di vertici senza aumentare la valenza (numero di edge che incidono su un certo vertice) e quindi mantenendo la stabilità della mesh stessa.



Fra gli algoritmi che effettuano edge decimation si distinguono quelli che utilizzano la quadric error matrix. Questi:

- associano a ciascun vertice un costo (formato dalla somma dei costi dati dalla distanza del vertice dai piani di cui il vertice stesso fa parte);
- per ogni edge calcolano il costo del suo collasso (dato dal costo dei due vertici che si vanno a collaudare);
- scelgono l'edge di costo minimo;
- effettuano la decimazione andando a posizionare il nuovo vertice nel punto di costo minimo;
- ricalcolano il costo del nuovo vertice;
- ripetono il procedimento fino a che non si raggiunge una certa soglia.

Per realizzare la funzionalità di decimation in Matlab si è partiti da una mesh composta da un elevato numero di vertici (863.210):



Figura 26: mesh con elevato numero di vertici

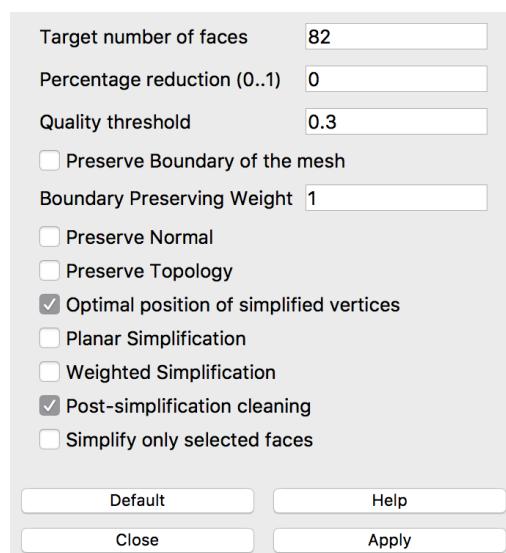
Per prima cosa è stato applicato il filtro “simplification: MC edge collapse” ottenendo il seguente risultato:



Figura 27: prima semplificazione con filtro "simplification: MC edge collapse"

Come si può notare il risultato non è dei migliori.

È quindi stato applicato il filtro “simplification: quadric edge collapse decimation” riducendo il parametro “target number of faces”.



I risultati ottenuti sono stati i seguenti:

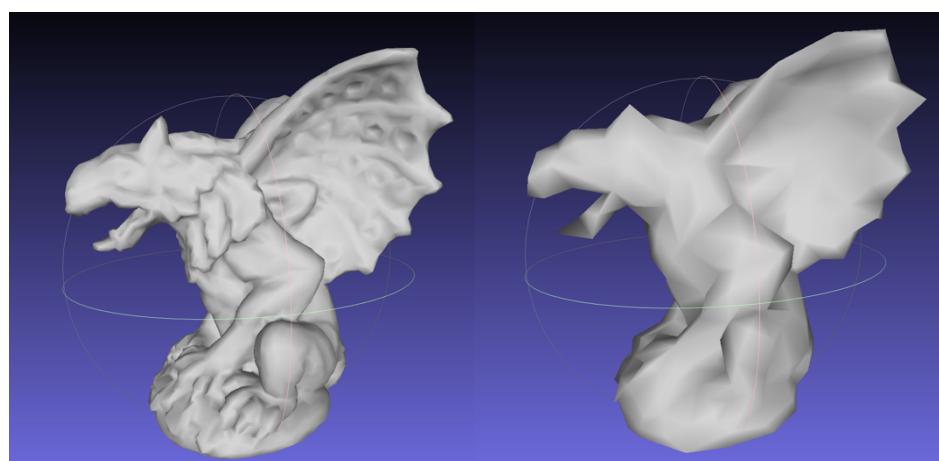
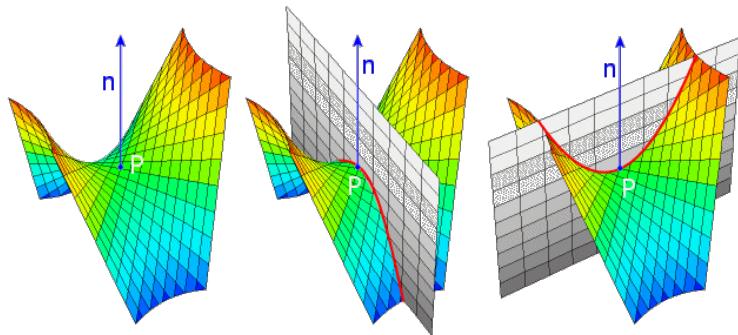


Figura 28: seconda semplificazione con filtro "simplification: quadric edge collapse decimation" con parametro 10000 e poi 1000

5) Utilizzare gli strumenti di misura della qualità della superficie (curvatura)

Preso un punto, ogni direzione nel piano tangente definisce una curva data dall'intersezione tra il piano ortogonale al piano tangente, contenente n , e la superficie.



Le curve passanti in direzioni differenti hanno diversi valori di curvatura.

Partendo dalla seguente mesh:



Figura 29: Mesh di partenza

la curvatura è stata calcolata in Matlab applicando il filtro “discrete curvature” e sono state selezionate le varie alternative del parametro “type”:

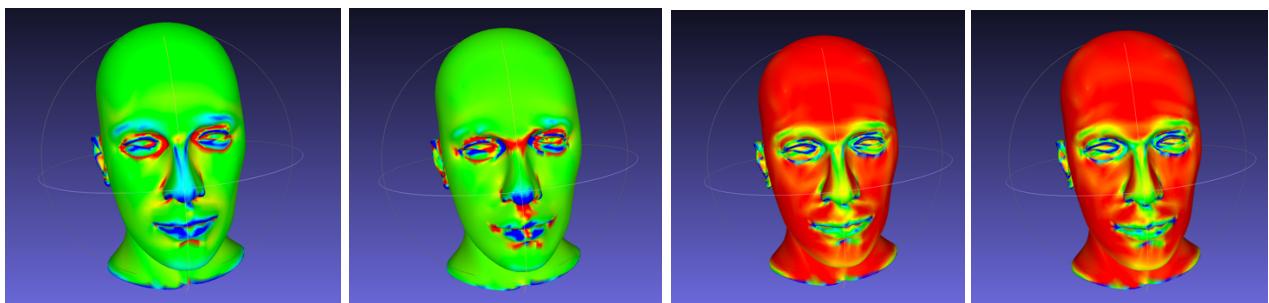
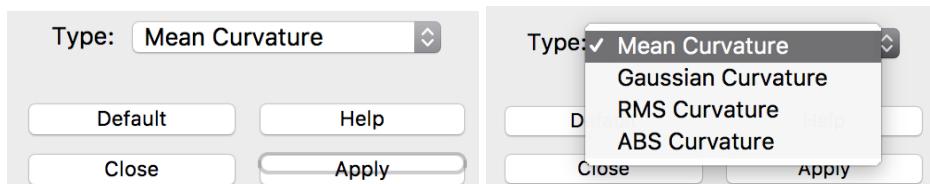


Figura 30: Mean Curvature - Gaussian Curvature - RMS Curvature - ABS Curvature