

Politecnico di Milano – Facoltà del design  
Laboratorio di computer grafica C2  
Prof. Michele Zannoni / Prof. Giovanni Gigante

## VRML per gli utenti di Maya

Versione 0.9 (16 luglio 2003)

### Introduzione

Il VRML (Virtual Reality Modeling Language) è un formato per la rappresentazione ed esplorazione di geometrie nello spazio consultabili tramite browser internet in grado di gestire sia oggetti statici che dinamici, eventualmente consentendo di associare a talune parti della scena dei comportamenti che permettendo all'utente di manipolare gli oggetti con ad esempio il mouse.

Il VRML fu dall'inizio concepito come formato comune di interscambio di geometrie 3D multiplatforma e consultabile tramite web con opportuni plug-in per browser liberamente sviluppabili da qualunque società interessata.

Come nella maggior parte dei motori di rendering in tempo reale, i solidi vengono gestiti come *mesh* ossia maglie di vertici a loro volta definiti da tre coordinate cartesiane e uniti a formare una "rete" che costituisce la superficie dell'oggetto. Tale soluzione offre il vantaggio di una rapida elaborazione in quanto tutte le facce sono piane e richiedono di conseguenza minori calcoli. Per contro tuttavia tutte le superfici curve, generalmente definite nei software di modellazione a partire dalle equazioni matematiche (di norma di 3° o 5° grado) delle linee generatrici (usualmente di tipo NURBS), devono essere ricondotte in fase di generazione del file VRML a superfici sfaccettate tipiche dei solidi euclidei. Svantaggio della scelta delle geometrie mesh è quindi dato dalla necessità di impiegare per approssimare adeguatamente superfici curve un grande numero di facce (e di conseguenza vertici) che vanno a richiedere consistenti quantità di memoria per gestire la scena.

Sebbene la visualizzazione dipenda dal particolare plug-in adottato tra i vari disponibili, generalmente i solidi sono resi a video con una rappresentazione di tipo *gouraud*. Questo significa che gli oggetti vengono mostrati con un sistema *scan-line* dove la viewport con cui si inquadra la scena viene scandita virtualmente con un raggio parallelo all'asse visivo che va a determinare il colore della prima faccia che incontra in funzione del materiale ad essa associato, della sua inclinazione (ossia il piano di giacitura) e dell'intensità di luce che la interessa. A parità di materiale e illuminazione bisogna sottolineare che tutte le superfici che hanno medesima giacitura (ossia si trovano su piani tra loro paralleli) avranno uguale colore. Questo sistema non consente inoltre di gestire le mutue riflessioni tra le superfici e nemmeno le ombre portate (rappresentazioni tipiche dei rendering in modalità ray-tracing).

Sfortunatamente i principali software di modellazione e rendering oggi impiegati come Maya, 3d studio max, Rhino, ecc., non sempre gestiscono il formato VRML in maniera ottimale. Essenzialmente infatti vengono utilizzate le funzionalità più comuni (in genere disponibili nella maggior parte dei formati con cui è possibile esportare una geometria spaziale):

- tutti i solidi sono ricondotti a insiemi di facce triangolari senza sfruttare le altre primitive grafiche disponibili
- nel caso di solidi impiegati in maniera seriale non si ricorre all'uso dei prototipi o delle istanze ma si duplicano ripetutamente tutti i dati relativi all'oggetto
- limitato uso degli interpolatori
- sensori non disponibili
- programmazione tramite script non disponibile

Per sfruttare al meglio le possibilità del VRML quindi bisognerebbe ricorrere ad ambienti di modellazione specifici per questo formato oppure intervenire direttamente nel file sorgente.

## Cenni sull'organizzazione del formato

L'organizzazione interna del formato denota la volontà di consentire l'interscambio tra piattaforme e la consultazione via web. I file infatti, come le pagine internet, sono dei normali documenti di testo (ASCII o Unicode utf8) strutturati secondo una propria sintassi che consente un'immediata lettura del codice sorgente da parte dell'uomo senza la necessità di alcun particolare software. Per contro tuttavia generalmente un file VRML standard ha un peso in termini di Mb superiore a quello di altri formati sviluppati più recentemente per la rappresentazione 3D sul web come Shockwave 3D di Macromedia e Viewpoint che sono leggibili sono con gli appositi plug-in prodotti dalle società detentrici dei diritti. Per il VRML esistono invece differenti software che ne consentono la visualizzazione, liberamente utilizzabili, anche se che in genere non garantiscono tra loro una perfetta corrispondenza in termini sia di prestazioni che di qualità visiva.

Essendo pensato per la consultazione via web, il VRML ha il vantaggio di non definire a priori le caratteristiche fisiche (come ad esempio la risoluzione o particolari periferiche di input per la consultazione ed esplorazione dello spazio) che il computer dovrà avere, delegando al plug-in e al suo motore di rendering il compito di restituire in maniera adeguata le informazioni presenti nel file.

Ogni elemento della scena è rappresentato da un *nodo* al quale può o meno essere associato un nome univoco nel file per consentirne il riutilizzo. Il nodo è composto dalla identificazione del tipo (es. forma, materiale, luce, gruppo, suono...) e dall'elenco delle sue proprietà racchiuse tra parentesi graffe. Ogni proprietà può ammettere solo determinati valori in funzione del tipo di dati che rappresenta. Il VRML definisce come tipo di dati valori booleani, stringhe di testo, numeri interi o decimali, matrici di rotazione (composte da tre coefficienti, uno per ogni asse, da moltiplicare per un angolo espresso in radianti<sup>1</sup> per ricavare la rotazione complessiva nello spazio 3D), lista 2D o 3D, riferimenti a nodi, definizione di colori (composte dalle tre componenti rgb espresse ognuna con un valore compreso tra 0 e 1). A seconda dei casi ogni proprietà può essere definita come singolo valore oppure come elenco (matrice) di elementi dello stesso tipo. I componenti delle matrici vengono racchiusi tra parentesi quadre mentre le stringhe di testo vanno racchiuse tra virgolette (").

