Algoritmi e strutture dati per la gestione multirisoluzione di superfici

di Claudio Rocchini

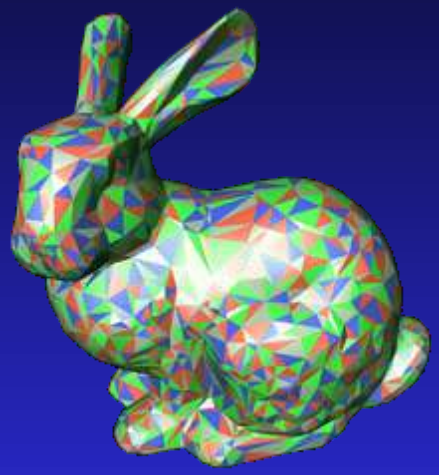
Mesh di grandi dimensioni

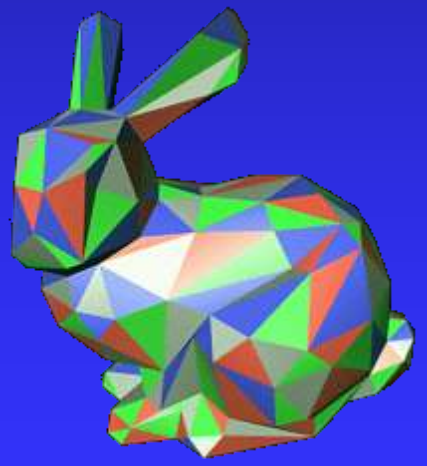
- Modellazione di terreni (simulatori di volo)
- Modellazione di solidi (CAD)
- Modellazione matematica di superfici
- Acquisizione automatica da scanner 3D
- Estrazione isosuperfici da dati volumetrici
- Descrizioni di ambienti per la realtà virtuale

(100k - 1 milione di triangoli)

Semplificazione di Mesh

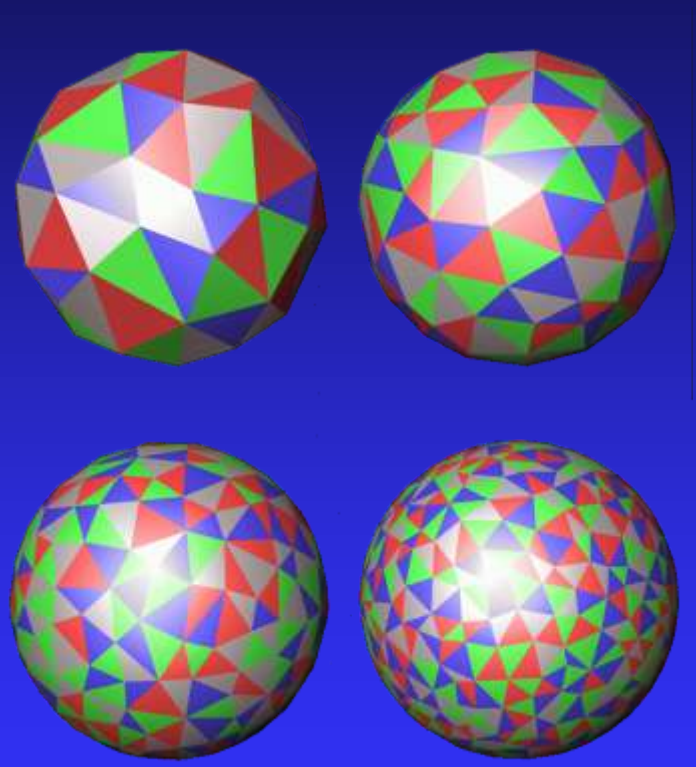
- Obiettivi:
  - Velocizzazione
  - Occupazione di memoria
- Algoritmi:
  - Sui vertici
    - Eliminazione (decimazione)
    - Inserimento (raffinamento)
    - Ridisposizione
  - Sugli spigoli
    - Collassamento di vertici
- Un nuovo obiettivo
  - Resolution modeling





Modelli multirisoluzione

- Rispetto al contenuto informativo:
  - Multilivello di dettaglio, collezione di livelli
    - *n* “piccolo”
  - Multirisoluzione
    - *n* livelli di dettaglio , con *n* proporzionale alla cardinalità dei dati
- Rispetto al processo costruttivo:
  - Schemi dipendenti dal processo di semplificazione
  - Schemi indipendenti dal processo

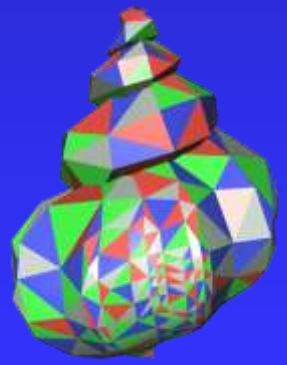


Obiettivi della tesi

- Algoritmi e strutture dati che permettano:
  - *Indipendenza* dagli algoritmi di raffinamento
  - Estrazione efficiente di una rappresentazione a *risoluzione fissa*
  - Gestione unificata del *raffinamento* selettivo e della *semplificazione* selettiva
  - *Composizione* di operazione di editing
  - Garanzia della *continuità* C^0 delle superfici a livello di errore variabile
  - Risposta in *tempo reale*

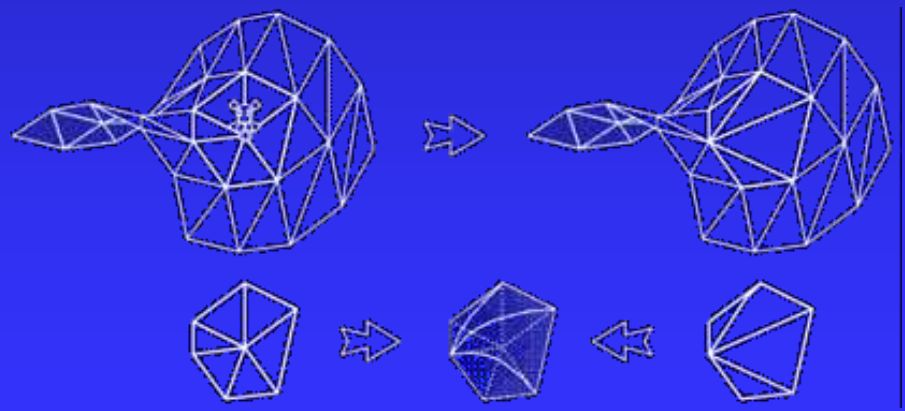
Modeling Phases

- Modellazione di superfici (design/acquisition)
- Costruzione modello multirisoluzione
- Resolution Modeling



