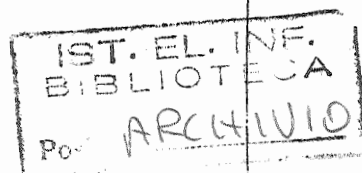


Consiglio Nazionale delle Ricerche



**ISTITUTO DI ELABORAZIONE
DELLA INFORMAZIONE**

PISA

RAPPRESENTAZIONE E VISUALIZZAZIONE DI
MODELLI VOLUMETRICI

C. Montani, R. Scopigno

Nota interna B4-54

Dicembre 1990

**PROGETTO FINALIZZATO
SISTEMI INFORMATICI E CALCOLO PARALLELO**

**SOTTOPROGETTO 1
Calcolo Scientifico per Grandi Sistemi**

Coordinatore Laura Moltedo

C. Montani *, R. Scopigno †

**RAPPRESENTAZIONE E
VISUALIZZAZIONE
DI MODELLI VOLUMETRICI**

N. 1/27

Settembre 1990

Rapporto Tecnico

* Istituto Elaborazione dell'Informazione
Consiglio Nazionale delle Ricerche
via S. Maria, 46 - 56100 Pisa ITALY - Tel. +39 50 500159

† Istituto CNUCE - Consiglio Nazionale delle Ricerche
via S. Maria, 36 - 56100 Pisa ITALY - Tel. +39 50 593211

Sommario

Obiettivo di questo lavoro e' fornire una panoramica il piu' possibile completa delle numerose tecniche sviluppate per la rappresentazione e la visualizzazione di dati volumetrici. La ricerca di nuove tecniche per il trattamento di dati volumetrici, caratterizzati dall'essere rappresentazione della occupazione spaziale di un fenomeno piuttosto che di descrizione delle sue superfici di contenimento, ha recentemente generato sviluppi particolarmente significativi in termini sia di definizione di nuove strategie algoritmiche che di efficienza globale delle tecniche proposte. In questo lavoro si cerca quindi di fornire, all'interno di un quadro di riferimento il piu' possibile unificante, descrizione ed analisi comparativa delle varie proposte presenti in letteratura. E' inoltre fornita una concisa descrizione delle caratteristiche di alcuni pacchetti software di visualizzazione volumetrica disponibili sul mercato.

Parole chiave

rappresentazione e rendering volumetrico, voxel, schemi di rappresentazione per compressione dei dati, ricostruzione di superfici, algoritmi di proiezione dei voxel, ray tracing, voxel shading.

Abstract

An overview of the great number of techniques recently proposed to represent and render volumetric data is reported in this paper. The great interest towards the management of volumetric data, i.e. data giving a direct representation of the spatial occupancy of the objects in a 3D scene instead of representing their boundaries, originates a large number of new visualization algorithms characterized by increasing efficiency and quality of results. An analysis and a comparison of these new solution is discussed in this paper. The characteristics of some commercial software packages are also briefly described.

Keywords

volume representation and rendering, voxel, compression representation schemes, surface reconstruction, voxel projection, ray tracing, voxel shading.

Indice

1. Introduzione
2. Spazio di definizione dei dati
 - 2.1 Rappresentazioni voxel-based
 - 2.2 Rappresentazioni cell-based
3. Strutture di rappresentazione interna
 - 3.1 Coerenza spaziale monodimensionale
 - 3.2 Coerenza spaziale bidimensionale
 - 3.3 Coerenza spaziale tridimensionale
 - 3.4 Rappresentazioni miste e valutazioni
4. Visualizzazione: approccio geometrico
5. Visualizzazione: approccio volumetrico (o volume rendering)
 - 5.1 Slice by Slice
 - 5.2 Tecniche proiettive
 - 5.3 Ray Tracing
 - 5.4 Tecniche di shading
6. Software commerciale per visualizzazione di modelli volumetrici
7. Bibliografia

1. Introduzione

Nel corso degli ultimi anni l'attenzione sia degli specialisti che del pubblico e' stata catturata sempre piu' frequentemente da immagini di modelli anatomici ottenute come viste tridimensionali di dati provenienti da apparecchiature per tomografia computerizzata o risonanza magnetica. Questa classe di risultati, dovuta a nuove tecniche di computer graphics dette di *Visualizzazione Volumetrica*, ha conosciuto nell'ultimo lustro una considerevole evoluzione, generando immagini sempre piu' sofisticate la cui diffusione tramite riviste di massa puo' aver generato l'impressione che tali tecniche siano divenute gia' di routine per lo meno in campo medico. Al presente cio' purtroppo non corrisponde al vero. Le motivazioni sono molteplici: dalla strutturale difficolta' dell'inserimento di tecnologie informatiche avanzate nei campi applicativi, alla particolare evoluzione di tali tecniche guidata, in un primo tempo, piu' dalla ricerca di quella che in campo grafico viene detta "real quality" (alta corrispondenza tra fenomeno osservato ed immagine di sintesi) piuttosto che da reali esigenze dell'utente finale, agli obiettivi e strategie direttamente definiti da esperti dei campi applicativi coinvolti. I campi applicativi, d'altra parte, non sono chiaramente limitati alle varie discipline mediche (radiologia, ortopedia, chirurgia, etc.) ma coinvolgono svariate discipline accomunate dalla necessita' di visualizzare dati volumetrici ottenuti sia con l'uso di scanner 3D (ad es. in biologia molecolare od in applicazioni di indagine non distruttiva [Jen89b]) che come risultato di simulazioni o, piu' genericamente, di computazioni (chimica molecolare, fisica, etc.). Le iniziali difficolta' sono state infine superate [Rho90] con la costituzione, soprattutto nell'ambito del medical imaging, di gruppi di ricerca interdisciplinari impegnati sia sul fronte del miglioramento della qualita' dell'immagine ma anche e soprattutto sull'accuratezza (e' essenziale che l'immagine sia il piu' possibile fedele o che sia possibile comprenderne le eventuali limitazioni), l'utilita' effettiva dei risultati e la possibilita' di reale introduzione delle tecniche sviluppate anche in termini di analisi costi-benefici.

Come spesso accade per le discipline scientifiche di recente sviluppo ed ancora prive di un inquadramento metodologico, anche la Visualizzazione Volumetrica e' andata assumendo significati e definizioni molto diverse rispetto all'approccio applicativo prescelto. Alla definizione classica [Smit87] che presenta la visualizzazione volumetrica come l'insieme delle tecniche che consentono di rappresentare sul piano immagine un volume di dati, piuttosto che le superfici o gli spigoli degli oggetti rappresentati, si contrappongono in realta' innumerevoli applicazioni in cui il termine di visualizzazione volumetrica e' in qualche modo riferito allo schema di rappresentazione interna degli oggetti 3D (tipicamente l'enumerazione esaustiva del contenuto delle celle elementari di un grigliato regolare 3D) piuttosto che alle tecniche di visualizzazione del volume stesso.

Chiariamo ulteriormente questo concetto. Per modelli in cui le superfici che delimitano gli oggetti siano descrivibili matematicamente, per scomposizione in liste di poligoni connessi o per composizione delle superfici di solidi elementari, le tecniche di visualizzazione basate su superfici si rivelano perfettamente adeguate. Esistono invece modelli, tipicamente quelli rappresentanti dati ottenuti per scansione diretta dello spazio 3D reale con apparecchiature per tomografia computerizzata, risonanza magnetica, tomografie a emissione di positroni, etc., in cui i confini tra regioni non sono descrivibili da semplici superfici bidimensionali quanto piuttosto da zone tridimensionali irregolari di transizione. In questi modelli di dati l'individuazione di una superficie da visualizzare, mediante tecniche di soglia o

altre approssimazioni non lineari, introdurrebbe una forte inaccuratezza nei dati.

Sempre secondo la definizione classica le tecniche di visualizzazione volumetrica, o volume rendering, rappresentano ogni punto del volume 3D evitando la ricostruzione approssimata di superfici o di spigoli. Ad ogni elemento del volume da visualizzare (voxel) e' associato infatti informazione su colore/densita'/coefficiente di trasparenza. Il colore di ogni pixel del piano immagine e' quindi formato tenendo conto dei contributi di colore e trasparenza di tutti i voxel che giacciono lungo la linea di vista passante per quel pixel.

Come vedremo in seguito, nonostante numerose esperienze dirette all'approccio "volume rendering" siano gia' state acquisite [Dreb88, West89, Upso88, Sabe88], risulterebbe senz'altro riduttivo limitarsi, in questa rassegna, solo a queste ultime. Infatti svariati approcci sono stati sviluppati per la visualizzazione di dati volumetrici, essenzialmente per la soluzione di problemi legati ai costi delle memorie, alla complessita' nella visualizzazione, alle manipolazioni che si intendono effettuare sui dati, alla forma in cui i dati si rendono disponibili ed alla necessita' di operare conversioni tra diverse rappresentazioni. Lo scopo di questo lavoro e' di presentare con una qualche organicita' le diverse tecniche e soluzioni individuate anche ove esse conducano a risultati intermedi o, a volte, alternativi, rispetto alla visualizzazione volumetrica classicamente intesa.