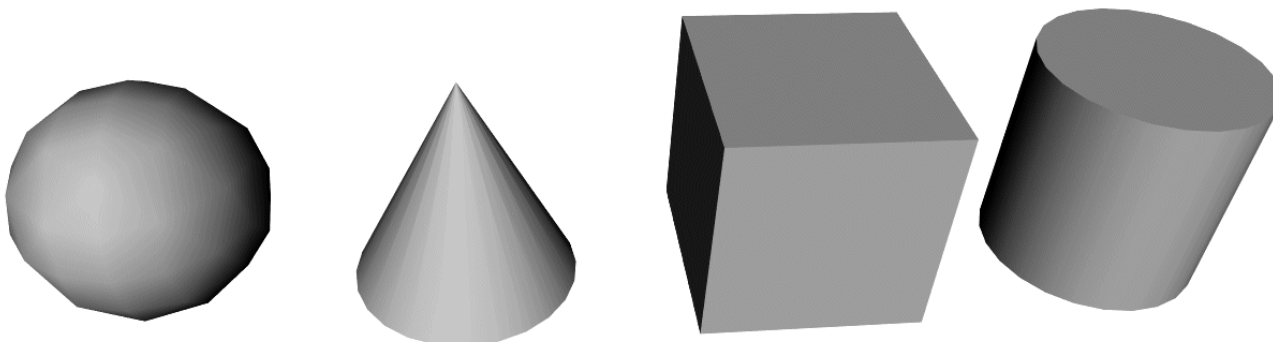
I commenti possono essere inseriti nel codice preceduti dal carattere #.

```
# commento
Transform {                               # tipo di nodo: trasformazione
  scale 2 0.5 2                             # lista 3D
  rotation 0 1 0 0.78                       # matrice di rotazione. Corrisponde ad una rotazione di circa 45° sull'asse y
  translation 0 -1.5 0                       # lista 3D
  children [                                 # matrice di nodi
    Shape {                                  # tipo di nodo: forma
      appearance Appearance {              # riferimento a un nodo
        texture ImageTexture {             # riferimento a un nodo
          url "brick.jpg"                   # stringa di testo
        }
      }
    geometry Box {}                         # riferimeno a un nodo
  ]
}
```

<sup>1</sup> Per ottenere l'angolo in radianti moltiplicare la misura espressa in gradi per pi greco e dividerla per 180.

## Primitive geometriche

Il VRML definisce una serie di primitive grafiche semplici tramite cui costruire la scena, queste sono la sfera, il cilindro, il cono, il parallelepipedo e il griglia di elevazione a cui si aggiunge l'indispensabile faccia 3D<sup>2</sup>.

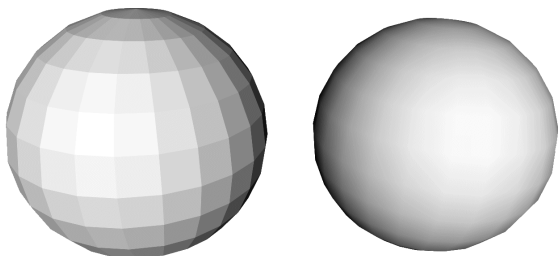


*Esempio delle primitive grafiche messe a disposizione dal VRML*

Per ciascuna delle primitive solide è possibile definire le dimensioni caratteristiche ed applicargli le più comuni funzioni di trasformazione geometrica come la traslazione, rotazione e scalatura. Con questi strumenti, assieme eventualmente alle funzioni di estrusione, è possibile rappresentare qualunque oggetto ricondotto a solido euclideo (costituito ossia da facce triangolari) come richiede la gestione tramite mesh.

Al fine di ottimizzare la gestione delle mesh, considerando che le facce condividono tra loro gli spigoli e quindi i vertici, il VRML memorizza l'elenco delle coordinate di tutti i vertici del solido e l'ordine con cui questi saranno usati, anche ripetuti, per definire le facce tridimensionali. Ciò consente in genere di ridurre il numero di coordinate da memorizzare evitando i doppioni.

Qualora il software di modellazione consenta la scelta, è bene specificare in fase di salvataggio di mantenere per le mesh derivate da superfici curve le direzioni delle normali al fine di ottenere nella visualizzazione la sensazione di levigatezza malgrado la geometria sia sfaccettata.



*Confronto tra una sfera (gestita come insieme di facce) esportate senza normali (a sinistra) e con le normali (a destra).*

Nel caso si sia applicato all'oggetto dei materiali con texture bisognerà inoltre esportare anche le coordinate della mappatura per mantenere la corretta distribuzione delle immagini sulle superfici dei modelli.

## Punti di vista e scena

Per ogni file è possibile definire un qualunque numero di punti di vista che sono assimilabili a delle camere di cui vanno specificati il punto di origine, la direzione di osservazione e il tipo di obiettivo che determina l'ampiezza della visuale (da zoom a grandangolo) e di conseguenza il livello di deformazione prospettica a cui i solidi sono soggetti.

Per rendere più realistica la scena e ambientare il modello è possibile specificare il colore di sfondo oppure inglobare il solido all'interno di un cubo virtuale le cui facce sono costituite da 6 immagini che rappresentano il paesaggio circostante.

<sup>2</sup> Come accennato tuttavia la maggior parte dei software di modellazione non espressamente designati per il VRML utilizzano la faccia 3D come unica primitiva geometrica.

## Materiali e sorgenti luminose

Ad ogni oggetto è possibile associare le proprietà ottiche del materiale di cui esso è costituito, nonché definire per la scena le sorgenti di illuminazione.

Un materiale viene caratterizzato dal colore, dal modo di riflettere e/o trasmettere la luce e dalla texture della superficie.

Riferendosi agli attributi di un nodo *Material* del vrml le proprietà fisiche sono:

<i>diffuseColor</i>	il colore normale dell'oggetto
<i>specularColor</i>	il colore dei riflessi
<i>emissiveColor</i>	colore che viene emesso considerando il solido come una sorgente luminosa. In questo caso non proietta comunque nessuna ombra su altri solidi
<i>ambientIntensity</i>	quantità di luce ambientale che l'oggetto riflette
<i>Shininess</i>	capacità di riflettere la luce
<i>transparency</i>	grado di trasparenza (non tutti i browser possono supportare questa funzione)

Come texture è possibile definire un'immagine di tipo .jpg o .png da utilizzare per rivestire la geometria. Il VRML non consente invece le mappe di rilievo.

