

Università degli Studi di Firenze
Corso di Laurea in Matematica
Anno Accademico 2003/2004

Tecniche di Rendering

Candidato: Claudia Amoruso
Relatore: Emanuele Paolini

Introduzione

Introduciamo il concetto di rendering come quel processo di creazione di un'immagine bidimensionale a partire da un modello tridimensionale; tale immagine deve tendere a rappresentare ciò che otterrei fotografando la scena tridimensionale a partire da un punto al suo interno.

E' inutile sottolineare l'importanza del processo di rendering, sia da un punto di vista concettuale, che didattico, che pratico. I campi d'applicazione sono i più disparati e vanno dalla progettazione CAD delle industrie meccaniche, al rendering d'ambienti interni per l'industria dell'arredamento e del mobile, fino alla simulazione di scene inesistenti in realtà come cartoni animati o effetti speciali di fantascienza. Ancora, si utilizza per la rappresentazione di molecole e proteine in campo bio-medico o per raffigurare una superficie matematica a partire dalla sua equazione.

In questa relazione prenderemo proprio spunto dal problema di rappresentare graficamente una determinata superficie, quella del Dini, per studiare le tecniche e i concetti legati al problema del rendering. In particolare ci concentreremo sul problema del calcolo dell'illuminazione degli oggetti (modelli d'illuminazione locali e globali, calcolo delle ombre) e del problema della rappresentazione interna degli oggetti (calcolo dei punti di intersezione di un oggetto con un raggio di luce e del vettore normale alla superficie dell'oggetto); infine andremo a rappresentare la superficie considerata tramite il programma Pov-ray (Persistence of Vision Ray Tracer) che utilizza le tecniche di ray tracing e radiosity.

Tecniche di visualizzazione

Consideriamo una generica scena tridimensionale costituita da un generico numero di oggetti e andiamo ad analizzare i vari tipi di approcci da adottare per la renderizzazione di tale scena.

In tutti gli algoritmi di visualizzazione normalmente usati vengono rappresentate tutte le parti componenti la scena comprese quelle che solitamente dovrebbero risultare invisibili. Per eliminare tali parti e creare così un'immagine più realistica, è necessario applicare un algoritmo per la rimozione delle linee nascoste (hidden line) o, nel caso più generale, delle superfici nascoste (**hidden surface**). Tali calcoli per l'eliminazione di elementi nascosti possono essere risolti utilizzando due tecniche fondamentali, denominate rispettivamente "Image-precision" e "Object-precision". Gli algoritmi che fanno parte della prima classe sono in grado di determinare, per ogni oggetto, quali parti dell'oggetto non sono oscurate da altri corpi nella scena; mentre la seconda classe comprende quegli algoritmi che, per ogni pixel, determinano qual è l'oggetto più vicino all'osservatore. Per la visualizzazione di un'immagine in modo realistico è necessario però tenere conto anche del fenomeno delle ombre proprie e portate che caratterizzano la scena. In generale, una volta identificate le superfici visibili della scena applicando un algoritmo per superfici nascoste (hidden-surface) si utilizza un modello d'illuminazione (**shading-model**) che descrive i fattori che determinano il colore di una superficie in un determinato punto, e quindi il colore del pixel corrispondente sullo schermo descrivendo le interazioni tra le luci e le superfici tenendo conto delle proprietà delle superfici stesse e della natura della radiazione luminosa incidente. Inoltre, oltre alla riflessione diretta della luce, si devono considerare anche le riflessioni reciproche tra gli oggetti della scena (o inter-riflessioni). In quasi tutti i modelli questo effetto viene simulato sotto forma di una illuminazione uniforme della scena, chiamata illuminazione **ambiente**, che fa sì che nessuna superficie risulti completamente scura. L'illuminazione ambiente si assume caratterizzata da un'intensità costante in ogni punto della scena. Il livello di realismo permesso dai diversi modelli di illuminazione varia in modo notevole, così come varia la complessità computazionale ad essi associata. Quindi le tecniche di visualizzazione che possiamo adottare sono diverse e si differenziano per la scelta dell'algoritmo hidden-surface e del modello di illuminazione; in questa relazione verrà approfondito il discorso sulla tecnica detta ray-tracing che

utilizza il modello d'illuminazione di Phong per modellare il fenomeno della riflessione; ma vediamo di accennare alcune altre tipiche tecniche di visualizzazione.