Anche se non e' agevole fissare a priori una rigida classificazione delle tecniche di volume rendering, e' comunque possibile discriminare tra le varie soluzioni valutando alcuni dei parametri fondamentali nel processo di visualizzazione. In particolare ci occuperemo di:

- (a) spazio di definizione del modello dei dati (rappresentazioni voxel-based e rappresentazioni cell-based);
- (b) strutture di rappresentazione interna del modello (essenzialmente il diverso grado di coerenza spaziale che si cerca di individuare per ridurre la quantita' di memoria necessaria alla rappresentazione senza sacrificare l'operativita' degli algoritmi);
- (c) algoritmi di rendering per la generazione di viste in spazio immagine 2D da modelli volumetrici (ricostruzione di superfici di interesse e loro proiezione con rimozione delle parti nascoste, approccio object-space con tecniche proiettive, approccio image-space con tecnica ray casting, modelli di shading).

Nell'ultima parte del lavoro e' inoltre riportata una breve descrizione di alcuni sistemi commerciali di visualizzazione di dati volumetrici.

La stesura di questo lavoro ben si inquadra nelle attivita' della Unita' Operativa del Progetto Finalizzato "Sistemi Informatici e Calcolo Parallelo", Sottoprogetto "Calcolo Scientifico per Grandi Sistemi", coordinata dalla dott.ssa Laura Moltedo, la cui attivita' si rivolge particolarmente alle problematiche associate all'uso di tecnologie informatiche, e piu' specificamente grafiche, per la comunicazione e l'analisi dei risultati del calcolo e della ricerca scientifica. Questo contributo vuole essere un approfondimento di tematiche gia' parzialmente trattate in un piu' generale ed ampio lavoro pubblicato nella medesima collana [Asc90].

Infine si sottolinea come il campo applicativo preso in esame sia caratterizzato da notevole interesse e velocita' evolutiva, e da cio' deriva anche la non completa stabilizzazione del un corpus terminologico associato. Questo lavoro vuole quindi essere un prima trattazione di review e richiedera' nel prossimo futuro i necessari aggiornamenti onde evitarne la rapida obsolescenza.

2. Spazio di definizione dei dati

2.1 Rappresentazioni voxel-based

Quando il dominio di definizione del modello volumetrico e' costituito da una serie di campionamenti nello spazio 3D, senza alcuna forma di interpolazione tra i nodi del grigliato, si parla di rappresentazioni a *voxel* (da volume-element). I dati raccolti mediante scanner 3D, quali le varie apparecchiature di tomografia, ricadono in questa categoria.

Ciascun campionamento nel dominio 3D rappresenta una intensita' media su un piccolo elemento di volume. La variazione esistente tra due campionamenti adiacenti e' persa nel processo di discretizzazione della scena reale oppure va oltre la risoluzione del dispositivo di acquisizione dei dati.

In una rappresentazione a voxel ogni elemento di volume e' in genere caratterizzato da un solo valore (intensita'). L'intensita' e' considerata costante su tutto il voxel cosi' come ogni altro possibile attributo associato al voxel stesso come, ad esempio, la trasparenza od il colore. Un esempio classico di rendering volumetrico con approccio voxel-based e' descritto nel lavoro di Drebin [Dreb88].

2.2 Rappresentazioni cell-based

Si fa riferimento a rappresentazioni *cell-based* (o *grid-based*) quando i dati da visualizzare sono definiti sui nodi del grigliato di riferimento piuttosto che nei centroidi dei voxel; su tali dati e' possibile calcolare il valore del campo scalare presente in ciascun punto dello spazio come funzione o interpolazione dei valori sul grigliato. La funzione di interpolazione puo' variare da algoritmo ad algoritmo, nelle versioni piu' comuni e' lineare lungo ciascuna dimensione del grigliato oppure trilineare su ogni cella. In questa rappresentazione il campo scalare risulta continuo e derivabile all'interno delle celle anche se non derivabile sulle facce di ogni cella. Altre forme di interpolazione piu' sofisticate (tri-cubica) assicurano continuita' e derivabilita' ovunque ma risultano molto piu' onerose dal punto di vista del calcolo. Le tecniche di visualizzazione che fanno uso di un approccio cell-based ricadono in due categorie: image-space oriented [Kaje84] [Sabe88] [Levo88] ed object-space oriented [Upso88].

3. Strutture di rappresentazione interna

Indipendentemente dallo spazio di definizione del modello (voxel- o cell-based) e dalle caratteristiche strutturali e geometriche dei dati raccolti (generalmente i dispositivi di acquisizione restituiscono i dati per piani successivi detti slice in cui le distanze tra i campionamenti adiacenti lungo i tre assi non sono identiche) la piu' immediata rappresentazione interna cui i dati possono essere ricondotti e' la classica matrice tridimensionale. Le dimensioni di tali matrici, $O(n^3)$ con n risoluzione dello spazio volumetrico, sono tali da non consentire la manipolazione diretta dei dati se non si opera una qualche forma di compressione. Nonostante questa affermazione vada perdendo significato con il decrescere dei costi delle memorie e la crescente potenza delle stazioni di lavoro [Jen89b], e' ancora senz'altro vero che solo soluzioni integrate hw/sw ad hoc consentono la gestione e manipolazione di grandi quantita' di dati nella loro forma classica [Jack85] [Kauf86] [Kauf88].

Gli schemi di rappresentazione proposti in letteratura sono essenzialmente basati sulla compressione dei dati mediante la ricerca di una qualche forma di coerenza spaziale: coerenza mono-, bi- oppure tri-dimensionale. Fatta eccezione per i volumi 3D ottenuti per discretizzazione analitica di scene artificiali, difficilmente e' possibile individuare forme di coerenza spaziale sui dati grezzi restituiti dai dispositivi di acquisizione a meno di provvedere, prima della compattazione, alla classificazione (o segmentazione) dei dati. Il processo di classificazione puo' condurre ad una rappresentazione binaria (se si e' interessati ad isolare una sola classe di informazioni) oppure ad una rappresentazione ad n valori (in cui n e' in genere uno o due ordini di grandezza inferiore al numero di intensita' dei dati originali). La tecnica di classificazione utilizzata e' dipendente dalla tecnica di visualizzazione da adottare, dal tipo di risultato atteso e, soprattutto, dalle caratteristiche del dispositivo di acquisizione e dalla natura dei dati. E' proprio per questa ultima dipendenza che non ci addenteremo nel tema della classificazione in quanto fortemente correlata alla specifica applicazione.

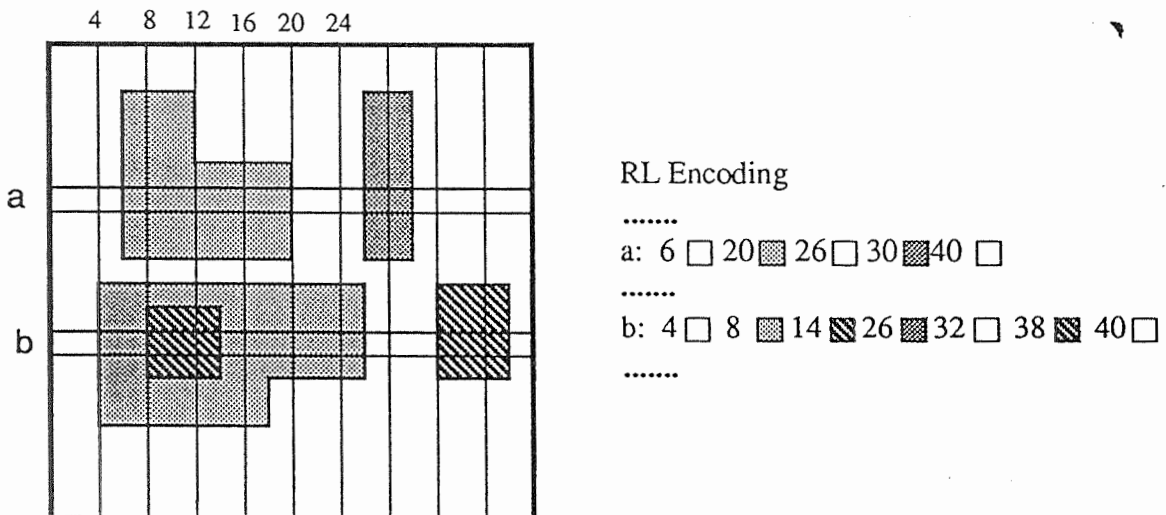


Fig.3.1.1: Rappresentazione di alcune righe di una slice con schema run length.