Come qualunque software di rendering, anche la visualizzazione di un file VRML richiede la presenza di qualche sorgente luminosa. Spesso, anche per migliorare le prestazioni, si può rinunciare ad avere delle sorgenti di luce e sfruttare la funzione della maggior parte dei plug-in che consente di attribuire a tutti i solidi un certo grado di emissione luminosa. Ciò permette di esplorare tutti i lati del solido con la sicurezza di averli illuminati in maniera uniforme. Per contro le superfici piane vengono appiattite a campiture omogenee differenziate unicamente, a parità di materiale, dall'angolo tra il piano di giacitura e il quadro di visualizzazione.

Il VRML mette altrimenti a disposizione delle sorgenti di luce puntuali direzionali a raggi paralleli (come il sole o sorgenti molto lontane) o divergenti (come lo spot) oppure unidirezionali. Per il tipo di visualizzazione che offrono i vari plug-in, non sono supportate le ombre portate. Questo fa sì che la luce "attraversi" gli oggetti impedendo di avere zone d'ombra sulle facce rivolte verso la sorgente anche se coperte da altri oggetti.

## Tipi di sensori – interazioni

Il VRML definisce una serie di "sensori" che consentono di intercettare e indirizzare l'utente nella manipolazione degli oggetti. Il meccanismo dei sensori si basa sullo scambio tra le varie parti della scena di eventi che si verificano ad intervalli di tempo regolari oppure in seguito alle azioni dell'utente. Ogni sensore è capace di generare degli eventi in uscita e/o di riceverne in entrata. La gestione degli eventi consiste nell'associare un evento in uscita con quello in entrata di un altro oggetto/sensore. In questo modo quindi è possibile collegare ad esempio il movimento del mouse su di un oggetto con l'esecuzione di un suono o l'avvio di una sequenza animata.

### **Sensori temporali**

Esiste un unico sensore in questa categoria che a differenza di tutti gli altri non è associato a qualche geometria o alla posizione dell'utente nella scena.

<i>TimeSensor</i>	Questo sensore funziona esattamente come un timer emettendo un evento in uscita ad intervalli di tempo regolari. E' possibile stabilire il momento in cui far avviare e/o fermare il timer, decidere se generare eventi ripetutamente ogni certo numero di secondi oppure una sola volta. Il sensore è inoltre in grado di emettere degli eventi in uscita intermedi tra un ciclo e l'altro indicando la percentuale di periodo trascorsa. Questo risulta particolarmente utile per la gestione degli interpolatori ( <i>vedi oltre</i> ).
-------------------	--

## **Sensori ambientali**

I sensori d'ambiente generano eventi in funzione della posizione dell'osservatore all'interno della scena.

- VisibilitySensor* Il sensore di visibilità genera un evento quando l'area a cui è associato rientra nell'attuale finestra di visualizzazione. Questo strumento è particolarmente utile per ottimizzare le risorse del computer in scene con numerose animazioni in quanto è possibile interrompere selettivamente quelle al di fuori dell'area visualizzata e riprenderle al cambiamento del punto di vista.
- ProximitySensor* Il sensore di prossimità genera invece un evento quando l'osservatore entra, o esce, nella porzione di spazio ad esso associato.
- Collision* Questo sensore intercetta le collisioni dell'osservatore con i solidi della scena. E' possibile considerare come superficie di collisione sia quella reale dell'oggetto oppure un'altra semplificata qualsiasi consentendo di ottimizzare le risorse necessarie per rilevare gli scontri.

## **Sensori di manipolazione**

Questo tipo di sensori consente di interagire con gli oggetti della scena ruotandoli o traslandoli nello spazio secondo i limiti prefissati dall'autore.

- TouchSensor* Questo sensore permette di catturare il movimento del mouse sulla superficie del solido a cui è associato. In funzione delle proprietà settate è possibile intercettare il movimento sulla superficie oppure il semplice stazionamento, oltre naturalmente la pressione di un tasto del mouse.
- SphereSensor* Questo sensore consente di ruotare un oggetto attorno ad un centro. La rotazione avviene cliccando nell'area sferica gestita dal sensore e trascinando il cursore
- CylinderSensor* Analogamente al precedente sensore, anche questo consente la rotazione della geometria tramite il mouse, ma limitatamente ad un asse. E' inoltre possibile indicare di quanti gradi ammettere la rotazione.
- PlaneSensor* Questo sensore permette di traslare gli oggetti su di un piano bidimensionale. E' possibile anche in questo caso limitare il movimento tra dei valori minimi e massimi per le due direzioni, ed eventualmente vincolare il movimento su di un solo asse.

## **Animazioni**

In VRML è possibile animare la scena modificando la posizione degli oggetti nel tempo oppure alterandone le caratteristiche geometrico/materiche.

Strumenti particolarmente utili per l'animazione sono i nodi di interpolazione che permettono di variare tramite interpolazioni lineari tra degli estremi determinate caratteristiche della scena al passare del tempo. Le interpolazioni possono essere utilizzate per cambiare la geometria (le coordinate), l'orientamento, la posizione, la scala, il colore e la direzione delle normali. Per ogni interpolatore è possibile definire una serie n di stati a cui dovrà essere impostata la proprietà del solido ad esso collegato e stabilire inoltre il tempo necessario per passare da uno stato all'altro.

Un metodo di animazione più complesso consiste nel combinare assieme dei timer e script realizzati in VRMLScript (una variante del javascript) con cui manipolare le varie componenti della scena.

## **Uso di LOD (Level of Detail)**

Come altri software di esplorazione di ambienti virtuali, il VRML definisce un sistema per semplificare ed alleggerire la visualizzazione di oggetti in funzione della distanza da cui vengono osservati. E' possibile definire per ogni oggetto differenti livelli di dettaglio associati ciascuno ad un certo intervallo di distanze a cui l'osservatore si deve trovare. In questa maniera si può progressivamente alleggerire la visualizzazione della scena mano a mano che

l'osservatore si allontana riducendo i tempi di elaborazione ottenendo inoltre una visione più nitida<sup>3</sup>.