Z-Buffer

E' uno dei più semplici algoritmi per la determinazione delle superfici visibili. Consideriamo una scena tridimensionale costituita da un dato numero di oggetti modellati tramite un certo numero di poligoni.

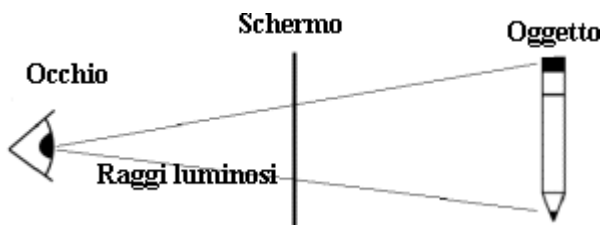
L'algoritmo si avvale dell'uso di due strutture dati quali lo z-buffer (un'area di memoria che mantiene per ogni pixel la coordinata z più vicina all'osservatore) ed il frame-buffer (che contiene le informazioni del colore relative ai pixel contenuti nello z-buffer). Per ogni pixel viene memorizzato il valore di z più grande (supponendo che l'asse z vada dallo schermo verso gli occhi dell'osservatore) e ad ogni passo il valore contenuto nello z-buffer viene aggiornato solo se il punto in esame ha la coordinata z più grande di quella attualmente nello z-buffer. Nel caso in cui lo z-buffer venga modificato, anche le informazioni relative al colore nel frame-buffer vengono aggiornate di conseguenza. La tecnica viene applicata ad un poligono per volta e dunque durante la scansione di uno di essi non è possibile disporre delle informazioni relative agli altri poligoni.

Scan line

Uno dei metodi più vecchi è quello scan-line, che fonde l'algoritmo di determinazione delle superfici visibili con quello di determinazione delle ombre riportate. Per ogni scan-line vengono proiettati rispetto alla sorgente luminosa gli spigoli di quei poligoni che fanno potenzialmente ombra ai poligoni intersecati dalla scan-line. Si parla perciò di spigoli d'ombra. Gli algoritmi che lavorano sulla linea di scansione sono image-precision e prendono il proprio nome dal fatto che per ogni scan-line determinano gli span (intervalli) di pixel visibili. Esso si differenzia dallo z-buffer poiché lavora con una linea di scansione alla volta.

Ray casting

Il ray casting e' un meccanismo image-precision che consente il rilevamento delle superfici visibili.



L'intero processo fa riferimento ad un centro di proiezione e ad uno schermo in posizione arbitraria pensata come una griglia regolare, i cui elementi corrispondono alla dimensione dei pixel della risoluzione desiderata. Si tracciano dei raggi immaginari di luce, dal centro di osservazione verso gli oggetti presenti nella scena, uno per ogni

cella della finestra stessa. Il colore di ogni pixel è settato all'oggetto più vicino in corrispondenza del punto di intersezione tra il raggio e gli oggetti trovati lungo il percorso dello stesso. Il cuore di questo meccanismo è l'operazione d'intersezione tra raggio e oggetto che vengono calcolate analiticamente.

Ray tracing

Il ray-tracing, che andremo ad analizzare approfonditamente tra breve, discende direttamente dal ray-casting al quale però viene applicato un particolare modello di illuminazione che tiene conto dei fenomeni fisici della luce quali riflessione e rifrazione che ci permette di raggiungere risultati fotorealistici sorprendenti.

Radiosity

E' un altro metodo image-precision che apporta ulteriori miglioramenti alle qualità fotorealistiche dell'immagine poiché tiene conto anche del fenomeno fisico della inter-riflessione tra gli oggetti.

Infatti, nel mondo reale, quando una superficie ha una componente di luce riflettente, essa non solo appare nella nostra immagine, ma illumina anche le superfici vicine.