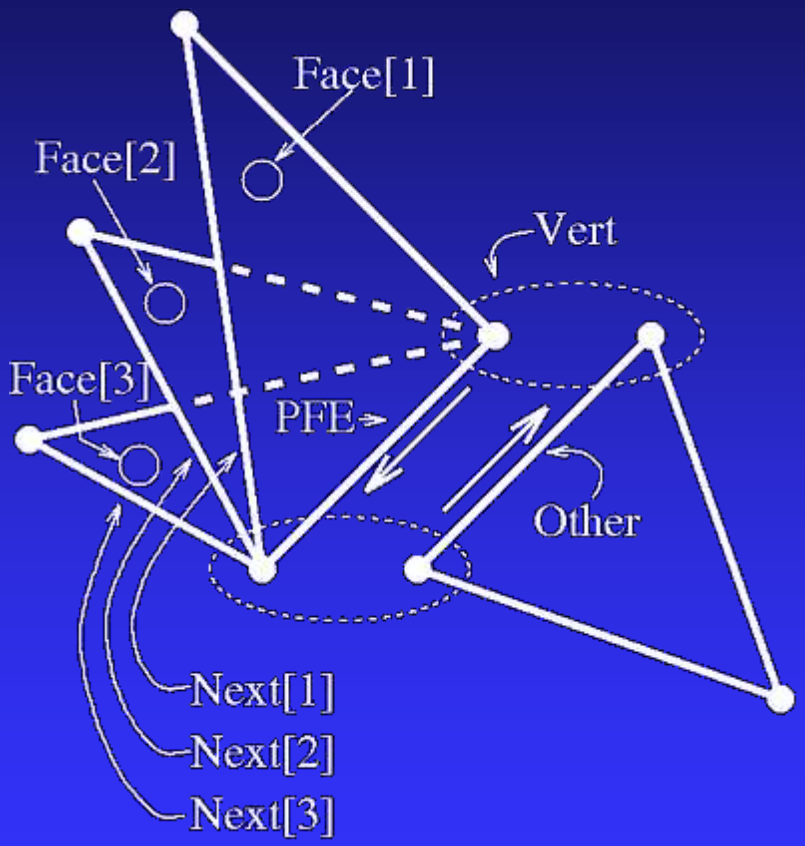
Ipertriangolazione

- Lista di triangoli con intervalli di errore
  - Errore di nascita (b_error)
    - 0, se il triangolo fa parte della superficie T_0
    - ε, se il triangolo è stato generato con un errore ε
  - Errore di morte (d_error)
    - ∞, se il triangolo fa parte della triangolazione meno definita
    - ε, se il triangolo è stato sostituito da un altro con errore ε
- Composizione:
  - Ricompattamento
  - Fusione

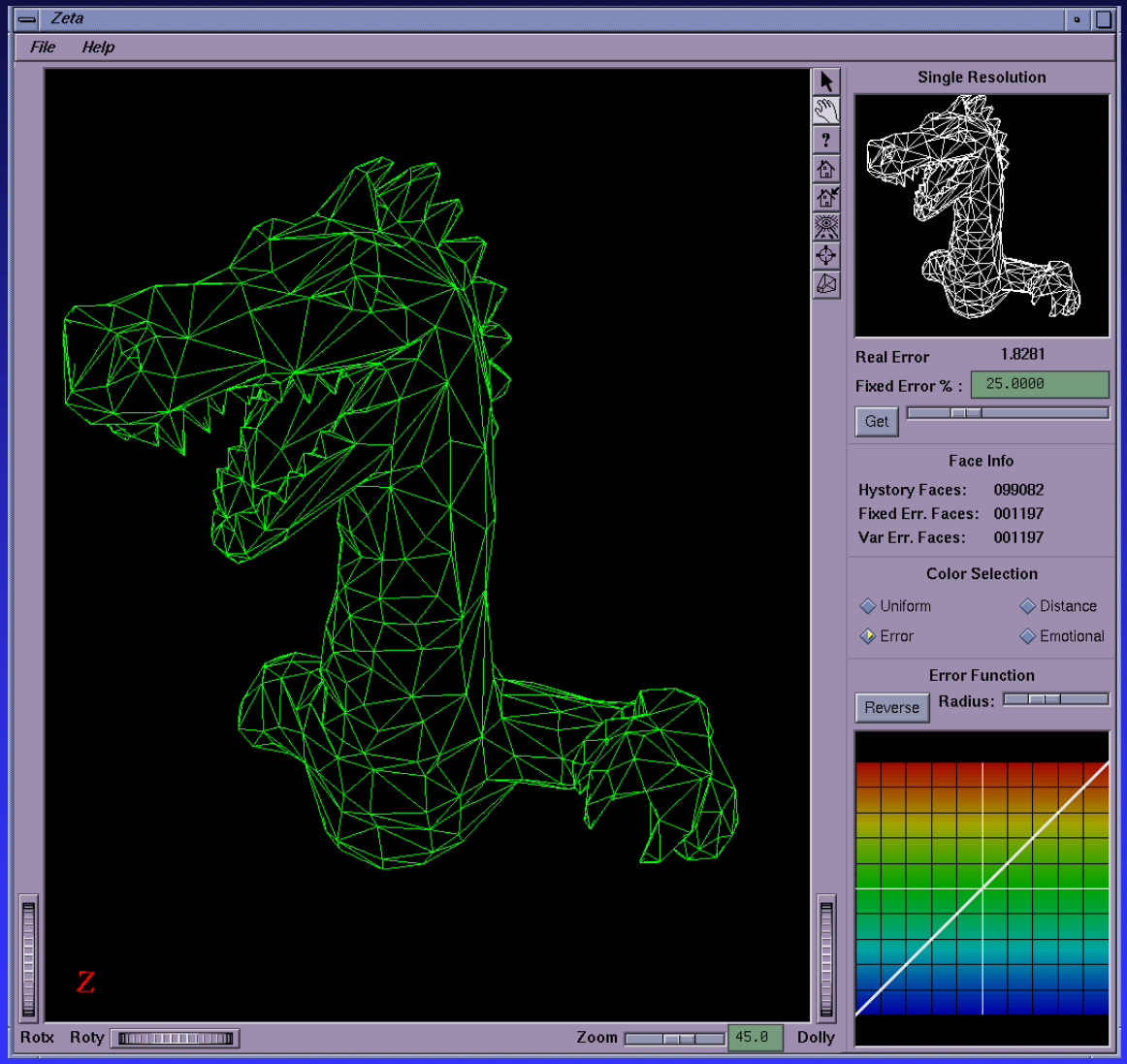


Struttura dati

```
PFEdge = Record
    Vert : ↑Point3D;
    Other : ↑PFEdge;
    Size : Integer;
    Pack : Array[1..Size] Of
        Record
            Face : ↑Face;
            Next : ↑PFEdge
        End
End;
```



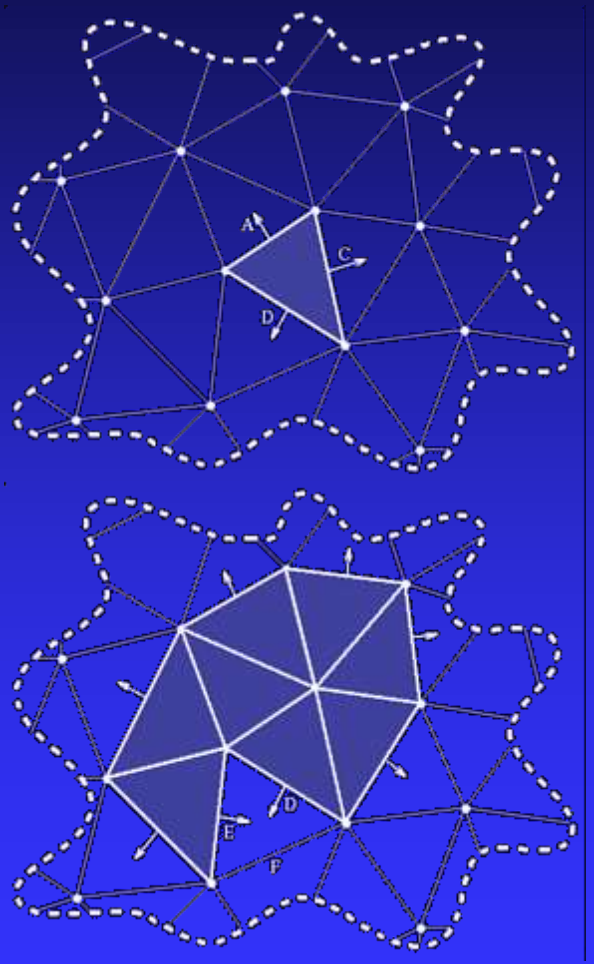
Prototipo: risoluzione fissa



Risoluzione fissa

- Algoritmo:

$T := \varnothing$; $Stack := e_{first}$;
<u>While</u> $Stack \neq \varnothing$ <u>Do</u>
$e := Pop(Stack)$;
$f := ValidFace(e, \varepsilon)$;
$T := T \cup \{ f \}$;
$e := e\uparrow.next$;
<u>If</u> $e \in Stack$ <u>Then</u>
$Stack := Stack\text{-}e$
<u>Else</u>
$Stack := Stack \cup e\uparrow.other$
<u>EndWhile</u>;



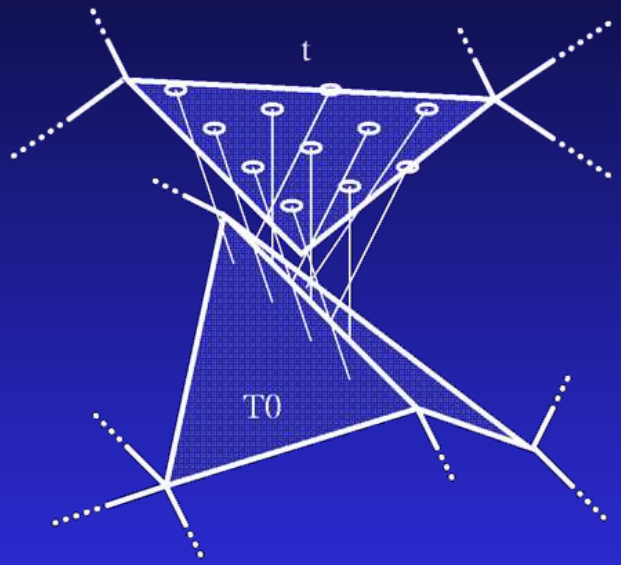
Calcolo dell'errore

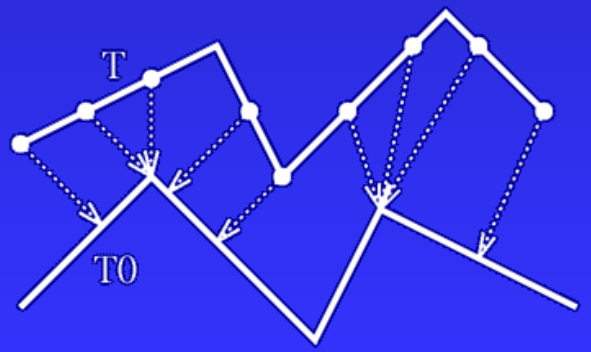
- Obiettivi
  - Integrazione
  - Uniformità di misurazione
- Definizione di distanza

$$e(p, T_0) = \min_{p' \in T_0} \| p - p' \|$$

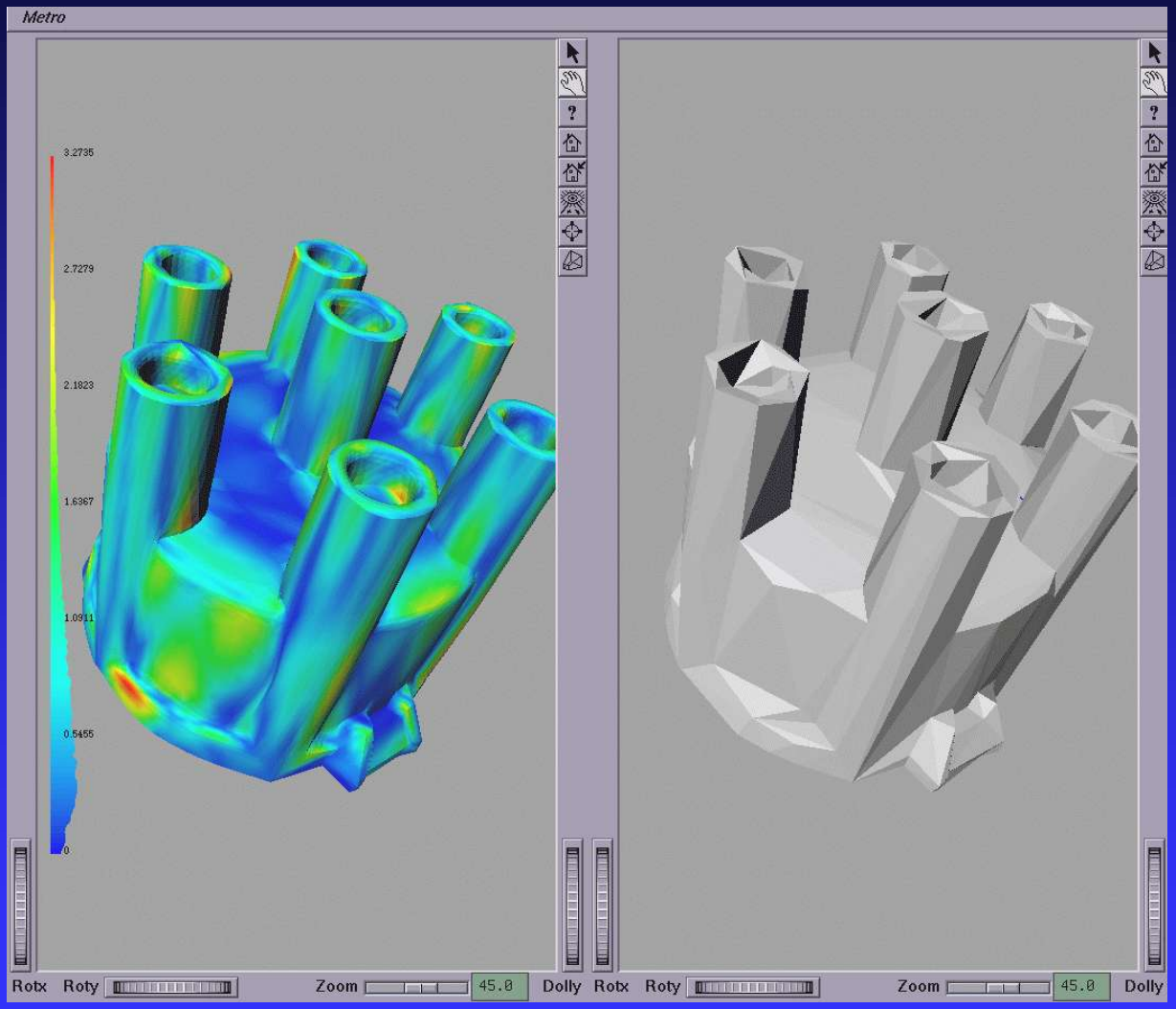
$$e(t, T_0) = \left(\int_t e(p, T_0) dp \right) \frac{1}{A(t)}$$