## Uso di INLINE

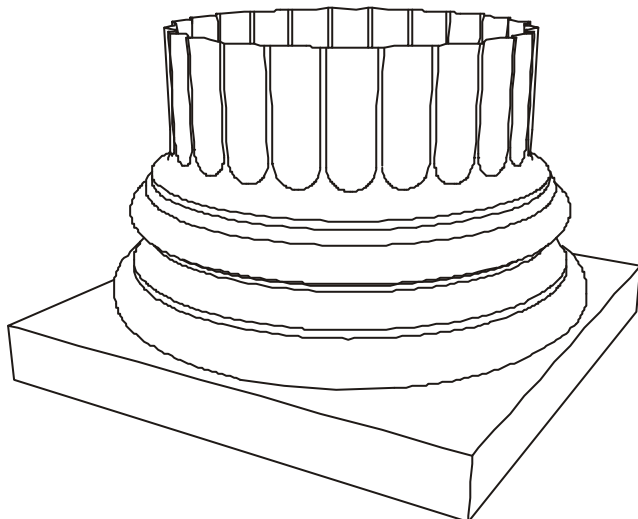
Il VRML offre un altro sistema per ottimizzare la gestione della geometria consentendo di definire delle vere e proprie librerie di oggetti utilizzabili ripetutamente in differenti file. Questo è consentito dalla possibilità di importare nella gerarchia della scena corrente altri file vrml specificandone semplicemente il percorso. Come a qualunque altra geometria è possibile applicare gli strumenti di manipolazione (traslazioni, rotazioni, scalature) permettendo di integrare tra loro le varie parti.

```
Transform {
  translation 5 2 0      # traslazione e scalatura applicata alle geometrie importate
  scale 2 2 2
  children [
    Inline {
      url "blocco1.wrl"  # riferimento al file da importare
    }
  ]
}
```

Vantaggio di questo approccio consiste nel poter gestire in maniera più razionale le geometrie della scena garantendo di avere sempre aggiornate tutte le copie di un solido usato in differenti progetti. Inoltre avere una scena distribuita in differenti parti permette di sfruttare meglio la banda passante della connessione scaricando in maniera asincrona le componenti del modello.

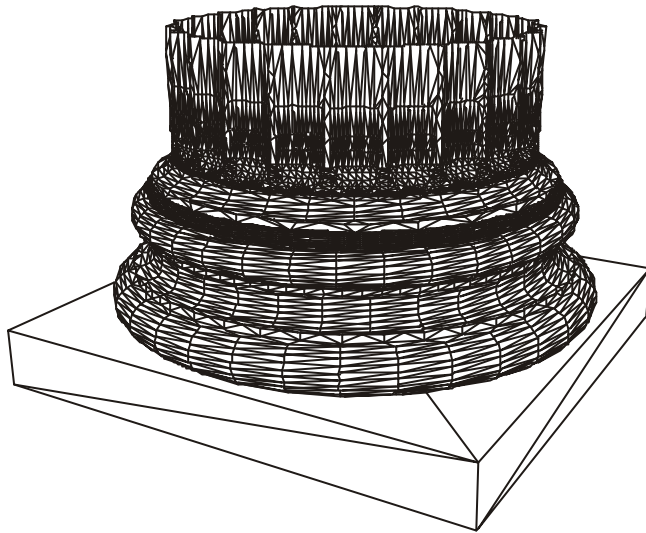
## Conversione da Nurbs a Mesh (tassellazione)

Sebbene molti programmi (tra cui Maya, 3d studio max, Rhino) consentono di generare automaticamente, ed in via provvisoria, le mesh dei solidi nurbs per l'esportazione in VRML, può in genere essere preferibile generare prima le mesh per tenere sotto controllo l'approssimazione a cui viene sottoposta la geometria in fase di tassellazione. Con tassellazione si indica appunto il processo con cui una geometria nurbs viene approssimata con una mesh.



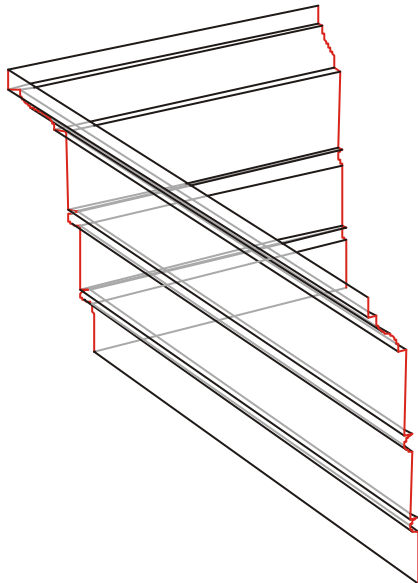
*Esempio di modello costituito da superfici NURBS.*

<sup>3</sup> Sfortunatamente la funzione *Level of Detail* di Maya (raggiungibile dal menu *Edit*) non viene mantenuta nell'esportazione ma tutti gli oggetti che ne definiscono il gruppo divengono simultaneamente visibili nel VRML.



*Lo stesso modello ridotto in una mesh di oltre 12'000 facce triangolari. Si può notare come siano essenzialmente le superfici curve a richiedere un grande numero di poligoni mentre quelle piane, a meno di profili molto articolati, vengono rappresentate con un ridotto numero di facce.*

In questa fase aspetto particolarmente importante da controllare è la complessità della mesh che viene generata a partire da superfici curve. Infatti se da un lato l'elevato numero di poligoni assicura un'ottima aderenza del solido "sfaccettato" con quello originario, per contro è causa di un incremento di occupazione di memoria e spazio su disco. Mentre infatti una superficie NURBS è rappresentabile con le equazioni matematiche delle curve generatrici, una mesh viene definita dalle coordinate spaziali di tutti i vertici che la compongono.



*Esempio di superficie NURBS complessa generata a partire dai tre profili messi in evidenza*

## Ottimizzazione delle geometrie per l'esportazione in VRML

La progettazione di uno spazio 3D dipende dal tipo di visualizzazione e consultazione che si vuole ottenere. Per l'esplorazione in locale infatti l'elemento discriminante principale sarà il grado di complessità della geometria mentre per la consultazione tramite web la quantità di kb che devono essere trasferiti.

In generale comunque è utile in ambedue i casi evitare di replicare inutilmente i dati relativi ad elementi utilizzati in serie nella scena. Soprattutto nella rappresentazione architettonica, non solo, l'ambiente è ottenuto dalla ripetizione di stessi elementi come colonne, capitelli ecc. In questi casi sarebbe auspicabile esportate tali oggetti individualmente su file distinti da caricare nella scena finale con istruzioni *inline*. In questo modo le varie copie vengono ottenute tramite rototraslazioni e scalature dell'oggetto di partenza.