La luce re-irradiata trasporta le informazioni sull'oggetto che l'ha rilanciata, in particolare il colore. Dunque, a causa di tale luce, le ombre risultano "meno nere" e si riesce a percepire il colore dell'oggetto vicino ben illuminato, un fenomeno spesso citato come "fuoriuscita di colore". L'algoritmo radiosity, come prima fase, individua e scompone le superfici in componenti più piccole e poi ne distribuisce l'energia luminosa diretta; come seconda fase calcola l'energia diffusa,

trasmessa e riflessa sull'ipotesi che le superfici riflettano luce allo stesso modo. Oltre a ciò, calcola le superfici che riflettono più energia e la ridistribuisce.

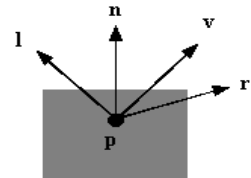
Modelli d'illuminazione

Ma per capire il funzionamento e le differenze dei vari metodi è necessario approfondire il discorso sui modelli di illuminazione.

Il modello d'illuminazione di Phong

Il modello di Phong è un semplice modello d'illuminazione diventato uno standard in computer graphics per il buon compromesso tra livello di approssimazione matematica dell'interazione fisica luce-superfici e il costo computazionale. Il modello di Phong è al tempo stesso efficiente e aderente alla realtà fisica, e consente di ottenere un buon *rendering* delle immagini, sotto un'ampia varietà di condizioni d'illuminazione e proprietà dei materiali. Nel modello di Phong, la radiazione luminosa si considera emessa da sorgenti puntiformi. Il colore di un punto arbitrario \mathbf{p} su una superficie viene calcolato utilizzando quattro vettori di modulo unitario:

1. il vettore \mathbf{n} è la normale in \mathbf{p} ;
2. il vettore \mathbf{v} è diretto dal punto \mathbf{p} verso il centro di proiezione (osservatore);
3. il vettore \mathbf{l} è diretto da \mathbf{p} verso la sorgente luminosa puntiforme;
4. il vettore \mathbf{r} , infine, è determinato dai vettori \mathbf{n} e \mathbf{l} , e ha la direzione che avrebbe un raggio riflesso perfettamente da \mathbf{l} .



Nel modello di Phong, la riflessione della radiazione luminosa viene modellata in termini di tre componenti additive, **diffusa**, **speculare** e **ambiente**, ciascuna modellata separatamente. La componente diffusa è direzionale, si riflette proporzionalmente all'angolo di incidenza con la superficie riflettente in tutte le direzioni e quindi è identica per tutti gli osservatori. La componente speculare è direzionale e si riflette lungo una direzione privilegiata (funzione della direzione di incidenza e della normale alla superficie). La componente ambiente si riflette in tutte le direzioni con uguale intensità ed è usata per schiarire le zone d'ombra. Si suppone inoltre che, ciascuna sorgente puntiforme abbia componenti separate per la luce diffusa, speculare e ambiente relative ai tre colori primari rosso, verde e blu. Per costruire il modello si suppone che sia possibile calcolare l'ammontare della luce incidente che viene riflessa nel punto di interesse. Il contributo di ciascuna sorgente luminosa si ottiene sommando le componenti ambiente, diffusa e speculare e l'intensità totale si ottiene infine sommando i contributi di tutte le sorgenti, a cui si aggiunge possibilmente un termine relativo alla luce ambiente globale. Il contributo all'intensità della luce sulla superficie si può esprimere come

$$I = I_a + I_d + I_s = R_a L_a + R_d L_d + R_s L_s$$

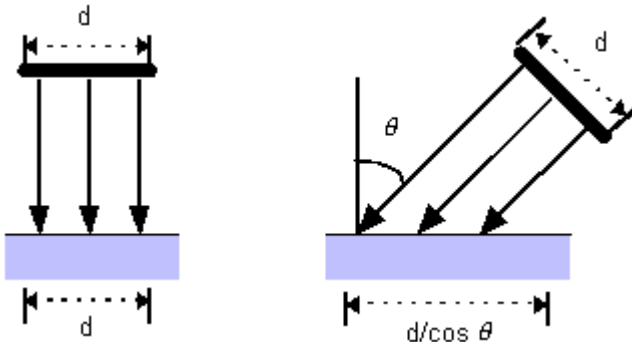
dove L rappresenta l'intensità della radiazione e R dipende dalle proprietà del materiale, dall'orientazione della superficie, dalla direzione della sorgente luminosa e dalla distanza tra la sorgente luminosa e l'osservatore.