- Calcolo effettivo
  - Campionamento
  - Distanza punto-superficie

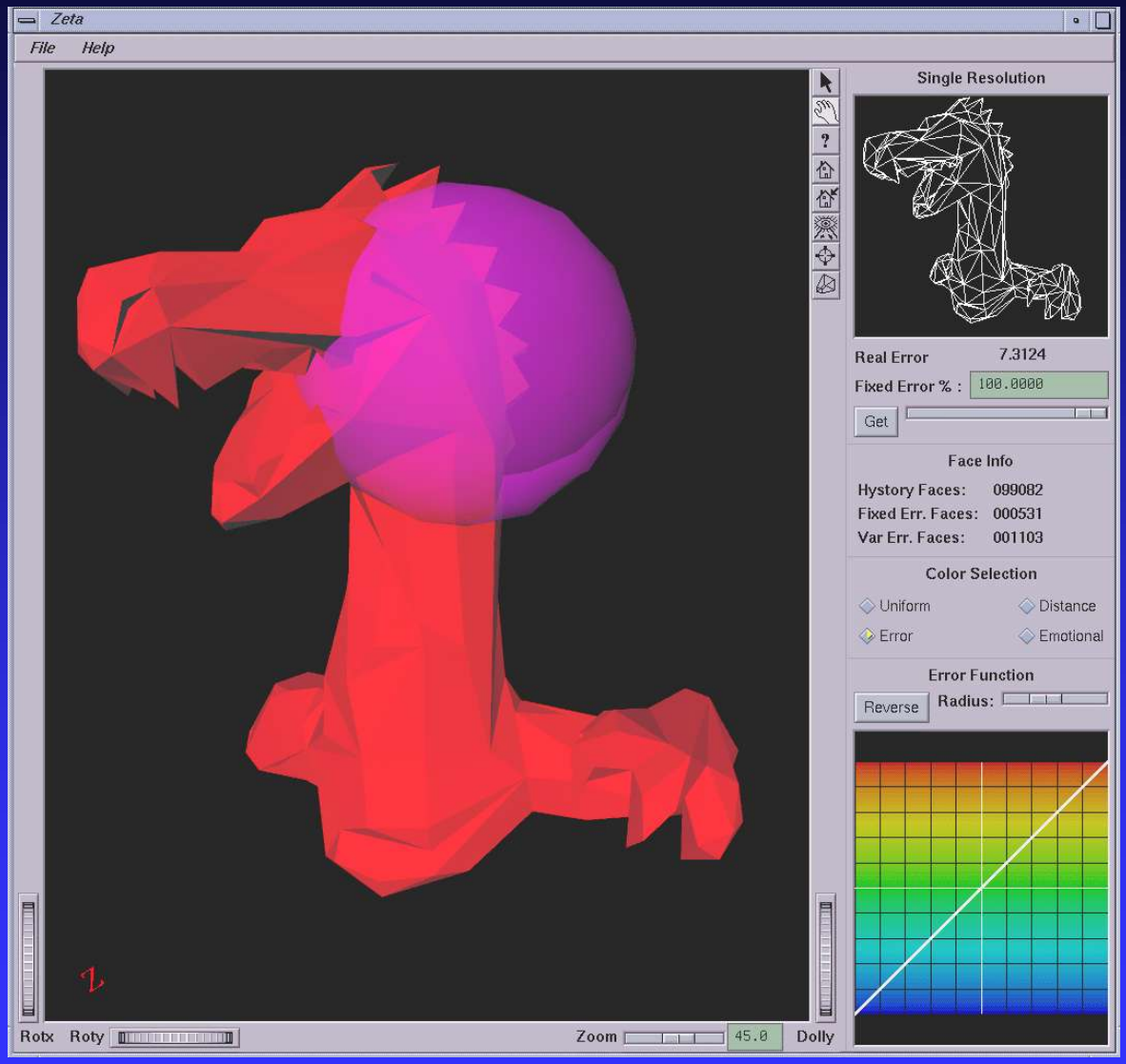




Metro: misurazione dell'errore

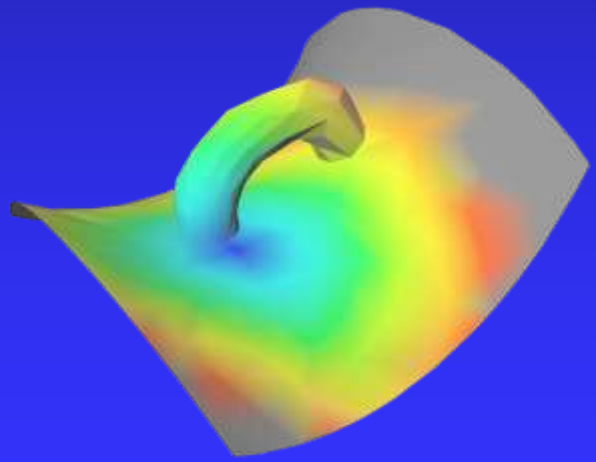


Prototipo: raggio di azione

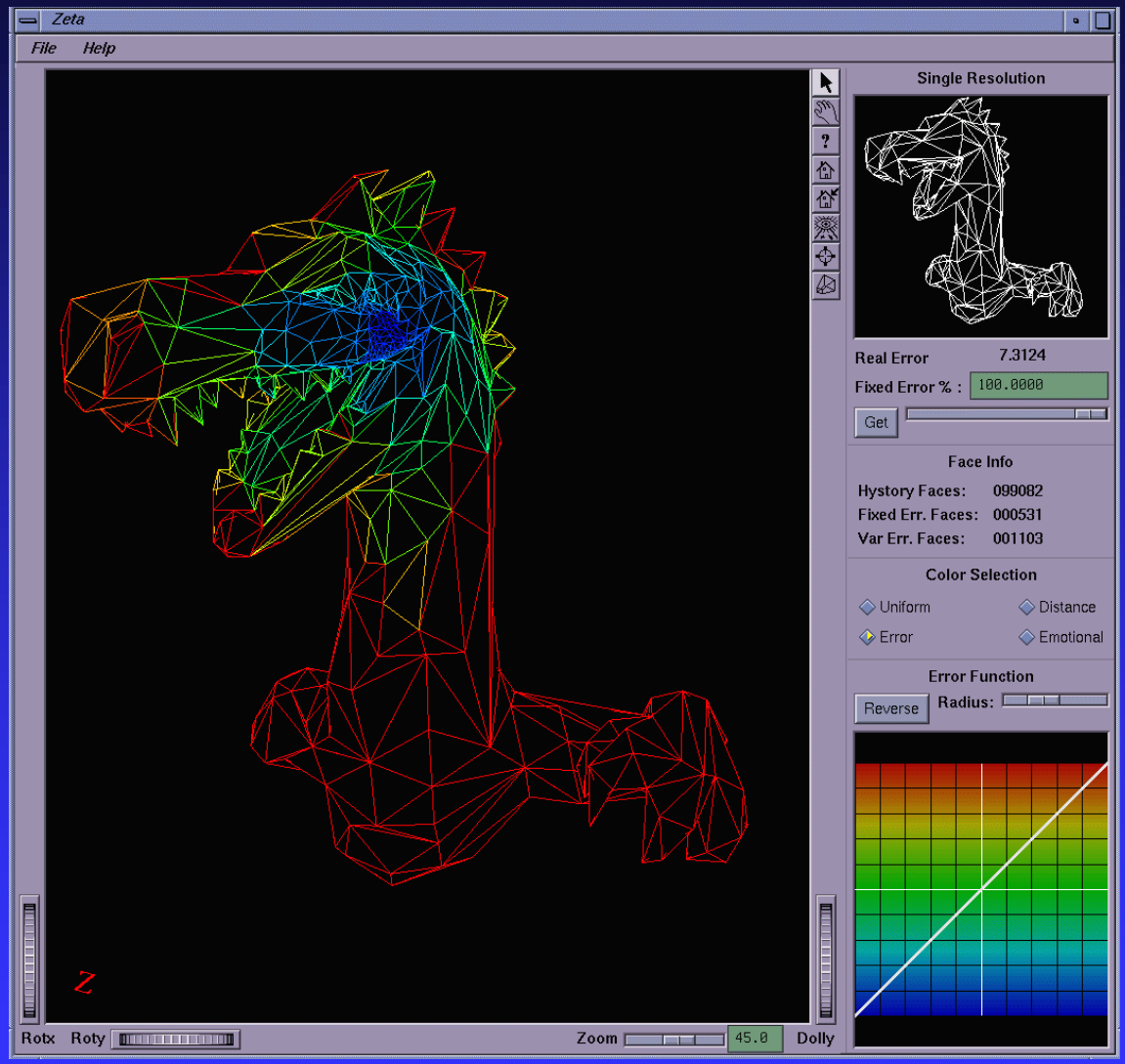


Metrica di una ipertriangolazione

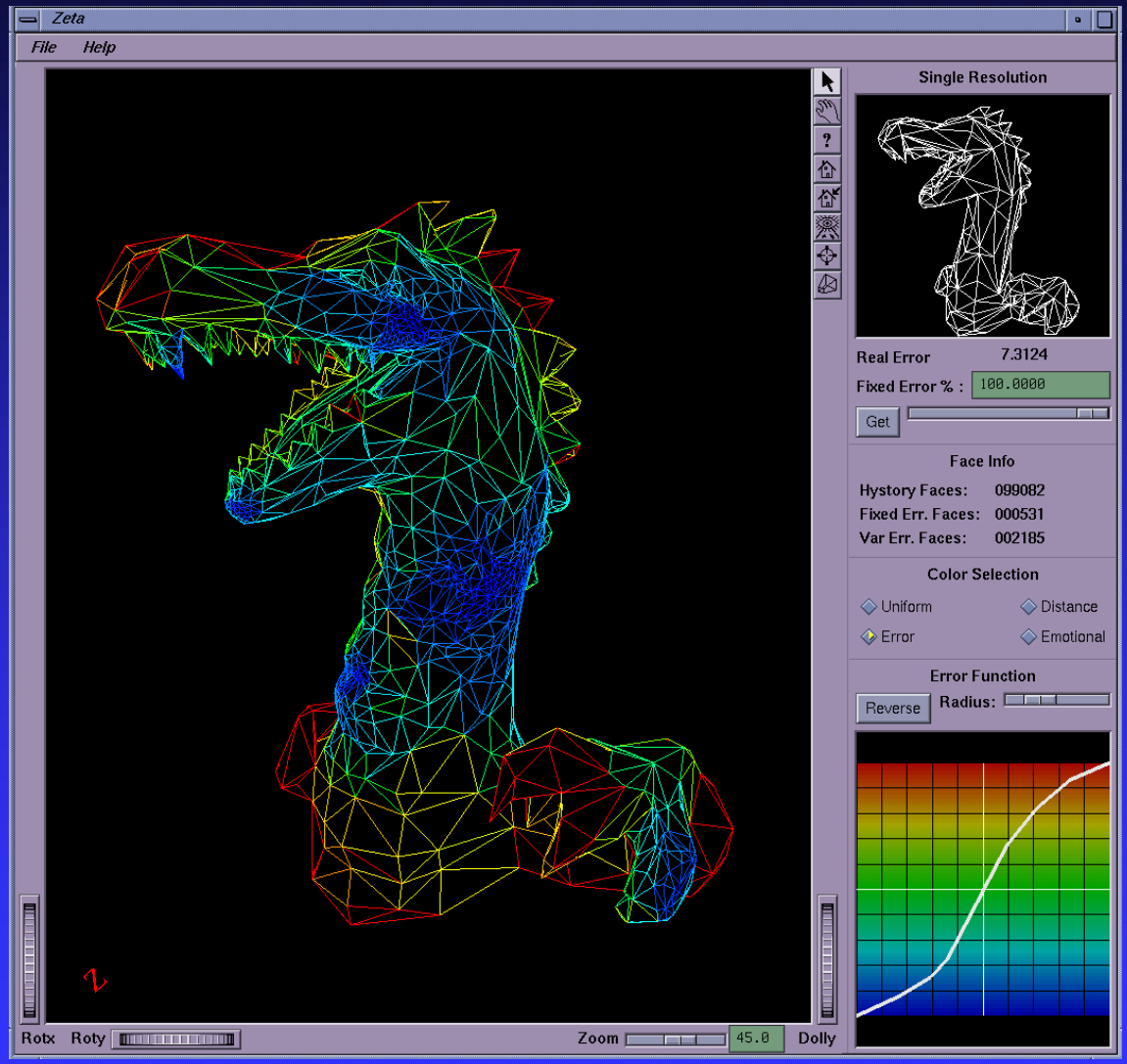
- Scopi
  - Definire il raggio di azione
  - Definire il livello di risoluzione in funzione della distanza dal punto di editing
- Definizione di una metrica
  - Distanza geodetica dei vertici
  - Approssimazione lungo gli spigoli
- Algoritmo di calcolo
  - Albero dei cammini minimi sul grafo degli spigoli