## Criteri guida per la consultazione in locale

Per la consultazione locale non si pone strettamente il problema dell'occupazione di spazio su disco e nemmeno dei tempi necessari per il caricamento dell'intera scena. Fattore invece da valutare sarà il grado di complessità (dettaglio) da raggiungere nelle varie geometrie, il tipo di illuminazione e le animazioni, aspetti che vanno a gravare in particolare sul processore, scheda grafica e memoria. All'aumentare della complessità della scena il numero dei poligoni cresce richiedendo quindi sempre maggiori risorse e capacità elaborative.

## Criteri guida per la consultazione tramite web

Se si prevede una consultazione tramite web non si potrà conoscere a priori le caratteristiche hardware del computer finale e di conseguenza bisognerà non esagerare nelle richieste elaborative. Ma in particolare ciò che sarà da contenere al minimo sarà l'occupazione di spazio su disco e di conseguenza il tempo necessario a scaricare la scena dalla rete. Sarà necessario limitare perciò sia la complessità della scena (numero di poligoni) sia la dimensione e qualità delle texture.

In questo caso risulterà quindi particolarmente utile la funzione di importazione di più file per sfruttare connessioni asincrone parallele e la riusabilità delle geometrie senza dover duplicare l'occupazione fisica su disco.

Un altro sistema per alleggerire i file consiste nel limitare la precisione della geometria riducendo il numero di cifre decimali usate per indicare le coordinate dei vertici ed omettere le proprietà settate sui valori predefiniti.

Nel caso si utilizzi il plug-in Cortona di Parallelgraphics è possibile optare per un formato compresso<sup>4</sup> (estensione .wrz) che riduce di molto l'occupazione fisica delle geometrie della scena.

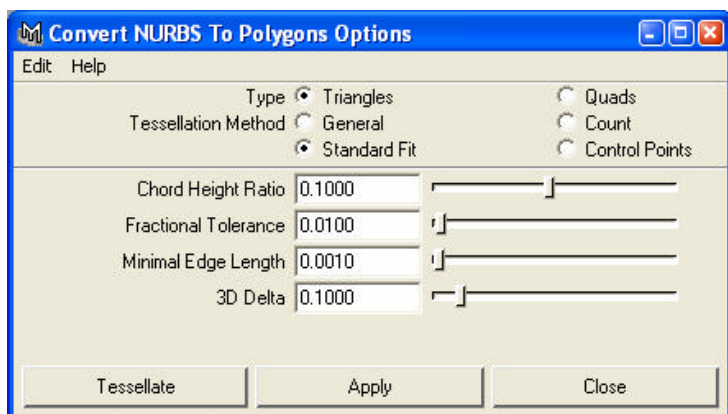
## Generazione di VRML da Maya

Di seguito vengono indicate alcune linee guida su come ottenere file vrml da Maya versione 4 o successiva.

### Tassellazione

Per la tassellazione di solidi NURBS Maya mette a disposizione il comando *NURBS to Polygons* raggiungibile dal menu *Modify>Convert*.

Come molti altri comandi di Maya, anche questo mette a disposizione numerose opzioni di configurazione<sup>5</sup>:



Opzioni per la conversione di superfici NURBS in mesh.

*Type* indica il tipo di facce che verranno create (triangolari o quadrate). E'

<sup>4</sup> La compressione è basata sul formato standard zip che ha un'ottima efficacia su file di tipo testuale.

<sup>5</sup> In riferimento alle opzioni disponibili si veda anche la guida in linea di Maya.

consigliabile preferire le facce triangolari nel caso di superfici a doppia curvatura estremamente complesse.

*Tessellation Method*

consente di scegliere il tipo di tassellazione da eseguire. In funzione della scelta verranno messe a disposizione diverse opzioni. Ogni metodo offre varie opportunità di ottimizzare le mesh e la scelta tra una modalità è un'altra dipenderà dalla particolare geometria nurbs di partenza.

*Standard Fit*

questo metodo di tipo "adattivo" esegue la tassellazione del solido fino a soddisfare i criteri specificati nelle opzioni.

*Chord Height Ratio* indica la distanza massima tra due poligoni, più piccolo è il suo valore e maggiore sarà il numero di facce create.

*Fractional Tolerance* indica il grado di approssimazione ammissibile tra la superficie originaria e quella interpolata. Maggiore è tale valore e minore sarà il numero di facce create, ciò potrà però determinare una minore corrispondenza della nuova geometria con la vecchia.

*Minimal Edge Length* misura minima dello spigolo delle facce. Minore è il suo valore e maggiore sarà l'accuratezza della mesh e il numero di poligoni generati.

*3D Delta* specifica l'iniziale distanza tra le curve isoparametriche U e V a partire dalla quale viene generata la maglia con cui è effettuata la tassellazione.

*General*

questo metodo crea in maniera uniforme una serie di poligoni in funzione dei valori U e V specificati.

*U Type/V Type* indica il tipo di scansione da adottare nelle direzione U e/o V.

*Number U/Number V* indica il numero di suddivisioni da dare lungo la direzione U e/o V.

*secondary tessellation controls* consente di affinare ulteriormente la mesh usando delle opzioni analoghe a quelle del metodo *Standard Fit*.