3.1 Coerenza spaziale monodimensionale

Alla prima classe appartengono gli schemi indicati generalmente con il termine run length (Fig.3.1.1), gli schemi cioè che riorganizzano le informazioni accumulando in unica entità i dati adiacenti (lungo una dimensione del volume 3D) aventi identici valori di intensità [Free74]. Numerose sono le variazioni sul tema: poiché, generalmente, i dati da trattare sono organizzati per slice successive, gli schemi run length si preoccupano di accorpate gli elementi adiacenti e identici di ogni riga di ciascuna slice. I vari metodi proposti in letteratura si differenziano essenzialmente nell'uso di particolari accorgimenti, quali l'omissione del valore considerato di fondo nell'immagine [Brig86], l'impiego di una struttura supplementare per l'accesso diretto alla generica riga di ogni slice [Triv85], l'uso di una struttura dati a puntatori [Mont90] in cui ogni entrata (x,y) di un array bidimensionale individua una lista di "run" nella direzione z (Fig.3.1.2); questo ultimo approccio ben si presta per la classe di algoritmi di visualizzazione ray casting [Mont90].

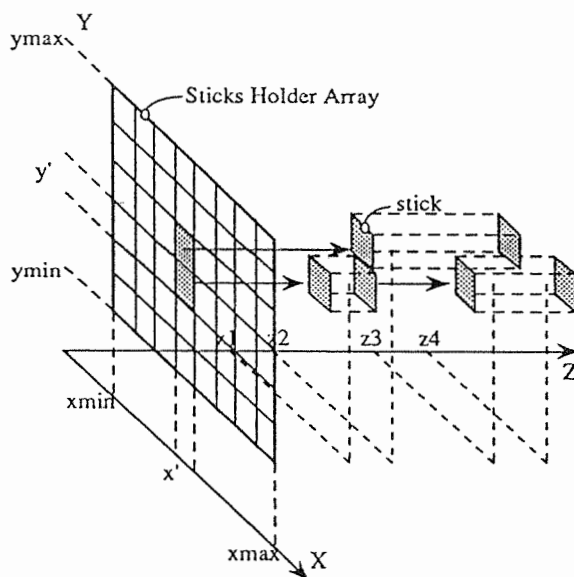


Fig.3.1.2: Rappresentazione STICKS di un volume binario.

3.2 Coerenza spaziale bidimensionale

Gli schemi di rappresentazione per spazi volumetrici che si prefiggono la ricerca di coerenza bidimensionale sono mutuati direttamente da tecniche ampiamente utilizzate nell'elaborazione di immagini 2D. Tra quelle più diffuse (quadtree, medial axis transformations, skeleton, etc.), il quadtree è certamente la struttura più utilizzata nella visualizzazione volumetrica [Same84]. Ogni slice delle informazioni da elaborare è considerata una immagine a se' stante ed eventualmente estesa (mediante valori fittizi) nelle due direzioni fino ad ottenere dimensioni che rappresentino potenze di due.

Se l'immagine non e' a valori uniformi viene suddivisa in quattro quadranti identici dando luogo a quattro nodi foglia del nodo radice che rappresenta l'intera immagine (Fig.3.2). Il processo di suddivisione procede ricorsivamente sui quadranti e si interrompe in caso di uniformita' delle informazioni o per raggiungimento della risoluzione di base delle immagini (quando, cioe', il quadrante in esame si e' ridotto alle dimensioni del pixel). Lo schema quadtree puo' essere impiegato sia nel caso di immagini binarie che ad n valori. In questo secondo caso, comunque, in letteratura e' comunemente proposto l'impiego di n quadtree binari anziche' un solo quadtree a n valori.

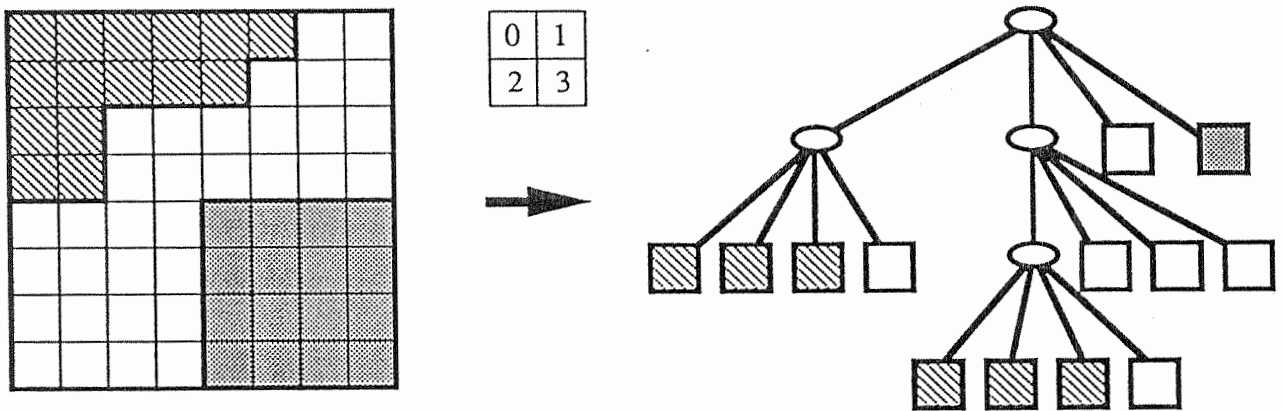


Fig.3.2: Rappresentazione a quadtree di una immagine ad n valori)

3.3 Coerenza spaziale tridimensionale

La coerenza spaziale 3D, infine, e' sfruttata dallo schema di rappresentazione octree [Meag82]. L'octree rappresenta la naturale estensione del quadtree a volumi di dati piuttosto che ad immagini. Il volume di rappresentazione (esteso eventualmente a dimensioni potenze di due) viene suddiviso in 8 ottanti. Ciascun ottante e' ricorsivamente suddiviso in 8 nuovi ottanti fino a che non si pervenga a nodi completamente uniformi o il generico ottante non coincida con un voxel.

3.4 Rappresentazioni miste e valutazioni

E' stato frequentemente osservato [Brig86] [Chen89] come, sia da un punto di vista occupazionale che algoritmico, non sia possibile definire a priori lo schema di rappresentazione migliore in assoluto, essendo il grado di compressione raggiunto da ognuno di questi fortemente dipendente dai dati e considerando che i vantaggi che si ottengono, in termini di compressione, aumentando la dimensionalita' della coerenza ricercata sono in genere bilanciati dal maggiore costo degli algoritmi di accesso alla struttura di rappresentazione stessa (si pensi ad esempio alla visita diretta completa di

una struttura di rappresentazione o la ricerca della intersezione raggio-modello volumetrico).

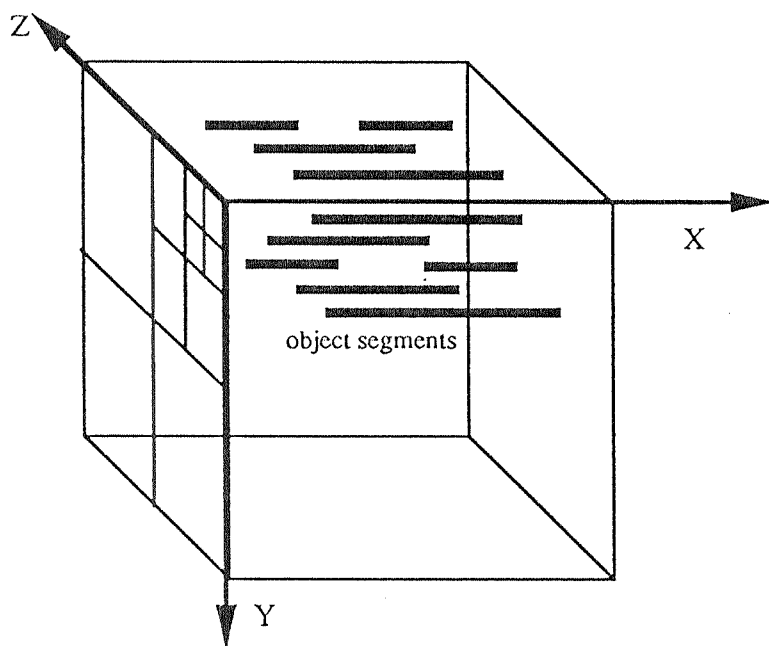


Fig.3.4: Rappresentazione volumetrica mista: run length sulle singole slice (direzione X), quadtree di gestione tra le diverse slice (piano YZ).

Per completezza di informazione e' necessario sottolineare che non esistono soltanto gli approcci alla rappresentazione interna appena descritti ma anche approcci misti. Nella soluzione di Chen e Sontag [Chen89], ad esempio, ogni slice e' codificata nello schema run length (precisamente nello schema suggerito da Trivedi [Triv85]), l'insieme delle slice e' invece mantenuto con una struttura a quadtree in cui il generico nodo foglia e' una matrice bidimensionale di puntatori ai "run" delle corrispondenti righe sulle slice (Fig.3.4). A conferma di quanto accennato riguardo i vantaggi o svantaggi dei diversi schemi di rappresentazione, gli autori sostengono che la scelta di un sistema misto (chiamato quadtree-segment) e' dovuta al costo di costruzione del quadtree, alla scarsa compressione che uno schema a quadtree raggiunge quando si operi con dati ottenuti da scanner e, conseguentemente, al costo di visita di un albero che presenta un altissimo numero di foglie al livello piu' basso dell'albero.