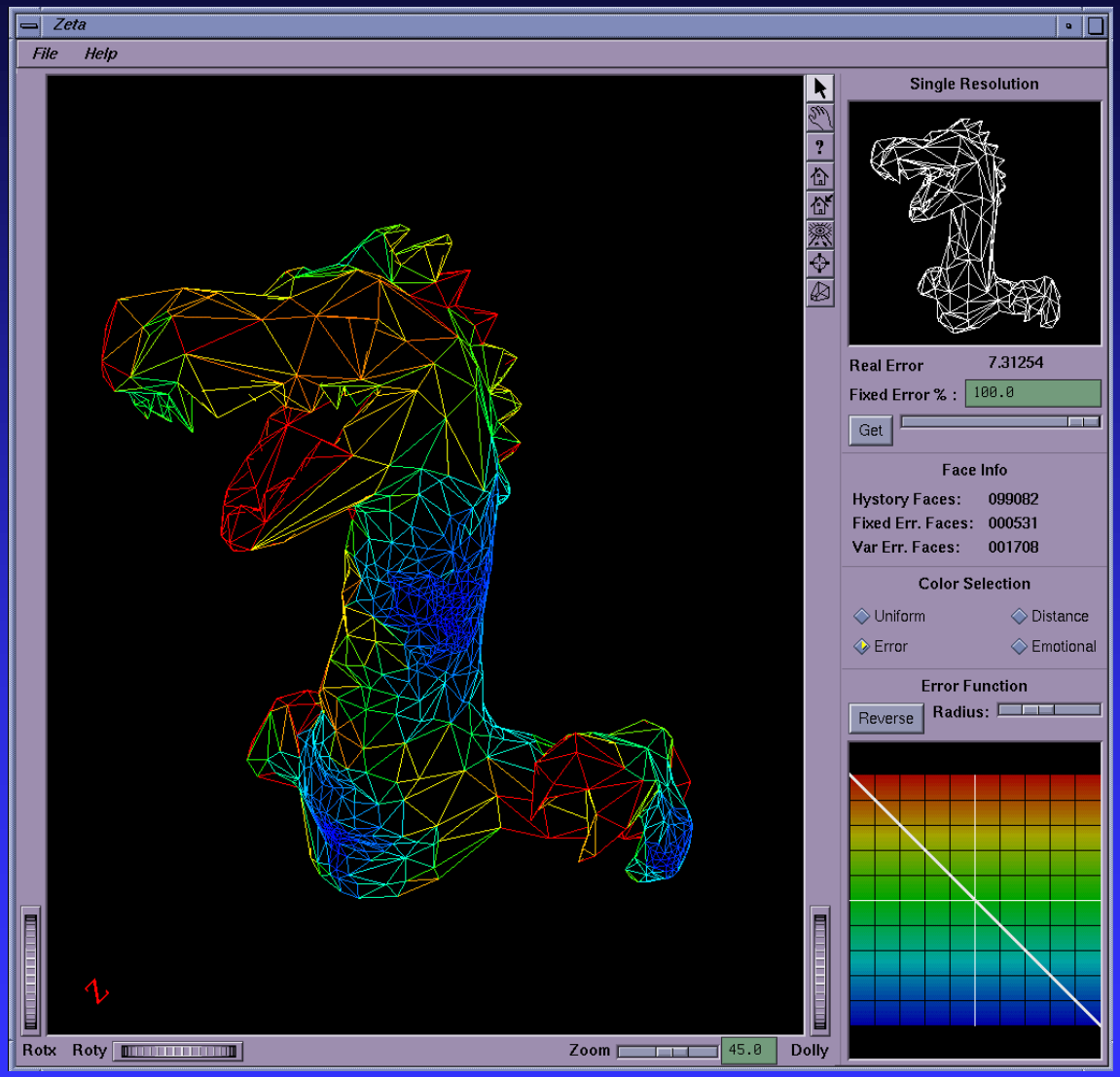
Prototipo: raffinamento



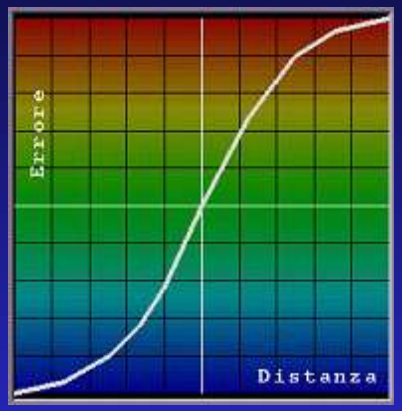
Prototipo: raffinamenti successivi



Prototipo: semplificazione



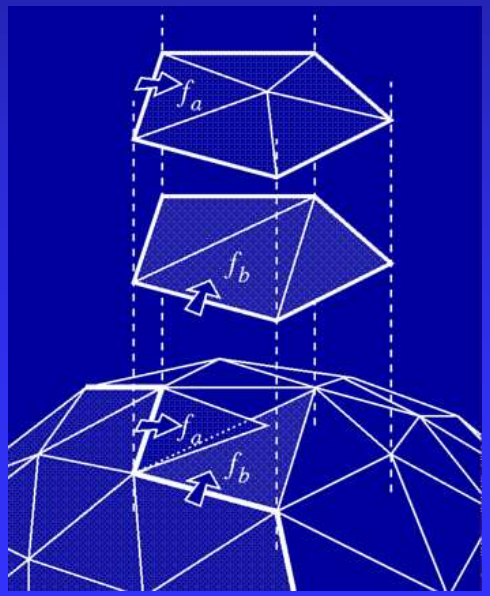
Risoluzione Variabile



- Errore variabile durante la visita
- Condizione di consistenza:

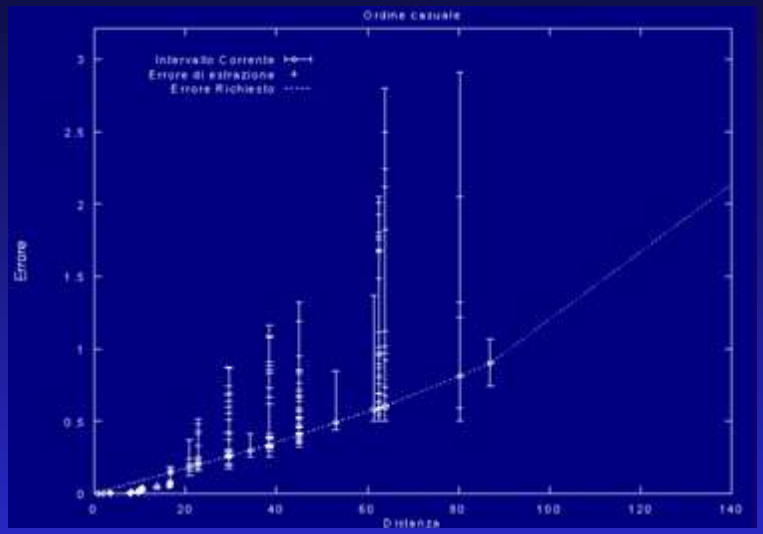
$$\bigcap_s \left[b_i..d_i\right) = \left[\max(b_i)..\min(d_i)\right) \neq \varnothing$$

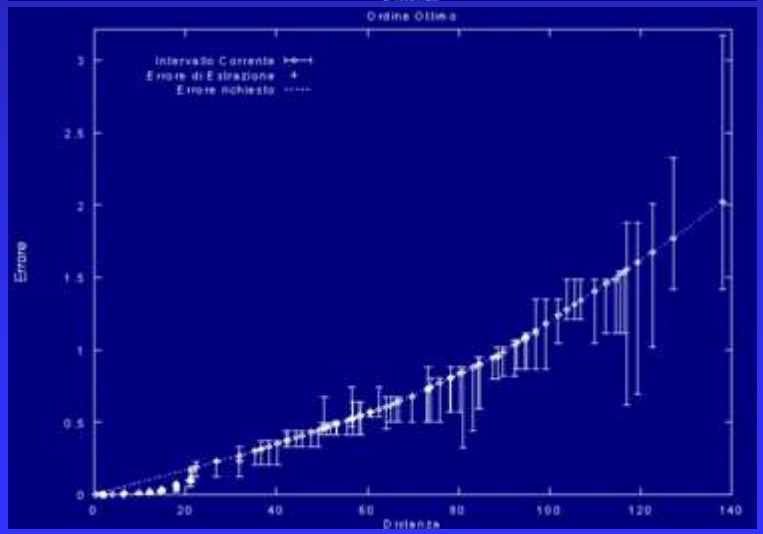
- Implementazione per mezzo di una doppia coda di priorità (heap)



Ordine di visita

- Scopi:
  - Minimizzare il numero di spigoli sul fronte
  - Uniformare l'intervallo corrente all'errore richiesto
- Possibili ordinamenti
  - Casuale
  - Per distanza
  - Per errore
  - Ordine misto distanza-errore
  - (Gestione delle "bolle")