*Count*

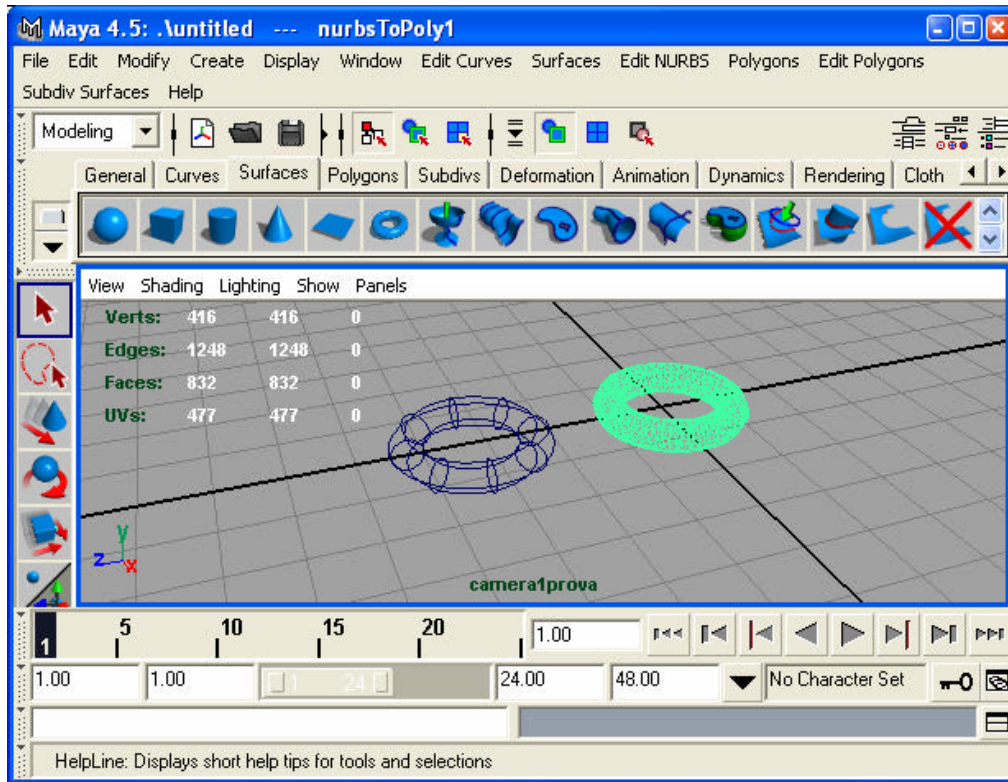
questo metodo permette di specificare semplicemente il numero di facce con cui schematizzare la geometria.

*Control Points*

genera la mesh con maglia quadrata (indipendentemente da quanto eventualmente specificato dall'utente) a partire dai punti di controllo della superficie NURBS.

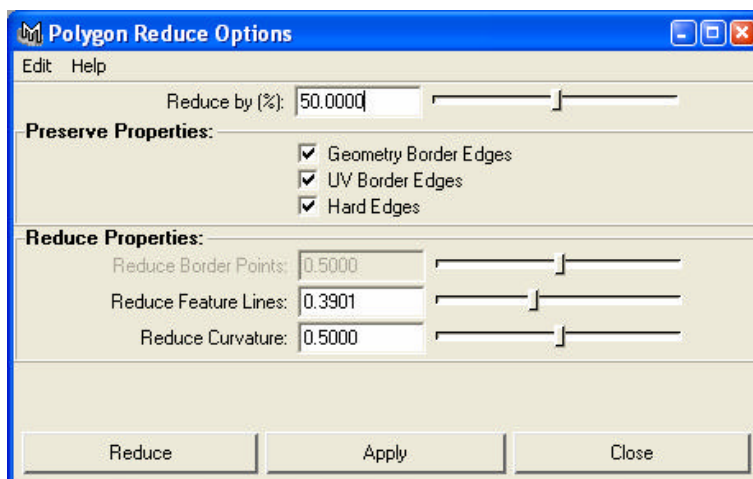
E' in genere consigliabile provare più di un metodo di tassellazione per trovare quello che dà i migliori risultati in base alla geometria che si intende processare.

Per avere indicazione sul numero di poligoni generati è disponibile la voce *Poly Count* del menu *Display > Heads Up Display*.



Con il comando *Poly Count* viene mostrato a video la statistica dei poligoni presenti nell'intera scena e, in seconda colonna, quelli dell'oggetto selezionato.

Maya offre vari strumenti con cui manipolare le mesh, tra cui la possibilità di ridurre il numero di poligoni con il comando *Reduce* del menu *Polygons*.

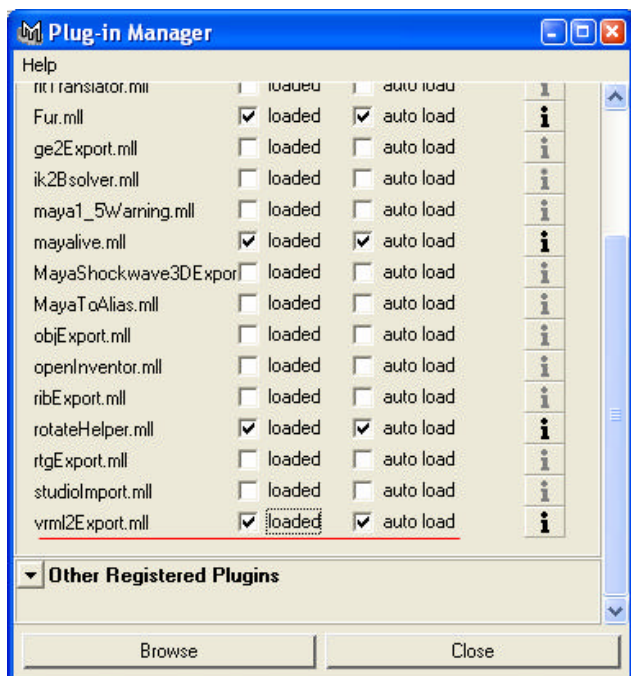


Opzioni messe a disposizione dal comando *Polygon Reduce*.

Questo comando consente in taluni casi di contenere il numero di facce, anche se bisogna prestare attenzione che la geometria non venga eccessivamente deformata.