4. Visualizzazione: approccio geometrico

Abbiamo accennato nell'introduzione come il processo di visualizzazione di dati volumetrici richieda la proiezione della scena 3D nello spazio immagine 2D. L'approccio forse più consolidato a questo problema e che prende le mosse dalle tecniche classiche della computer graphics tridimensionale e' quello di proiettare sul piano immagine le superfici degli oggetti che compongono la scena, eliminare le superfici, o parte di esse, logicamente nascoste all'osservatore ed eventualmente applicare tecniche di shading alle superfici risultanti.

Quando si operi in ambienti di visualizzazione volumetrica tale approccio non e' direttamente applicabile in quanto le superfici di delimitazione degli oggetti non sono date ne' in modo analitico ne', tantomeno, sintetico. Il primo passo, quindi, per una visualizzazione mediante eliminazione delle parti nascoste e shading di superfici, consiste nella ricostruzione delle superfici stesse. Le superfici da ricostruire, che molto spesso non hanno una corrispondenza fisica esatta con il modello reale di cui il volume dei dati rappresenta la discretizzazione, potranno, di volta in volta, essere associate a superfici di isointensita' giudicate di interesse per l'utente. E' opportuno sottolineare innanzitutto che, per loro natura, le tecniche di ricostruzione di superfici ben si adattano a modelli in rappresentazione cell-based; la possibilita', infatti, di interpolare linearmente tra nodi adiacenti del grigliato consente grande flessibilita' nella individuazione di superfici.

Le prime soluzioni proposte [Kepp75] [Chri78] consistevano nella generazione di curve di livello di isointensita' per ognuna delle slice in esame e nella costruzione di isosuperfici poligonali mediante la correlazione dei punti corrispondenti tra due curve appartenenti a slice contigue (Fig.4.1). Questo metodo lavora molto bene fino a quando una curva su un piano non si divide in due o più curve sul piano adiacente. In questi casi e' molto difficile definire una tecnica non ambigua di connessione tra le curve.

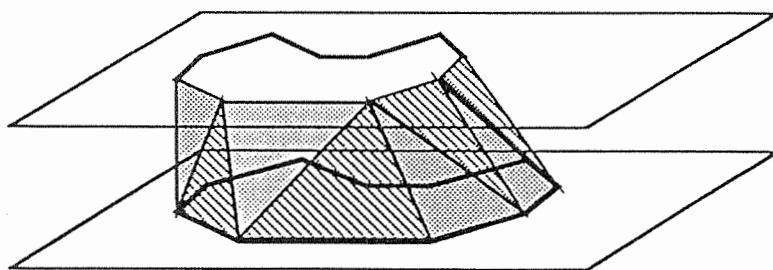


Fig.4.1: Costruzione di superfici per correlazione di curve di livello contigue.

Recentemente sono state proposte diverse tecniche che generano superfici poligonali direttamente a partire dai dati [Lore87] [Upso88]. Per meglio chiarire il tipo di approccio descriviamo in qualche dettaglio l'algoritmo "Marching Cubes" proposto da Lorensen e Cline [Lore87]. La struttura di rappresentazione e' visitata cella per cella; per ogni cella si determina se essa e' attraversata dalla superficie che si sta ricostruendo, cioe' se il campo scalare misurato sugli otto vertici della cella contiene il valore di soglia che individua la superficie; in caso affermativo e' necessario individuare il numero ed il tipo di poligoni che passano nella cella (Fig.:4.2);

si costruisce pertanto un indice che rappresenti, in binario, i vertici con valori di intensita' superiori (bit a 1) o inferiori (bit a 0) al valore di soglia prefissato (8 vertici generano 256 combinazioni possibili); tale indice indirizza una tabella che fornisce, per ogni possibile combinazione, il numero e la posizione dei triangoli che formano la superficie interna alla cella mentre i valori di intensita' ai nodi forniscono (per interpolazione) la posizione sugli spigoli della cella dei vertici dei triangoli individuati; i triangoli cosi' individuati possono essere proiettati sul piano immagine ed utilizzati come input del classico pipeline di eliminazione delle parti nascoste e shading di superfici.

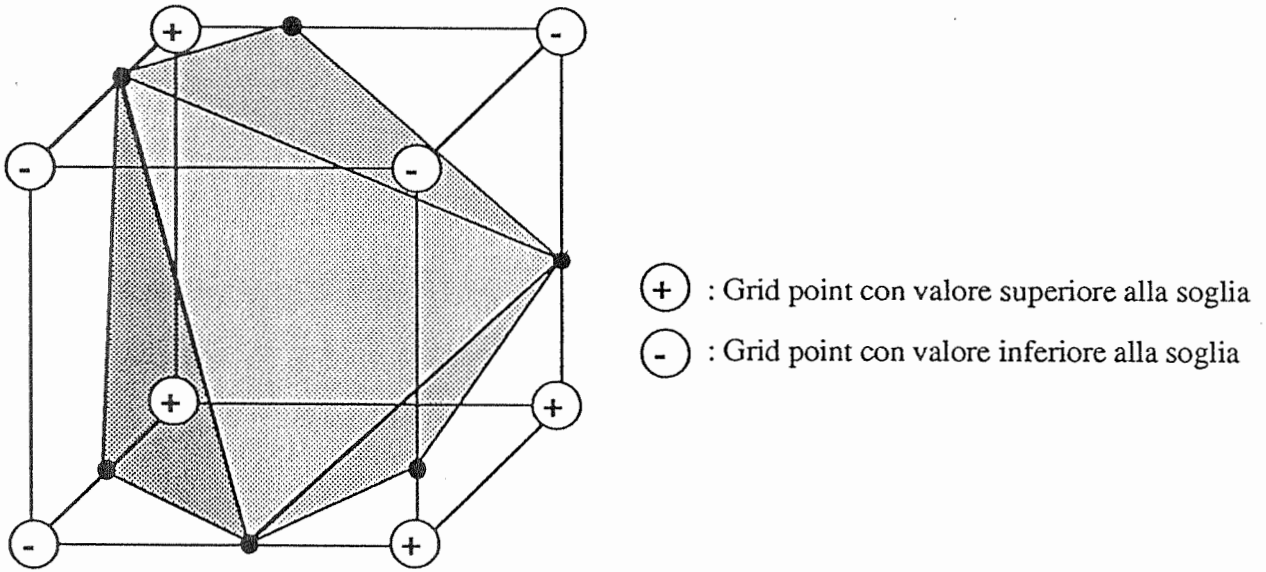


Fig.4.2: Algoritmo Marching Cubes: ricostruzione della parte di superficie interna ad una cella per triangolazione guidata dal valore dei vertici.

5. Visualizzazione: approccio volumetrico

Sono descritte in questa sezione le tecniche che permettono la visualizzazione diretta dei dati volumetrici senza richiederne la conversione in strutture di rappresentazione alternative intermedie (attraverso la costruzione di curve o superfici di livello). Caratteristica discriminante tra le varie tecniche di visualizzazione sarà il diverso grado di qualità, di flessibilità e costo computazionale necessario alla generazione di immagini bidimensionali che, nel rappresentare il contenuto del volume tridimensionale di dati, rendano evidente l'effetto di profondità e di tridimensionalità implicitamente codificato nel volume di dati. Facendo riferimento alla classificazione dei dati volumetrici introdotta nella prima sezione, le seguenti tecniche sono tutte particolarmente orientate a dati *voxel-based*, mentre solo la tecnica Ray Tracing è applicabile a dati *cell-based*.

Le tecniche di visualizzazione sono classificate, nel contesto dei successivi paragrafi, rispetto alla metodologia scelta per la visita del volume dei dati. Prescindiamo per ora dalla particolare tecnica di visita e visualizzazione adottata. Di un volume di dati, in cui sono generalmente rappresentati più oggetti sovrapposti o strati differenti dello stesso, potranno essere restituite sia immagini in cui sia visualizzato per trasparenza il contenuto del volume, che immagini in cui compaiano le superfici di transizione (eventualmente con simulazione di effetti di trasparenza di superfici) tra i diversi strati od oggetti rappresentati. Nel primo caso sono ottenute immagini semi trasparenti in cui il valore di ogni pixel è dipendente dall'opacità dei voxel che nella particolare vista vi sono proiettati. Nel secondo caso le superfici sono identificate usando un insieme di valori soglia ottenuto con una precedente classificazione del contenuto del volume dei dati. È chiaro che dati relativi ad oggetti opachi sono generalmente rappresentati tramite la visualizzazione delle loro superfici; si tende invece ad usare la visualizzazione per trasparenza ove si abbia interesse a guardare all'interno di un oggetto che non abbia contorni distinguibili od ove non sia possibile determinare automaticamente le superfici che racchiudono l'oggetto. Nel caso di visualizzazione per trasparenza le immagini ottenute ricordano molto quelle producibili con le comuni tecniche radiografiche (raggi X).