Riflessione ambiente Nel modello di Phong il fenomeno dell'inter-riflessione diffusa tra gli oggetti della scena viene approssimato per mezzo della componente ambiente, che si assume distribuita uniformemente, con intensità L_a uguale in ogni punto della superficie. Parte di questa luce è assorbita, e parte è riflessa dalla superficie in base al valore del **coefficiente di riflessione ambiente** $R_a = k_a$. Poiché solo una frazione positiva della luce è riflessa, abbiamo $0 < k_a < 1$ e quindi $I_a = k_a L_a$.

Si osservi che l'assunzione di distribuzione uniforme della componente ambiente costituisce una approssimazione notevole poiché non tiene conto dell'effettiva interazione tra gli oggetti della scena.

Riflessione diffusa La riflessione diffusa è la componente della radiazione luminosa riflessa da una superficie perfettamente diffusiva ugualmente verso qualsiasi direzione. L'intensità della luce riflessa dipende dall'ampiezza dell'angolo θ tra la direzione d'incidenza della sorgente di luce \mathbf{l} , e

la normale \mathbf{n} alla superficie. Infatti, in accordo alla **legge di Lambert**, noi vediamo solo la componente verticale della luce incidente. Per comprendere questa legge osserviamo che quando la sorgente luminosa si abbassa, lo stesso ammontare di luce si distribuisce su un'area più vasta, e la superficie appare meno luminosa.



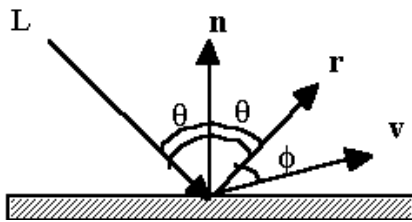
La riflessione diffusa si può dunque caratterizzare matematicamente, in accordo alla legge di Lambert, nel seguente modo $R_d = k_d \cos \theta$, dove $\cos \theta = \mathbf{l} \cdot \mathbf{n}$, e k_d è il **coefficiente di riflessione** che approssima il grado di diffusività della superficie, ossia la frazione di luce incidente che viene riflessa. Complessivamente, il termine relativo alla riflessione diffusa risulta uguale a $I_d = R_d L_d = k_d \cos \theta L_d = k_d (\mathbf{l} \cdot \mathbf{n}) L_d$.

Per tenere conto dell'attenuazione della luce che viaggia dalla sorgente luminosa alla superficie, si può inoltre introdurre un termine quadratico:

$$I_d = \frac{k_d}{a + bd + cd^2} (\mathbf{l} \cdot \mathbf{n}) L_d$$

dove d è la distanza tra la sorgente e la superficie e a , b e c sono costanti. Questa attenuazione della luminosità con la distanza consente di aumentare l'effetto di profondità.

Riflessione speculare Quasi tutti i materiali possiedono un grado più o meno alto di riflessione speculare. In particolare, più la superficie di un oggetto è liscia, più il suo comportamento è simile a quello di uno specchio. Modellare una superficie speculare in modo realistico è un compito piuttosto complesso, poiché la superficie in base a cui la luce viene diffusa dipende dalla lunghezza d'onda della luce incidente e cambia con l'angolo di riflessione. La riflessione speculare è modellata da Phong assumendo che, solo in prossimità della direzione \mathbf{r} di riflessione speculare, la radiazione luminosa venga riflessa su tutto lo spettro, a prescindere dalle caratteristiche di assorbimento proprie del materiale. L'ammontare di luce vista dall'osservatore dipende quindi dall'angolo ϕ tra \mathbf{r} e \mathbf{v} .



Nel modello di Phong si introduce un termine additivo che tiene conto della riflessione speculare dato dall'equazione $I_s = k_s L_s \cos^\alpha \phi$.