## Configurazione del plug-in per l'esportazione

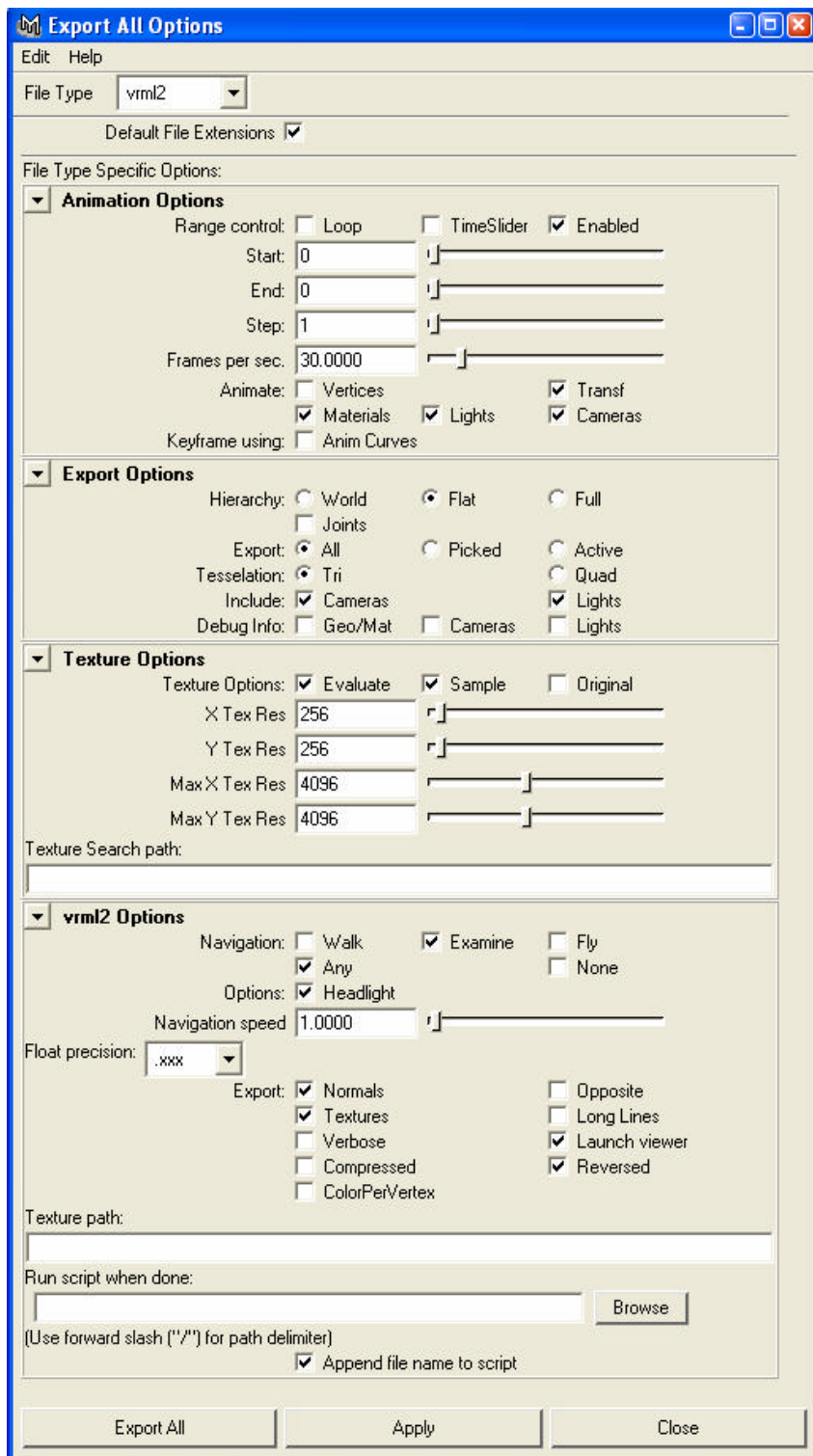
Maya viene fornito di un apposito plug-in che consente di esportare le geometrie create nel formato VRML. Normalmente tale plug-in non viene caricato all'avvio del programma ma richiede di essere eseguito manualmente. In questo caso è sufficiente aprire il *Plug-in Manager* dal menu *Window>Settings/Preferences* e caricare il plug-in *vrml2Export.mll* (eventualmente indicando di caricarlo automaticamente ad ogni riavvio di Maya).



*Il Plug-in Manager con evidenziato il modulo per l'esportazione in vrm12 da caricare.*

Questo consentirà di esportare i modelli creati semplicemente selezionando il comando *Export All* o *Export Selection* dal menu *File* e scegliendo il formato *vrml2*.

Il filtro consente di specificare una serie di opzioni relative alle modalità di esportazione, le prime tre parti contengono le impostazioni per le geometrie e i materiali, mentre l'ultima è esplicitamente riservata al vrml<sup>6</sup>.



Finestra con le opzioni per l'esportazione nel formato vrml.

Nella sezione *Animation Options* vengono specificate le opzioni per esportare le animazioni create in Maya:

- Enabled*                    indica se esportare l'animazione della scena.
- Loop*                        indica se ripetere ciclicamente l'animazione.
- TimeSlider*                indica se usare le informazioni sull'animazione impostate nella scena invece

<sup>6</sup> Per ulteriori chiarimenti sulle singole opzioni del filtro si veda la guida in linea di Maya.

	di specificarle in fase di esportazione.
<i>Start End</i>	frame iniziale e finale dell'animazione.
<i>Step</i>	numero di frames che intercorrono ad ogni incremento.
<i>Frames per sec</i>	numero di frames da riprodurre per ogni secondo di animazione.
<i>Vertices</i>	indica se animare la posizione dei vertici degli oggetti.
<i>Transf</i>	indica se animare le trasformazioni geometriche degli oggetti.
<i>Materials</i>	indica se animare i materiali.
<i>Light</i>	indica se animare le luci.
<i>Cameras</i>	indica se animare le camere.

La sezione *Export Options* riporta le opzioni su come esportare le geometrie

<i>Hierarchy</i>	indica se e come esportare la gerarchia degli oggetti: <i>World</i> : nessuna gerarchia, tutti gli oggetti appartengono allo spazio globale <i>Flat</i> : unico livello di gerarchia <i>Full</i> : esportazione completa della gerarchia della scena.
<i>Joints</i>	indica se esportare i collegamenti gerarchici tra le parti, utile nel caso di strutture a "scheletro".
<i>Tessellation</i>	indica il tipo di tassellazione a cui sottoporre le geometrie nurbs: <i>Tri</i> : genera mesh a facce triangolari <i>Quad</i> : genera mesh a facce quadrangolari.
<i>Cameras, Lights</i>	indica se esportare le camere e/o le luci.
<i>Debug Info</i>	indica se generare in fase di esportazione delle informazioni aggiuntive sugli oggetti e materiali ( <i>Geo/Mat</i> ), camere ( <i>Cameras</i> ) e luci ( <i>lights</i> ).

La sezione *Texture Options* controlla il modo con cui vengono esportate le texture dei materiali.