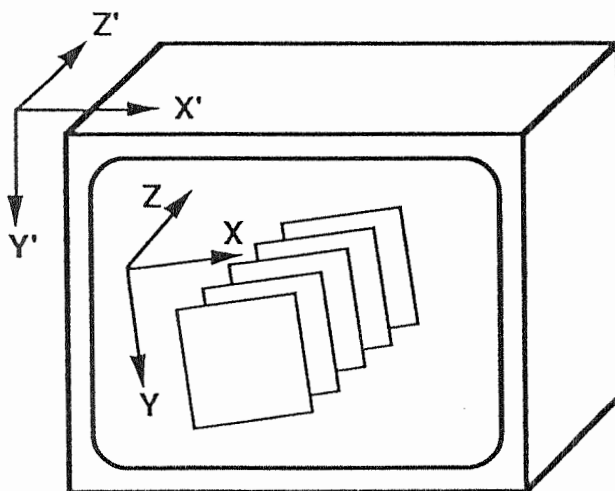


Fig.5.1: Object space (X,Y,Z) ed image space (X',Y',Z').

Va inoltre accennato come per tutti i metodi di visualizzazione presentati, con l'unica eccezione dei metodi slice by slice, sia necessario operare trasformazioni dei dati dallo spazio di definizione (N^3 object space) allo spazio dell'immagine (image o display space). Tale trasformazione e' regolata dalle ben note tecniche di trasformazione geometriche della geometria proiettiva [Fole82].

5.1 Slice by Slice

Tecnica di visualizzazione estremamente semplice, in termini di complessita' computazionale, in cui il contenuto del volume di dati (che si suppone memorizzato in n array bidimensionali) viene visualizzato per sezioni. La generazione delle sezioni puo' essere banalmente realizzata visualizzando direttamente le N sezioni di cui il volume e' costituito; in tal caso e' richiesto il solo trasferimento dei dati da memoria secondaria a memoria video con tempi di visualizzazione dipendenti dalle caratteristiche hw della macchina (da pochi secondi su personal computer a frazioni di secondo su workstation). D'altro canto, poter visualizzare solo sezioni predefinite dello spazio volumetrico costituisce sicuramente una modalita' di indagine insufficiente all'analisi dei dati. Per una piu' flessibile analisi dei dati, un pacchetto software che gestisca la visualizzazione *slice by slice* deve necessariamente permettere la semplice definizione interattiva di sezioni, sia ortogonali ad uno qualsiasi degli assi dello spazio n^3 che ad orientamento qualsiasi. In questo caso, la generazione delle sezioni richiede una scansione dei dati la cui complessita' e' dipendente dal modello di rappresentazione utilizzato, mantenendo comunque complessita' modesta se paragonata alle tecniche che saranno descritte successivamente.

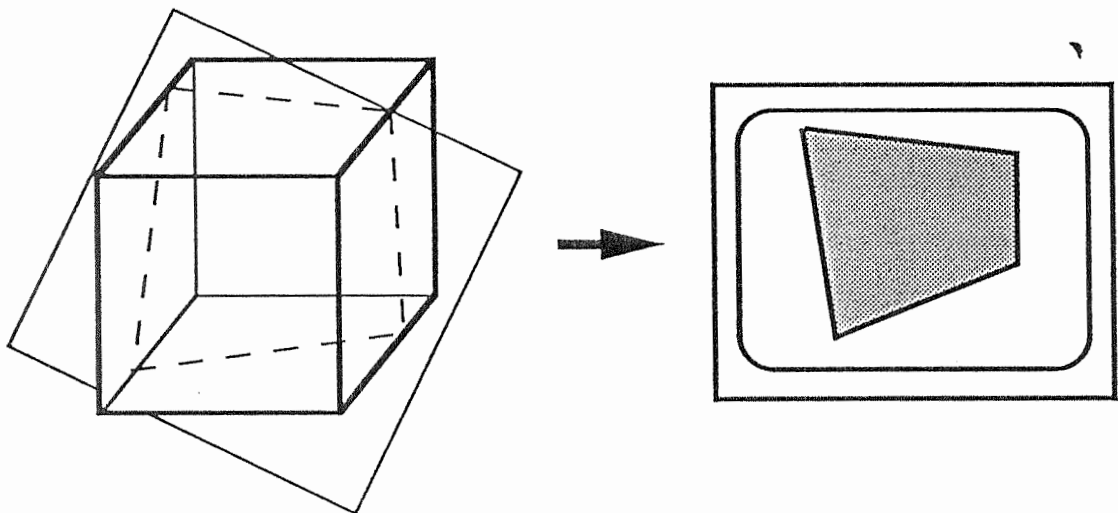


Fig 5.2: Visualizzazione *Slice by slice* : estrazione di slice generalizzata

Ovvio limite delle tecniche *slice by slice* e' il non permettere l'estrazione dell'informazione tridimensionale contenuta nei dati; l'analisi dei dati non subisce infatti alcuna evoluzione rispetto alla classica visione in controluce delle lastre, correntemente operata in campo medico, lasciando completamente a carico dell'utente la critica fase di ricostruzione dell'informazione tridimensionale implicita nei dati. Si perde in questo modo gran parte delle potenzialita' che lo strumento informatico offre per facilitare l'analisi dei dati ed una loro piu' semplice e diretta intellegibilita' [Fren89].

5.2 Tecniche proiettive

Sia la tecnica di visualizzazione *back-to-front* (BTF) che la *front-to-back* (FTB) possono essere caratterizzate come tecniche proiettive, in quanto ogni immagine del volume di dati e' ottenuta visitando opportunamente l'intero volume di dati e proiettando i voxel significativi sul piano di vista.

Back-To-Front

L'algoritmo BTF, proposto da Frieder, Gordon e Reynolds [Frie85], si basa su un attraversamento dell'array 3D ottenuto facendo variare gli indici di slice, riga e colonna in ordine di distanza decrescente rispetto all'osservatore. La lista di slice e' quindi attraversata partendo da quella piu' distante dall'osservatore per finire con quella piu' vicina. A sua volta ogni slice e' attraversata, proiettando i rispettivi voxel su video, facendo variare sia l'indice di riga che quello di colonna in ordine crescente o decrescente in funzione della posizione dell'osservatore rispetto agli assi di riferimento.

Con il metodo BTF possono essere generate viste del volume di dati ottenute come proiezioni parallele con posizione dell'osservatore qualsiasi. La possibilita' di generare solo proiezioni parallele non va intesa come un limite: tali proiezioni infatti risultano del tutto adeguate alla maggior parte delle applicazioni interessate alla visualizzazione volumetrica. In esse e' garantito, tra l'altro, il mantenimento di un corretto rapporto di visualizzazione degli oggetti nella scena, privo ad esempio delle deformazioni introdotte dalle tecniche proiettive prospettiche; cio' permette di estrarre informazioni geometriche (distanze, misura di superfici, etc) direttamente dalle immagini. Inoltre, se e' vero che le proiezioni prospettiche risultano un valido strumento per l'estrazione e la rappresentazione di informazione tridimensionale per scene in cui siano presenti andamenti rettilinei delle superfici, esse in realta' non aggiungono molta informazione rispetto alle proiezioni parallele quando applicate a scene contenenti corpi di ridotte dimensioni e/o caratterizzate da superfici molto irregolari.

In fig.5.3 e' riportato un esempio 2D ove una slice binaria e' proiettata sullo schermo; l'ordine in cui i voxel della particolare slice sono visitati, essendo l'origine degli assi il punto piu' lontano dall'osservatore, e' per indice di colonna (asse X) crescente e di riga (asse Y) decrescente. Ogni voxel attraversato a valore 1 e' quindi proiettato sullo schermo. La scelta dell'indice a crescita piu' veloce (l'indice che varia nel loop piu' esterno) e' ininfluente rispetto all'immagine risultante, e tale scelta puo' essere quindi arbitraria. Ove il volume di dati sia rappresentato da una lista di slice, la precedente assunzione permette, qualsiasi sia la vista scelta dall'utente, di scegliere come loop piu' esterno la variazione dell'indice di slice e di attraversare cosi' i voxel leggendo una slice alla volta. Cio' consente

all'algoritmo di mantenere una sola slice in memoria centrale in ogni istante, con ovvi vantaggi in termine di occupazione di memoria. Su ogni slice i voxel sono quindi visitati scegliendo opportunamente l'ordine di visita sui due assi della slice. Va notato che l'ordine di visita dei voxel non costituisce un ordinamento basato sulla distanza dall'osservatore [Dist((x,y,z), viewer)], ma un ordinamento parziale guidato dalla seguente regola:

$$[\text{Project}(x,y,z)=\text{Project}(x',y',z')] \quad \text{AND} \\ [\text{Dist}((x,y,z),\text{viewer})<\text{Dist}((x',y',z'),\text{viewer})]$$

==>

$$(x,y,z)>(x',y',z')$$

Con approccio BTF si possono generare sia immagini che raffigurano il guscio piu' esterno contenuto nel volume :

(1) LOOP

```
(x,y,z)=Next_B-T-F(VoxVol);
IMAGE(Project(x,y,z))=Val(x,y,z);
```

sia immagini che visualizzano solo la/le superficie/i associata/e a predeterminati valori soglia (treshold rendering):

(2) LOOP

```
(x,y,z)=Next_B-T-F(VoxVol);
if (Val(x,y,z)=soglia) then IMAGE(Project(x,y,z))=Val(x,y,z);
```

o, infine, immagini in cui ogni pixel e' ottenuto come somma del valore dei voxel la cui proiezione cade nel particolare pixel (tecniche additive)[Dre88] [Sab88]; in questo caso si usa una funzione di combinazione dei valori colore per simulare densita' e/o trasparenza delle diverse strutture attraversate:

(3) LOOP

```
(x,y,z)=Next_B-T-F(VoxVol);
IMAGE(Project(x,y,z))=Compose(Project(x,y,z),(x,y,z));
```

dove:

$$\text{Compose}(\text{pixel},(x,y,z))=(1-\text{Opacity}(x,y,z))*\text{IMAGE}(\text{pixel})+ \\ +\text{Opacity}(x,y,z)*\text{Val}(x,y,z)$$

Immagini che rappresentino il contenuto di particolari zone del volume dei dati sono ottenibili per mezzo della definizione di piani di clipping e l'associata modifica della routine di attraversamento del volume.