Il coefficiente k_s rappresenta la frazione di luce speculare riflessa. L'esponente α è chiamato **coefficiente di brillantezza**. All'aumentare di α la luce riflessa si concentra

in una regione sempre più stretta centrata sull'angolo di riflessione. Al limite, quando α tende all'infinito, il comportamento simulato è esattamente quello di uno specchio. Se i due vettori \mathbf{r} e \mathbf{v} sono normalizzati, possiamo utilizzare il loro prodotto scalare per definire il coseno dell'angolo ϕ , e il termine speculare diventa $I_s = k_s L_s (\mathbf{r} \cdot \mathbf{v})^\alpha$. Si osservi che, anche in questo caso, il modello è locale: si considera solo la riflessione della radiazione luminosa proveniente dalla sorgente, e non quella potenzialmente riflessa da altri oggetti nella scena.

Complessivamente, l'equazione d'illuminazione del modello di Phong risulta uguale a

$$I = \frac{I}{a + bd + cd^2} (k_d L_d \mathbf{l} \cdot \mathbf{n} + k_s L_s (\mathbf{r} \cdot \mathbf{v})^\alpha) + k_a L_a$$

dove si è aggiunto il termine di attenuazione della luminosità all'aumentare della distanza tra la sorgente e l'osservatore. Questa formula deve essere calcolata per tutte le sorgenti e per ciascun colore primario.

Ray Tracing

Il modello d'illuminazione locale usato fin qui ha delle limitazioni. Se un oggetto blocca la luce proveniente da una sorgente, gli oggetti al di là di esso restano in ombra. Se un oggetto è riflettente, la luce che si riflette da esso illumina altri oggetti. Un modello d'illuminazione locale non può produrre questi effetti né produrre ombre.

Il modello d'illuminazione di Phong, non permette di modellare:

- * la **riflessione diffusiva** tra gli oggetti nella scena (inter-riflessione)
- * la **riflessione speculare** di radiazioni tra gli oggetti
- * la gestione del **calcolo delle ombre**

Per risolvere i problemi di:

- trasparenza;
- riflessioni speculari da altri oggetti;
- ombre;

si applica il metodo **Ray Tracing**.

Il metodo **ray tracing** è basato sull'osservazione che, di tutti i raggi luminosi che lasciano una sorgente, i soli che contribuiscono all'immagine sono quelli che, dopo aver colpito l'oggetto, raggiungono l'osservatore. I raggi luminosi possono raggiungere l'osservatore sia direttamente, sia per effetto delle interazioni con le altre superfici. La maggior parte dei raggi che lasciano la sorgente non raggiungerà l'osservatore, e dunque non contribuirà all'immagine. Naturalmente, non è possibile seguire la traiettoria di ciascun raggio, tuttavia, se invertiamo la traiettoria dei raggi, e consideriamo solo quelli che partono dalla posizione dell'osservatore, possiamo determinare i raggi che contribuiscono all'immagine. Questa è esattamente l'idea alla base del metodo ray tracing, che simula **all'indietro** il cammino compiuto dalla radiazione luminosa per giungere all'osservatore. Poiché si deve assegnare un colore a ciascun pixel, si deve considerare almeno un raggio luminoso per ogni pixel, questo raggio è detto **raggio primario**. Ciascun raggio primario può intersecare una superficie oppure può andare all'infinito senza colpire nulla. Ai pixel che corrispondono all'ultimo caso verrà assegnato solo un colore di sfondo. I raggi che colpiscono le superfici che supponiamo opache richiedono il calcolo di una gradazione di colore per il punto d'intersezione. Nel metodo ray tracing, prima di applicare il modello di riflessione, si controlla se il punto d'intersezione tra il raggio primario e la superficie è illuminato. A questo scopo, si genera e si traccia un **raggio ombra**, diretto dal punto sulla superficie verso ogni sorgente luminosa, per calcolare se il punto d'intersezione è in ombra o no rispetto alla sorgente. Se un raggio ombra interseca una superficie prima di incontrare la sorgente, la luce è bloccata e non può raggiungere il punto considerato, che rimane dunque in ombra, almeno rispetto a questa sorgente. Nessun calcolo di luminosità è necessario per le sorgenti che sono bloccate da un punto sulla superficie. Se le superfici sono tutte opache, e se non si considerano gli effetti della luce diffusa da superficie a superficie, abbiamo un'immagine che ha, oltre all'illuminazione, anche delle ombre. Il prezzo che si deve pagare per l'introduzione delle ombre è quello di dovere effettuare un calcolo di superfici nascoste per ogni punto di intersezione tra un raggio primario e una superficie. Il metodo ray tracing consente inoltre di trattare in modo adeguato anche i casi in cui le superfici siano a trasparenza non nulla. Quando un raggio luminoso colpisce un punto di una superficie a trasparenza non nulla, la luce è parzialmente assorbita, e parte di questa luce contribuisce al termine di riflessione diffusa. Il resto della luce incidente si divide in un raggio trasmesso ed un raggio riflesso. Pertanto, quando una sorgente luminosa è visibile nel punto di intersezione tra una superficie a trasparenza non nulla ed un raggio primario, si eseguono le operazioni seguenti:

1. si calcola il contributo della sorgente luminosa nel punto di intersezione, usando un modello standard di riflessione;
2. si genera e si traccia un **raggio speculare** nella direzione di riflessione perfetta (\mathbf{r}), per calcolare il contributo colore portato dalla radiazione riflessa da altri oggetti sul punto di intersezione;

3. si genera e si traccia un **raggio trasparenza** nella direzione di trasmissione della luce (\mathbf{t}), per calcolare il contributo colore portato dalla radiazione trasmessa sul punto di intersezione.

Il raggio trasparenza e il raggio speculare sono trattati come se fossero raggi primari: sono intersecati, se possibile, con altre superfici, e terminano ad una sorgente luminosa tramite un raggio ombra, o all'infinito. Ad ogni superficie intersecata, essi possono inoltre generare raggi trasparenza e speculari addizionali, e così via. Per ogni raggio primario si costruisce allora un **albero di raggi**, che mostra quali raggi devono essere tracciati. L'albero è costruito dinamicamente dal processo di ray tracing.

Riassumendo, nel metodo ray tracing si eseguono le operazioni seguenti:

- **per ogni pixel**, si traccia ricorsivamente l'albero dei raggi, in base al seguente criterio di terminazione: si sceglie una profondità massima di ricorsione (numero di livelli dell'albero dei raggi) e non si tracciano i raggi il cui eventuale contributo sia inferiore ad un valore minimo dato;
- **per ogni raggio**, si interseca il raggio con tutti gli oggetti della scena e si sceglie l'intersezione più vicina all'osservatore;
- si applica il modello di illuminazione al **punto di intersezione**: se il raggio corrente è un raggio primario si ottiene il colore del pixel, se il raggio corrente è un raggio trasparenza o un raggio speculare, il colore si calcola come componente trasmessa o speculare.

Il metodo ray tracing può essere facilmente esteso al trattamento di primitive di output qualsiasi (ad esempio primitive solide, sfere, coni, blocchi, etc.) poiché basta dotare il metodo di un codice per il calcolo delle intersezioni tra i raggi e la nuova primitiva. Inoltre, esso gestisce correttamente la riflessione speculare tra gli oggetti poiché è basato su una modellazione globale della radiazione riflessa in modo speculare. La gestione della componente diffusa dell'illuminazione, realizzata usando un modello solo locale, non riesce invece a rappresentare i fenomeni di inter-riflessione della luce tra gli oggetti della scena. Il metodo è inoltre caratterizzato da un elevato costo computazionale, dovuto principalmente al calcolo delle intersezioni.

Rappresentazione degli oggetti

Per poter rappresentare un oggetto nella scena tramite il metodo Ray-tracing devo essere dunque in grado di determinare le intersezioni tra i raggi luminosi e l'oggetto stesso, nonché la direzione normale alla superficie dell'oggetto in ogni suo punto illuminato. L'idea alla base del metodo è quella di sostituire l'equazione matematica di ogni raggio in quella dell'oggetto e di determinare le soluzioni reali. Se queste esistono significa che il raggio colpisce l'oggetto e si va a considerare il punto d'intersezione più vicino al punto d'osservazione

$$t = \min\{t \geq 0 : P + tD \in G\}$$

dove P è la posizione dell'osservatore, D è la direzione del raggio e G è l'oggetto in questione.