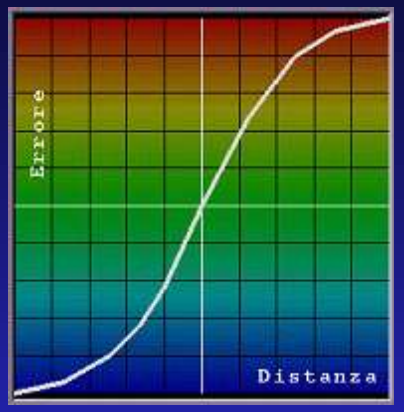


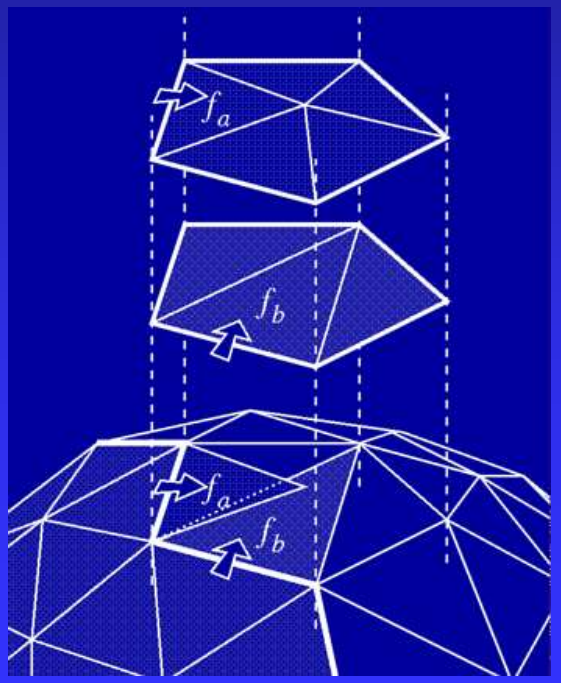
Risoluzione Variabile

- Errore variabile durante la visita
- Condizione di consistenza:

$$\bigcap_s \left[b_i..d_i\right) = \left[\max(b_i)..\min(d_i)\right) \neq \varnothing$$

  - Implementazione per mezzo di una doppia coda di priorità (heap)
- Ordine di visita
  - Scopi:
    - Minimizzare il numero di spigoli sul fronte
    - Uniformare l'intervallo corrente all'errore richiesto
  - Possibili ordinamenti
    - Per distanza
    - Per errore
    - Ordine misto

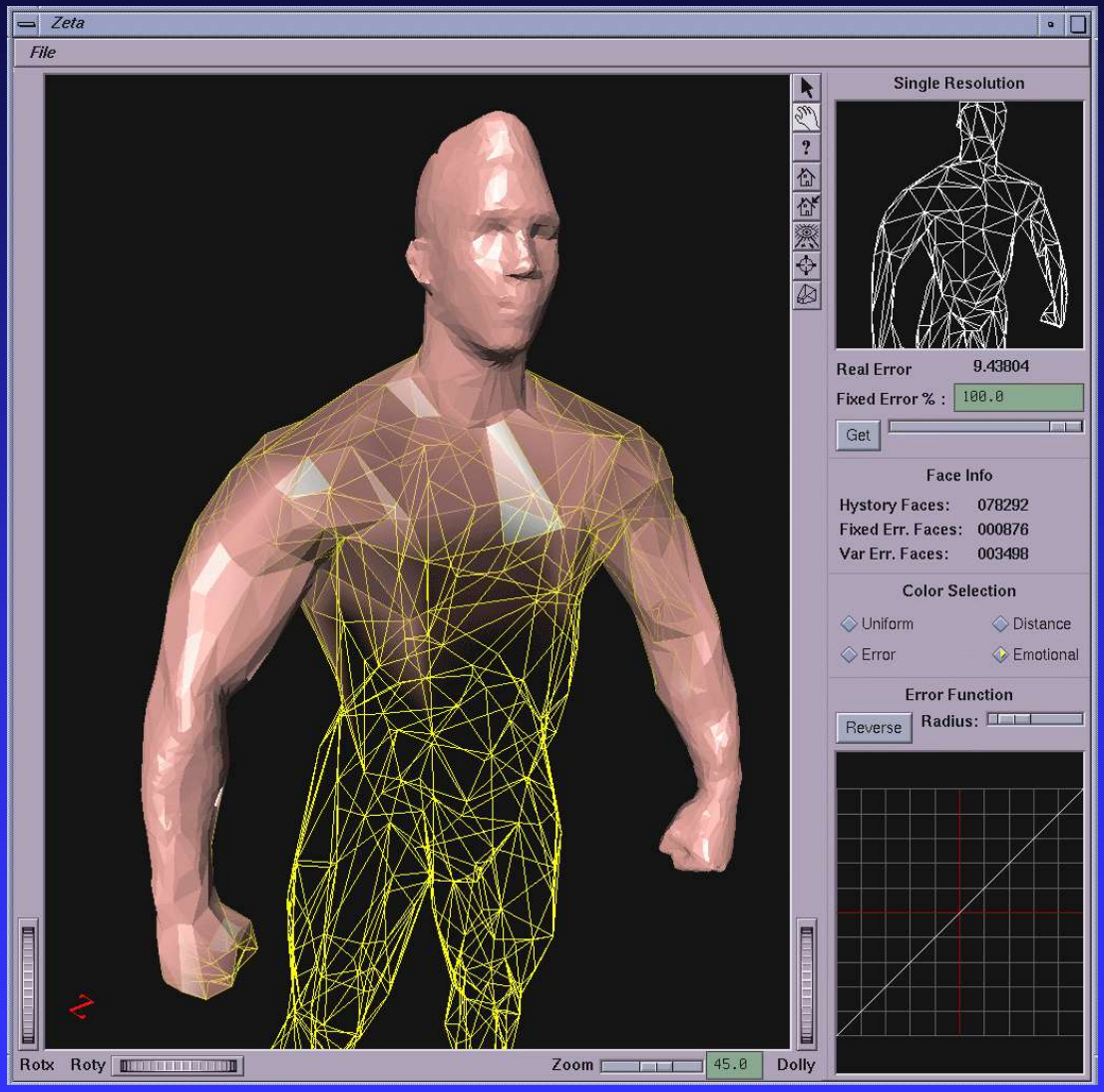




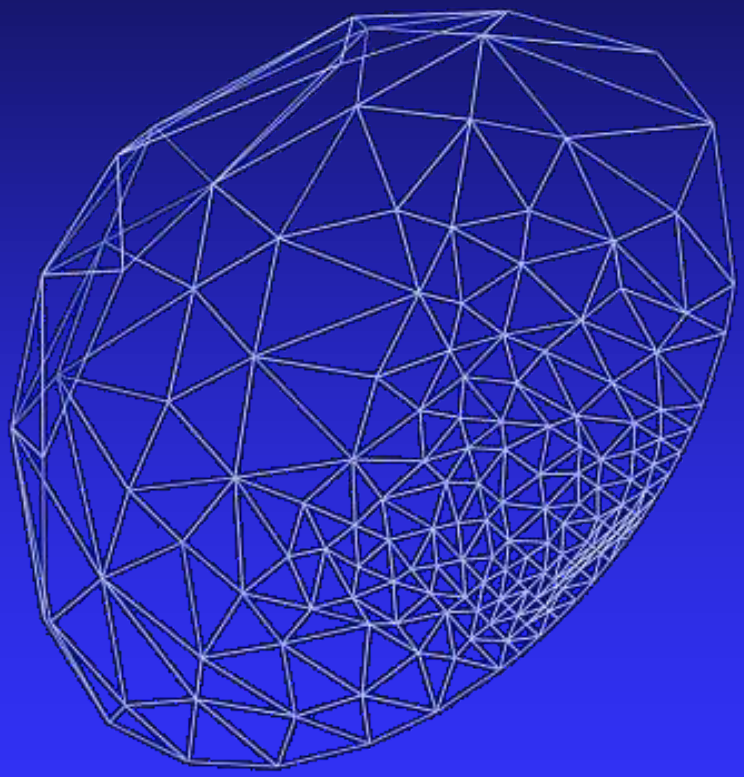
Interpretazione semantica della risoluzione