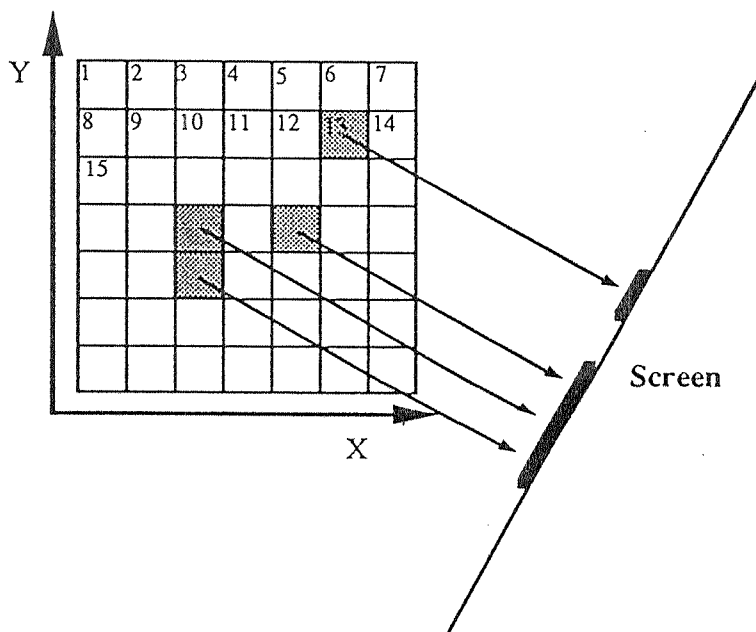


Fig.5.3: Proiezione dei voxel significativi di una slice sul piano di vista (in figura, piano di vista perpendicolare al piano della slice; la numerazione dei voxel indica l'ordine di visita).

Il punto dello spazio immagine in cui, scelta una particolare vista, viene proiettato un qualsiasi voxel (x,y,z) e' calcolato ($Project(x,y,z)$) utilizzando le note tecniche proiettive (moltiplicazione di matrici 4×4 col vettore coordinate del voxel [Fole82]). La funzione $Project$ richiede quindi per ogni voxel nove moltiplicazioni (ove non si faccia uso di tabelle contenenti valori precalcolati di prodotti intermedi [Frie85]); derivando da cio' una grossolana valutazione della complessita' computazionale si ha che per un generico volume $512 \times 512 \times 512$ sono quindi richiesti, per la sola proiezione dei voxel, 1208M prodotti, mentre un meno complesso $256 \times 256 \times 256$ necessita di "soli" 150M prodotti.

Immagini di migliore qualita' (con ombreggiatura delle superfici basata su tecniche depth-shading o su calcolo della normale e simulazione di effetti diffusivi) possono essere realizzate memorizzando in un array di valori profondita' il valore della coordinata Z di ogni voxel proiettato sul piano immagine. Utilizzando tali valori alla fine della fase di proiezione dei voxel potra' essere calcolato un semplice depth-shading dell'immagine o, ove si voglia far uso di piu' sofisticati modelli di illuminazione [Hall89], potranno essere approssimate le normali alla superficie in ogni voxel associato ad un pixel dell'immagine. Le tecniche di shading e di calcolo di normali approssimate saranno descritte nella successiva sezione 5.4.

Nel caso si adotti l'approccio (3) precedentemente descritto, il calcolo dello shading risultera' ovviamente estremamente costoso in quanto non sara' sufficiente calcolare lo shading su N^2 voxel (quelli associati ai pixel dell'immagine, come per i precedenti metodi (1) e (2)); risultera' invece necessario calcolare normali o gradienti per tutti i voxel situati in intorno ad alta discontinuita', cioe' anche per voxel che risulteranno non contribuire al valore di alcun pixel dell'immagine finale. Ove si voglia generare un'immagine con shading delle superfici ed effetti di trasparenza le tecniche BTF additive, se non supportate da HW specializzato, presentano quindi

complessita' comparabile o superiore a quelle della tecnica Ray Tracing, descritta nel successivo paragrafo.

Una limitazione dei metodi proiettivi e' legata al fattore di scala scelto dall'utente nel definire la particolare vista, cioe' al rapporto esistente tra dimensioni del voxel e dimensioni del pixel immagine. Se infatti l'area di spazio immagine coperta da un qualsiasi voxel risulta minore dell'area del singolo pixel l'immagine risultante potra' essere caratterizzata da "artifacts", cioe' pixel a valore non affidabile (ad esempio pixel a valore nullo che non corrispondono ad aree vuote del volume). Il fattore di scala scelto dall'utente (voxel/pixel) dovra' essere non piu' alto di 1:1 nel caso di viste ortogonali, o di 1:SQRT(2) se lo spazio volumetrico e' ruotato intorno ad un solo asse e 1:SQRT(3) nel caso di orientamento arbitrario [Frie85].

Front-To-Back

Le tecniche Front-To-Back (FTB) costruiscono l'immagine attraversando il volume di voxel con ordine inverso a quello BTF (prima i piu' vicini, poi i piu' lontani), con lo scopo di elaborare (ad es. per il calcolo dello shading) solo quei voxel che risultino effettivamente visibili nell'immagine (Fig.5.4). Il vantaggio delle tecniche FTB e' quindi che l'elaborazione dei voxel la cui proiezione coincide con pixel il cui valore colore risulta del tutto definito risulta del tutto inutile. Svantaggio dell'approccio FTB , d'altro canto, e' una maggiore complessita' algoritmica essendo necessario gestire le strutture dati necessarie a determinare efficientemente quali regioni dello schermo siano gia' state effettivamente trattate.

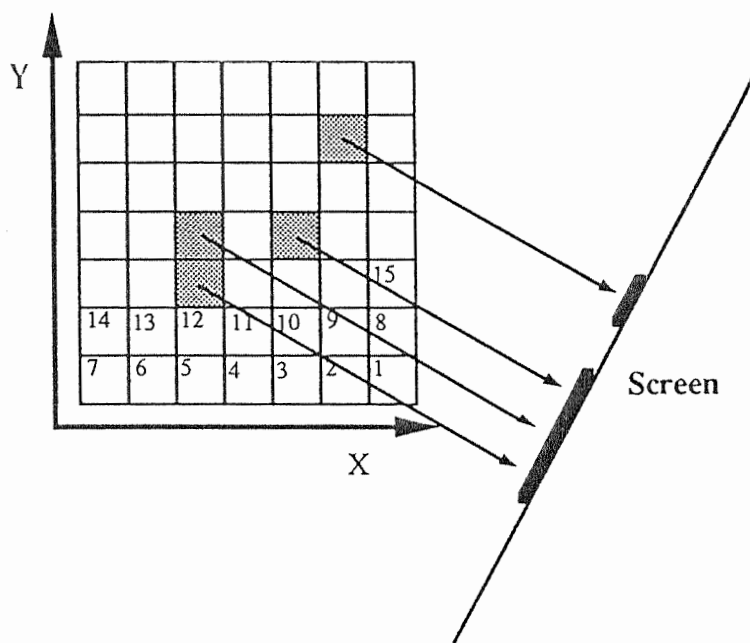


Fig.5.4: Attraversamento di una slice con criterio Front-To-Back.