Questo calcolo può essere svolto in maniera diretta quando si ha a che fare con superfici piane o quadriche. Molti ray-tracer utilizzano solo tali superfici per approssimare gli oggetti della scena.

Lo stesso calcolo può essere svolto in maniera approssimata quando si ha a che fare con superfici descrivibili in forma implicita. Sia $G = \{(x, y, z) : f(x, y, z) \leq 0\}$ l'equazione che descrive il mio oggetto; quindi si tratta di determinare la più piccola soluzione positiva t dell'equazione $f(P + tD) = 0$. Questa è un'equazione (eventualmente non lineare) in un'incognita e può essere risolta facilmente con metodi numerici (metodo di bisezione, di Newton..).

La normale alla superficie è facilmente determinabile dato che per il teorema del Dini essa è data dal ∇f calcolata nel punto di incidenza.

Quando però considero superfici espresse in forma parametrica, come nel caso della superficie del Dini, il discorso si complica. Sia $S = \{\varphi(u, v) : (u, v) \in D \subseteq \mathbb{R}^2\}$, con $\varphi : D \rightarrow \mathbb{R}^3$

$$\varphi(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}$$

La normale alla superficie è data dal semplice prodotto esterno

$$\frac{\partial \varphi}{\partial u} \wedge \frac{\partial \varphi}{\partial v}$$

mentre il t cercato è il minimo tale che $P + tD = \varphi(u, v)$; si tratta in questo caso di un sistema di tre equazioni in tre incognite, generalmente non lineare, che non è quindi facilmente risolvibile. Per la sua risoluzione il processo ray-tracing di POV-RAY non sfrutta il tipico metodo numerico di Newton ma una sorta di metodo di bisezione; vediamo in particolare. Innanzitutto viene implementato un algoritmo che permette, data una stima del massimo valore assunto dal modulo del gradiente di φ e dato un rettangolo R nel dominio di φ di determinare un parallelepipedo in \mathbb{R}^3 che contiene sicuramente $\varphi(R)$.

L'algoritmo vero e proprio consiste quindi nel suddividere il dominio iniziale D in rettangoli e per ciascuno di essi calcolare il corrispondente parallelepipedo. Se il parallelepipedo non interseca il raggio in questione il rettangolo corrispondente viene scartato, altrimenti, viene suddiviso in rettangoli più piccoli.

Quando si raggiunge la precisione desiderata, l'algoritmo termina.

La superficie del Dini

A questo punto abbiamo tutti gli strumenti necessari per comprendere come realizzare la nostra superficie matematica tramite un programma che utilizza la tecnica ray tracing.

Nel programma Pov_ray:

```
#include "colors.inc"
#include "textures.inc"
    background { color rgb<0,0.3,1>}
// Equazioni parametriche della superficie del Dini
parametric {
    function {cos(u)*sin(v)}
    function {sin(u)*sin(v)}
    function {cos(v)+ln(tan(v/2))+0.2*u}
    <0,0.001>,<4*pi,2> // Intervallo di variazione di u e v
    contained_by { box {<-3,-3,-3>,<3,3,3>} }
    max_gradient 3    accuracy 0.001
    precompute 18 x,y,z
    texture{ pigment {rgb<0,0.4,0.4>}}
        finish{
            ambient 0.4
            diffuse 0.6
            phong 0.5
            reflection 0.1}
        }
rotate <-90,0,0>
}

//Camera e sorgente luminosa
camera {
    location <0,3,-6>
    look_at <0,0,0> }

light_source {<10,10,-10>}
plane{y,-3
    pigment {checker color White color Grey}
}
}
```


Bibliografia

Links:

1. <http://it.wikipedia.org>
2. http://www.liberliber.it/biblioteca/p/pov_ray_team/manuale_pov_ray
3. <http://www.povray.org>
4. <http://www.ph-heidelberg.de/wp/filler/3D/anageo.html>
5. <http://translate.google.com> <http://www.uib.no>
6. <http://www.hmg.polimi.it/Demos/AJ0103/didattic/graphics/G3d>
7. <http://www.noemalab.org>
8. <http://karlotto.altervista.org>

Software:

POV-Ray