Infine in *vrml2 Options* vengono indicati i settaggi specifici per il vrml:

<i>Navigation</i>	imposta il modo di esplorazione della scena: <i>Walk</i> : esplorazione che simula il cammino vincolando l'osservatore al piano zero del terreno <i>Examine</i> : esplora la scena consentendo di ruotarci attorno in tutte le direzioni <i>Fly</i> : analogo a Walk ma senza la limitazione al piano del terreno <i>Any</i> : consente qualunque tipo di esplorazione <i>None</i> : disabilita le funzioni di esplorazioni predefinite dal browser consentendo solo quelle settate nel file come i collegamenti ipertestuali.
<i>Headlight</i>	indica al plug in di illuminare omogeneamente tutta la scena (anche in assenza di definizione di sorgenti luminose) consentendo di osservare in luce qualunque lato degli oggetti.
<i>Navigation speed</i>	velocità con cui la scena viene rinfrescata e avvengono i passaggi da un punto di vista all'altro.
<i>Float Precision</i>	indica il numero di cifre decimali utilizzate per definire le coordinate dei vertici dei solidi, i coefficienti delle trasformazioni geometrici, le componenti dei colori dei materiali. Un maggiore numero di decimali consente una esportazione più fedele della scena (sebbene già tassellata) ma comporta anche un incremento delle dimensioni finali del file.
<i>Normals</i>	indica se esportare le normali alle superfici. Si consiglia la loro esportazione nel caso di superfici curve che altrimenti risulterebbero sfaccettate. L'esportazioni delle normali causa un aumento delle dimensioni del file.
<i>Opposite Textures</i>	inverte la direzione delle normali. esporta le texture dei materiali.
<i>Long Lines</i>	indica se generare il file vrml con linee di codice lunghe oltre 80 caratteri.
<i>Verbose</i>	indica se generare informazioni aggiuntive di controllo durante l'esportazione.
<i>Launch viewer</i>	esegue dopo l'esportazione il programma associato ai file vrml visualizzarlo.
<i>Compressed</i>	comprime il file vrml creato con gzip. Non tutti i plug-in consentono di

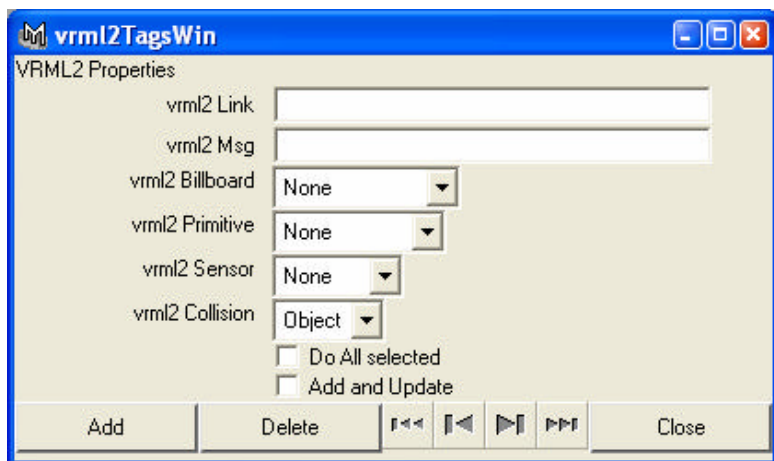
visualizzare file vrml compressi<sup>7</sup>.

*Reversed*            inverte la direzione delle superfici nurbs a singola faccia.  
*ColorPerVertex*    abilita l'esportazione di un colore per ogni vertice della geometria.

*Texture path*       percorso in cui salvare le immagini delle texture dei materiali.  
*Run script when*    esegue l'applicazione specificata alla fine dell'esportazione.  
*done*

## Attributi VRML

Maya oltre ad esportare la scena consente di associare alle geometrie una serie di attributi speciali che verranno considerati nella fase di salvataggio in vrml. Nella barra dei comandi di Maya immettere il comando `vrml2Tags`<sup>8</sup> che aprirà nuova finestra in cui sarà possibile aggiungere gli attributi specifici del vrml agli oggetti selezionati.



La finestra visualizzata dal comando `vrml2Tags` per associare agli oggetti gli attributi specifici del VRML.

Selezionato l'oggetto da modificare per la prima volta cliccare sul pulsante *Add* per aggiungere le proprietà vrml. Per rimuoverle cliccare invece su *Delete*. Nel caso di più oggetti selezionati utilizzare i pulsanti di scorrimento per passare dagli attributi di un oggetto a quelli del precedente/successivo<sup>9</sup>.

*vrml2 Link*            associa all'oggetto selezionato un link del tipo `http://www.url.com`. Nel caso sia selezionata una camera è possibile associarle come link un indirizzo del tipo `#name`, dove *name* è il nome attribuito alla camera, rendendola così disponibile nell'elenco dei punti di vista disponibili nel browser.

*vrml2 Msg*            messaggio descrittivo mostrato quando il cursore staziona sull'oggetto associato ad un link.

*vrml2 Billboard*     il Billboard in VRML rappresenta una particolare tipologia di oggetti che vengono visti sempre dal medesimo lato indipendentemente dal punto di vista dell'osservatore. In pratica il browser provvede dinamicamente a ruotare l'oggetto in concomitanza con l'angolo di osservazione dell'utente. Questa funzione è particolarmente utile nel caso di etichette di testo.

In Maya è possibile indicare se ruotare dinamicamente l'oggetto secondo il punto di vista solo rispetto l'asse X (*X Rotating*), solo rispetto l'asse Y (*Y Rotating*), oppure rispetto ambedue gli assi (*Screen aligned*).

*vrml2 Primitive*     indica di utilizzare una delle primitive predefinite del vrml (parallelepipedo, cono, cilindro, sfera, griglia di elevazione. Se si sceglie una delle primitive

<sup>7</sup> Affinché si possa comprimere il file è necessario che `gzip.exe` sia presente nel percorso di ricerca del sistema operativo.

<sup>8</sup> Il comando `vrml2Tags` corrisponde allo script `vrml2Tags.mel` presente in `scripts\others` nella cartella di installazione di Maya.

<sup>9</sup> Una volta associati gli attributi vrml all'oggetto, questi sono modificabili anche dall'*Attribute Editor* alla sezione *Extras Attributes*.

disponibili tutte le informazioni sulla geometria reale dell'oggetto verranno ignorate.

*vrml2 Sensor*  
*vrml2 Collision*

associa all'oggetto uno dei tipi di sensori vrml disponibili.

indica se impedire o meno le collisioni con l'oggetto selezionato.

*Do All Selected*

se attivato applica le modifiche a tutti gli oggetti selezionati e non solo a quello principale.

*Add and Update*

selezionare questa opzione per fare in modo che la pressione sul pulsante *Add* non crei semplicemente l'insieme vuoto predefinito di attributi vrml ma vengano subito applicati quelli al momento visualizzati nella finestra.