La piu' efficiente tecnica FTB , l'algoritmo "Dynamic Screen", e' dovuta a Reynolds et al.[Rey87]. L'algoritmo combina un attraversamento completo FTB del volume dei dati, il poter scegliere solo opportune trasformazioni di vista ed, infine, una efficiente tecnica di rappresentazione delle zone dell'immagine che richiedono ulteriore elaborazione (il "dynamic screen"). Alla base dell'algoritmo e' appunto la limitazione nella possibilita' di scelta dell'orientamento delle viste: sono ammesse solo le viste che trasformano le "righe" di voxel del volume (voxel ad ordinata e profondita' z costanti) in righe di pixel paralleli all'asse X dello spazio immagine. Tale condizione non e' limitativa dal punto di vista dell'analisi dei dati, in quanto una proiezione generica puo' comunque essere ottenuta come composizione di una proiezione del tipo precedentemente descritto e di una successiva rotazione dell'immagine ottenuta. D'altro canto il dover gestire solo viste del tipo descritto semplifica molto come vedremo la gestione dei risultati intermedi.

Il metodo di scansione FTB del volume dei dati e' derivabile in modo immediato dal metodo BTF con la semplice inversione del criterio di scansione degli indici, in modo da ottenere un ordinamento parziale inverso. Naturalmente, la lettura ed analisi FTB del volume risulta meno costosa della lettura BTF solo se vengono utilizzate tecniche che permettano di non effettuare le computazioni associate ai voxel che non contribuiscono all'immagine finale. L'algoritmo di Reynolds usa una struttura dati, il dynamic screen, nel quale e' possibile memorizzare informazione sullo stato corrente di ogni pixel dell'immagine: calcolo del colore ultimato o, altrimenti, colore non completamente definito. Poiche' la tecnica FTB richiede comunque la completa scansione e trasformazione geometrica dei voxel, l'algoritmo FTB risulta conveniente solo ove si usi una tecnica di shading e/o simulazione della trasparenza per cui il rapporto tra overhead di gestione del dynamic screen ed elaborazioni di shading non effettuate sia favorevole. L'algoritmo di Reynolds e' inoltre particolarmente efficiente poiche', essendo basato sulla tecnica di rappresentazione Run Length, necessita di un minor numero di accessi alla memoria per la scansione del volume dei dati e riduce sia la scansione del volume che l'accesso al dynamic screen ad operazioni tra intervalli lineari di voxel. In questo modo la complessita' computazionale e' ridotta dello stesso fattore ottenuto a livello di rappresentazione dei dati (il fattore di compressione ottenuto convertendo da volume $N \times N \times N$ alla associata rappresentazione Run Length). La limitazione sulle viste gestibili dall'algoritmo e' appunto necessaria per permettere che ogni intervallo di voxel a valore costante individuato dalla codifica Run Length (sulla stessa riga) venga proiettato su una riga di pixel dell'immagine parallela all'asse X, non richiedendo quindi le costose operazioni di rasterizzazione altrimenti necessarie.

5.3 Ray Tracing

Lo sviluppo di tecniche Ray Tracing (RT) per la sintesi di immagini ad alto realismo ha costituito uno dei campi di maggiore interesse nell'area della Computer Graphics negli ultimi anni.

L'algoritmo RT applicato a modelli geometrici calcola il valore colore di ogni pixel dell'immagine generando un raggio "visivo", cioè un raggio passante per tale pixel ed avente direzione dipendente dalla posizione dell'osservatore, nel caso di proiezioni prospettiche, o dalla direzione di vista, nel caso di proiezioni parallele (fig. 5.5). Il raggio viene quindi lanciato nella scena, alla ricerca di intersezioni con i solidi contenitivi. Il primo solido colpito dal raggio è chiaramente visibile dal particolare pixel associato al raggio: se ne calcola quindi il colore, applicando tecniche di shading e di ricerca di ombre riportate che richiedono in genere il tracciamento di raggi secondari.

Il recente ed ottimo testo di Glassner [Glas89] può senz'altro essere considerato di riferimento, in quanto fornisce una completa trattazione dell'algoritmo Ray Tracing applicato a modelli geometrici.

Tecniche di visualizzazione RT sono state proposte anche per il rendering di dati volumetrici [Fuyi86] [Tuy84] [Levo88] [Schl86] [Upso88] [Sabe88]. Punto di forza dell'algoritmo di RT applicato alla visualizzazione volumetrica è il poter generare immagini, con simulazione di effetti di trasparenza ed evidenziazione delle superfici contenute, senza richiedere l'attraversamento completo del volume dei dati. Infatti, dalla definizione stessa dell'algoritmo RT deriva che il tracciamento di ogni raggio ha termine non appena il valore colore del pixel associato sia stato sufficientemente definito.

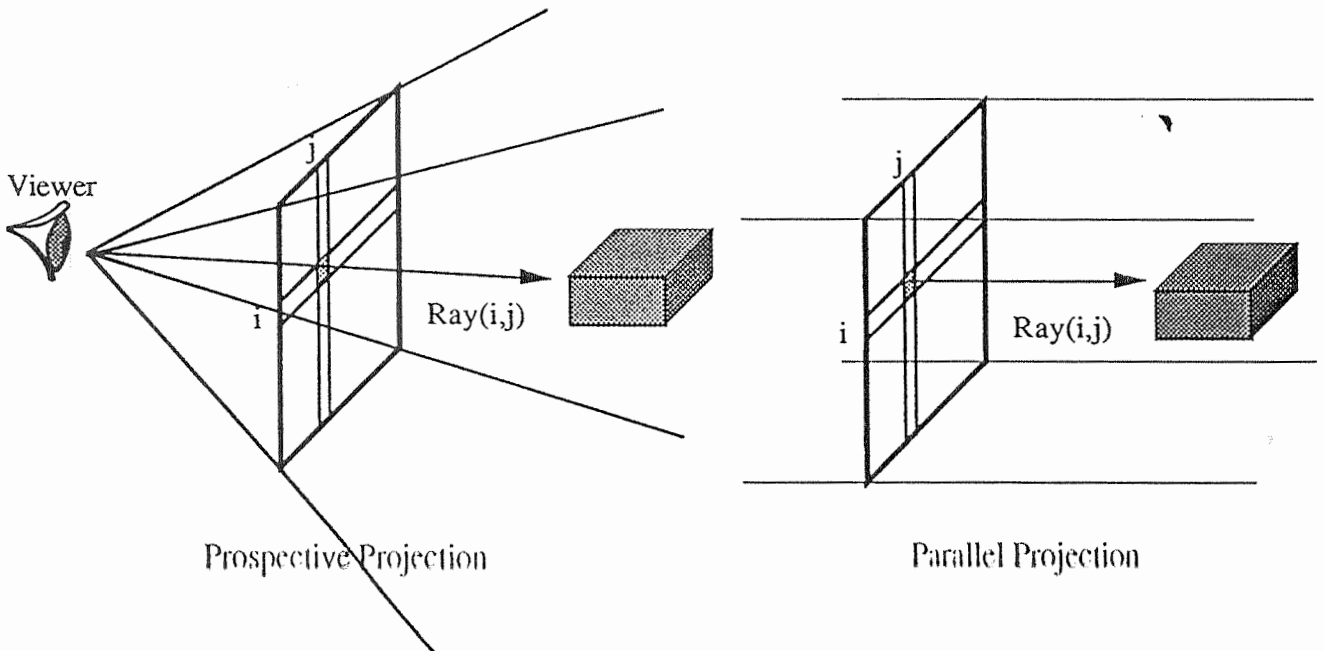


Fig.5.5: Algoritmo Ray Tracing su modelli geometrici, con vista prospettica e parallela

Il modo in cui le tecniche RT sono state applicate alla visualizzazione volumetrica e' chiaramente dipendente sia dalle caratteristiche dei campi applicativi interessati (e quindi dal tipo di immagini richieste), che dal tipo di struttura di rappresentazione usato per i dati volumetrici. L'alta qualita' ("real quality") delle immagini, quale quella a cui ci ha abituato l'applicazione del RT a modelli geometrici, e' generalmente considerata non necessaria; e' infatti ritenuta sufficiente la simulazione di effetti di trasparenza, senza controllo della diffrazione, e della sola componente diffusiva nel calcolo dello shading. Eliminare quindi la generazione dei raggi secondari per il calcolo delle componenti di riflessione speculare ed, eventualmente, anche quelli necessari al calcolo delle ombre riportate riduce notevolmente la complessita' di per se' considerevole del metodo RT.

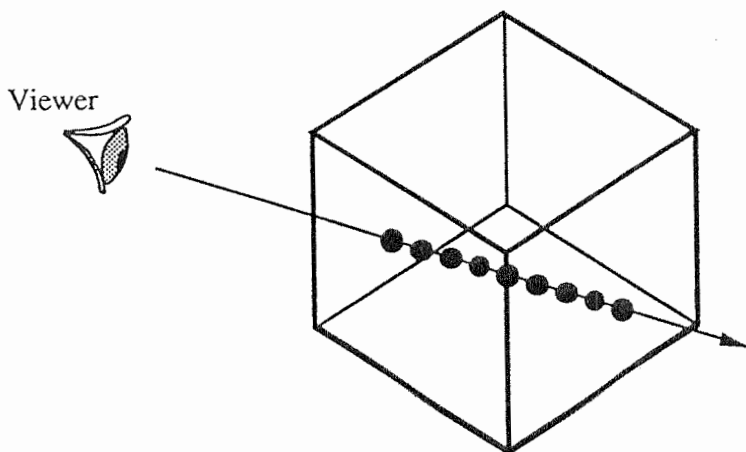


Fig.5.6: Ray Tracing su voxelspace, visita dei voxel (•) attraversati da un raggio.