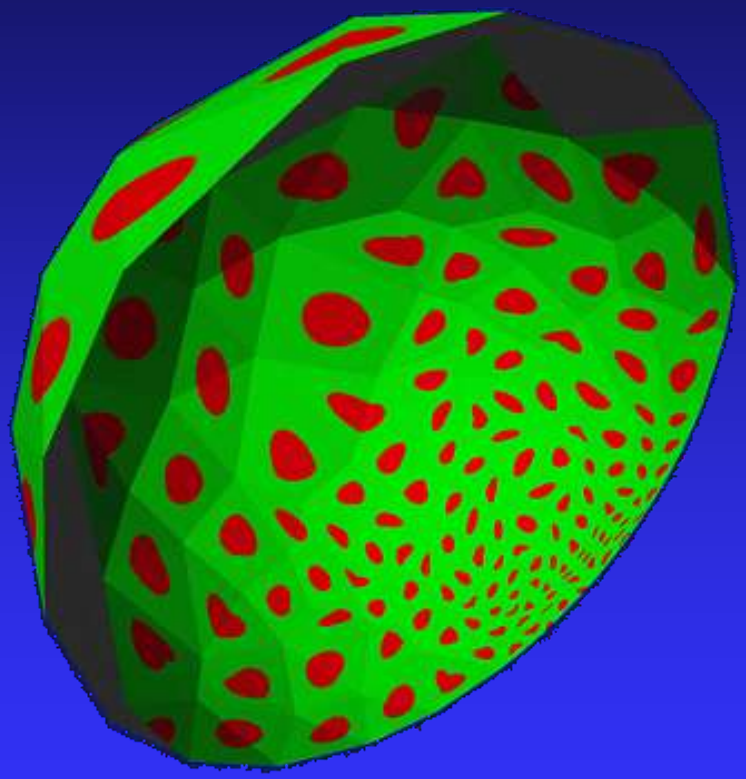
- Modalità :
  - Semitrasparenza
  - Shaded - Wireframe
  - Texture mapping
  - Shaded - spigoli vivi

Visualizzazione Shaded-Wireframe

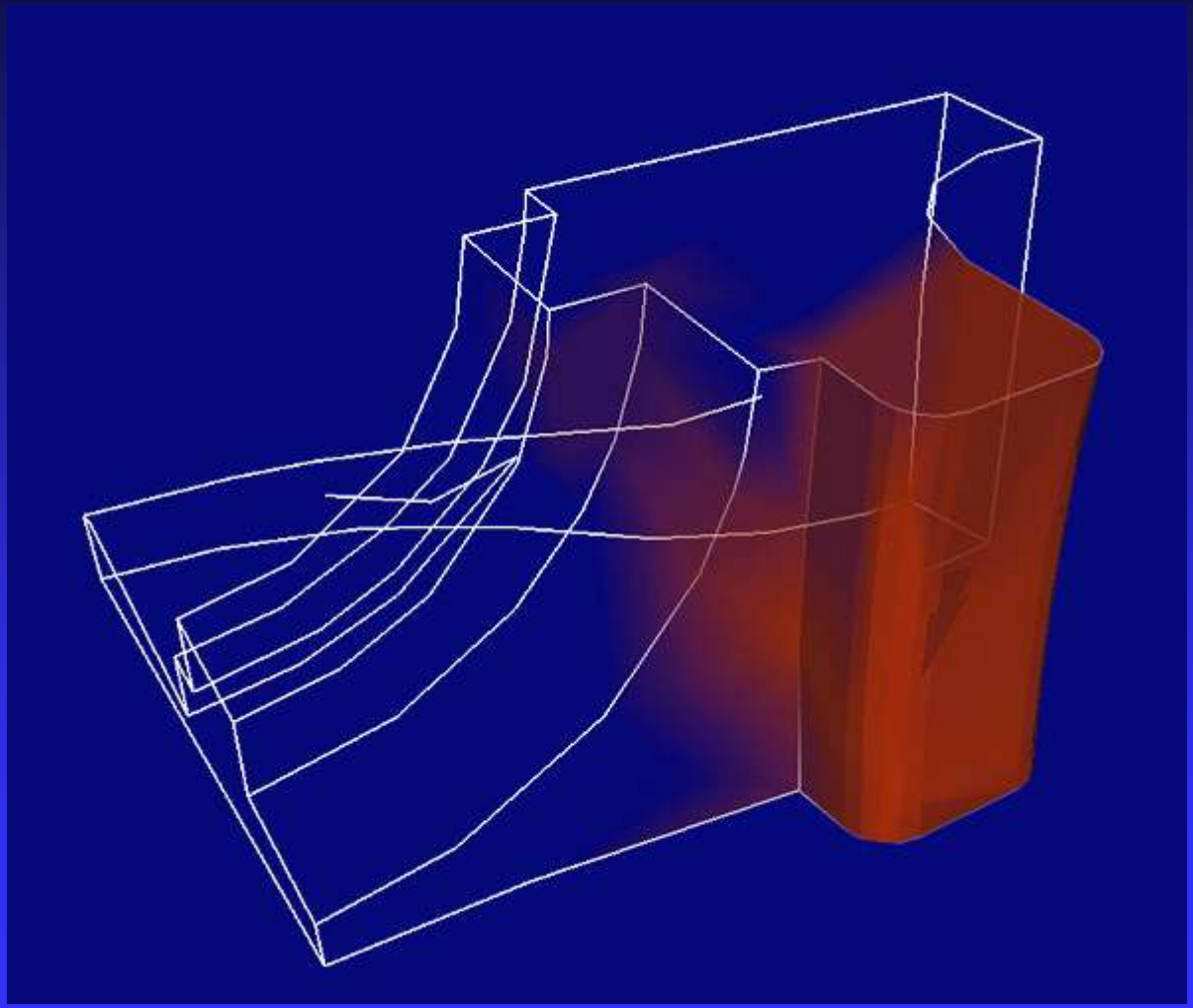


Visualizzazione Texture Mapping





Visualizzazione Spigoli



Conclusioni

- Contributi
  - Un nuovo paradigma: Resolution Modeling
  - Definizione di algoritmi e strutture dati che permettono:
    - Estrazione efficiente a *risoluzione fissa*
    - Garanzia della *continuità* C^0 delle superfici
    - *Composizione* di operazione di editing della risoluzione
    - Risposta in *tempo reale*
  - Loro realizzazione prototipale
- Sviluppi
  - Strutture dati di grandi dimensioni: gestione su memoria secondaria o su rete