Il tracciamento di ogni raggio viene realizzato su un modello volumetrico di tipo *voxel space* calcolando le coordinate dei singoli voxel attraversati dal raggio, chiaramente in ordine FTB, e visitando conseguentemente il volume di dati (Fig.5.6). L'efficienza del metodo RT e' chiaramente dipendente dal poter operare accessi casuali al volume dei dati; e' quindi proponibile solo su strutture che garantiscano bassi tempi per l'accesso diretto al voxel. Per il tracciamento del raggio in uno spazio discreto N^3 sono stati sviluppati veloci algoritmi basati su semplici operazioni incrementali e controllo sull'attraversamento di soglie [Fuji86] [Aman87].

L'algoritmo RT puo' essere applicato anche a spazi volumetrici del tipo *cell space* [Ups88]. In tale caso, assumiamo che la griglia m^3 sia immersa in uno spazio N^3 in cui ogni punto del grigliato sia a distanza maggiore di una unita' intera dai suoi vicini (Fig.5.7). Assumiamo che un particolare valore soglia s individui una superficie di separazione tra sostanze diverse. Dopo aver scelto la particolare vista, l'associato rapporto pixel/intervalli del grigliato ed aver definito lo step di incremento, per ogni raggio devono essere individuati tutti punti dello spazio N^3 raggiunti in corrispondenza di ogni incremento di uno step lungo la direzione del raggio. Per ognuno di tali punti si calcola il valore di intensita' associato per interpolazione trilineare del valore dei punti di grigliato piu' vicini. Cambiamenti di valore dei punti

attraversati dal raggio vengono interpretati dall'algoritmo come cambiamenti di densita' e il passaggio da un valore minore di s ad uno maggiore (o viceversa) viene interpretato come attraversamento della superficie associata alla soglia s .

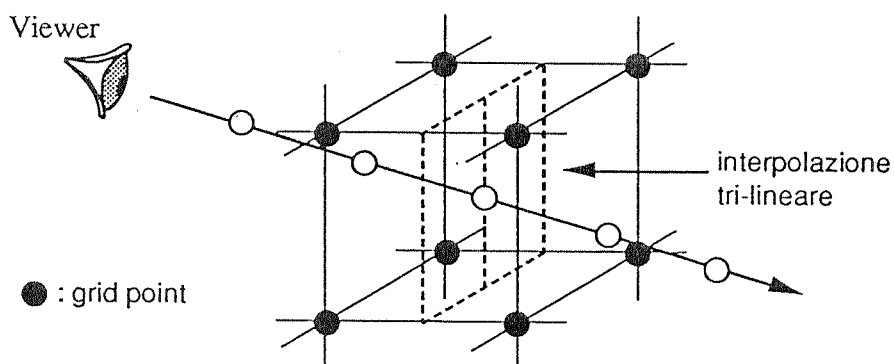


Fig.5.7: Ray Tracing su cell space, calcolo del valore associato ad ogni step per interpolazione dei valori dei punti del grigliato circostanti.

5.4 Tecniche di shading

Parte delle tecniche di visualizzazione descritte permettono di individuare i voxel visibili secondo una particolare vista; in altre parole i voxel posti su una o piu' superfici dell'oggetto rappresentato. Per rappresentare visivamente tali superfici non e' chiaramente sufficiente assegnare al pixel associato al voxel visibile un valore colore predefinito per ogni classe o strato presente nel volume di dati e dipendente dal tipo di interpretazione dei dati operata in classificazione. Affinche' nell'immagine sia presente anche informazione sulla terza dimensione (profondita' nello spazio) e' necessario assegnare al pixel un colore che tenga conto di come la superficie di cui l'associato voxel fa parte sia orientata rispetto alla sorgente luminosa ed all'osservatore. Modulando opportunamente il colore, appunto con tecniche di *shading*, e' quindi possibile rendere nell'immagine l'effetto di curvatura nello spazio delle superfici.

Una difficolta' che si incontra nella valutazione dello shading di modelli volumetrici e' la mancanza nei dati di informazione esplicita sull'andamento delle superfici degli oggetti rappresentati, in altri termini sul valore che la normale alla superficie assume in un qualsiasi voxel posto sulla superficie dell'oggetto rappresentato. Il valore di tale normale, una delle tre componenti principali in gioco nel calcolo dello shading (normale alla superficie, punto di vista e posizione della sorgente luminosa [Hall89]), e' in genere espresso in modo esplicito o facilmente calcolabile nei modelli geometrico-topologici classici (ad esempio per interpolazione di normali note su punti nodali). Nel caso dei modelli volumetrici il valore della normale va invece approssimato, derivandolo con particolari tecniche dai dati stessi. Riguardo ai modelli di shading, che faranno uso di tali normali, anche per il volume rendering si fara' riferimento ai modelli classici proposti per modelli geometrici (in genere Phong shading), utilizzandone a volte semplificazioni in cui si tiene conto del comportamento della sola componente diffusiva ed eventualmente del fenomeno della trasparenza.

Descriviamo qui solo alcune tra le piu' significative tecniche di shading e/o di calcolo della normale proposte, fra cui alcune comuni ad altre aree della computer graphics ed altre specificamente sviluppate per la visualizzazione di dati volumetrici. Una valida descrizione e valutazione delle tecniche di shading per Volume Rendering, confrontate rispetto alla qualita' delle immagini generate, e' nel lavoro di Tiede et al. [Tie90].

Distance-only shading

Il modo piu' semplice per evitare il costoso calcolo delle normali alle superfici visibili e' quello di non tenerne conto nel calcolo dello shading. Assunto che la sorgente di luce sia posta nella stessa posizione dell'osservatore, un sommario andamento delle superfici puo' essere reso calcolando la luminosita' associata ai pixel in funzione della sola distanza dall'osservatore, e quindi dalla sorgente di luce, del voxel associato. La tecnica distance-only shading, originariamente sviluppata per la visualizzazione di modelli geometrici, necessita per ogni pixel dell'immagine sia di informazione sul colore, o sulla classe, del voxel che vi si proietta che della profondita' dello stesso. Notevoli svantaggi di tale tecnica sono sia la perdita dell'informazione di dettaglio che la perdita di informazione sulle linee di separazione tra superfici diverse. Nel primo caso non vengono rappresentate le particolarita' strutturali di ridotte dimensioni della superficie quali ad esempio la tessitura, mentre nel secondo non viene

rivelata informazione significativa poiché tra i punti circostanti l'intersezione tra due superfici non si presentano generalmente sensibili cambiamenti di profondità'. A favore di tale tecnica di shading e' invece l'estrema semplicita' e la bassa complessita'.

Image-based gradient shading (Z-buffer gradient shading)

A partire da una immagine contenente informazione sulle Z, quale ad esempio quella prodotta per l'applicazione del metodo precedente, si calcola la normale alla superficie per ogni pixel (i,j) come vettore gradiente Δz in un intorno 2D del punto immagine (i,j); il gradiente viene in genere calcolato come media tra backward e forward difference (+/-1). Il metodo produce una sufficiente approssimazione della normale e non risulta particolarmente costoso essendo un post-processing sull'immagine a complessita' proporzionale al numero di pixels. Uno svantaggio e' la non invarianza delle superfici rispetto a rotazioni del punto di vista; particolare cura deve essere inoltre posta nell'individuazione di discontinuita' tra superfici diverse in quanto possono essere interpretate dal metodo come zone della stessa superficie in cui la curvatura varia in modo particolarmente sensibile. Sono state definite varianti di tale metodo che permettono di migliorarne notevolmente le prestazioni in termini di qualita' dell'immagine [Bri86].

Constant shading

La normale alla superficie e' approssimata dalla normale alla faccia del voxel che e' proiettata nel particolare pixel [Her79]. In tale modo per ogni particolare vista solo tre distinti valori della normale sono possibili e nell'immagine risultante compaiono molte discontinuita' dipendenti solo dalla particolare rappresentazione e non dall'andamento effettivo della superficie del corpo rappresentato.

Image-based contextual shading

Rappresenta un miglioramento del precedente metodo in quanto le normali in ogni voxel vengono calcolate in un passaggio di post-processing sull'array di normali calcolate con tecnica constant shading [Chen85]. Per ogni pixel (i,j) dell'immagine si calcola la normale come somma di due vettori, il primo calcolato tenendo conto dei due pixel adiacenti a (i,j) rispetto all'asse X, il secondo dei due sull'asse Y. Usando delle tabelle, in funzione dei rispettivi valori delle normali constant shading restituite per (i,j) e per i vicini si ottiene un vettore normale mediato sul valore del constant shading nell'intervallo +/-1 di (i,j). Il metodo image-based contextual shading risulta avere approssimativamente complessita' simile a quella del constant shading, producendo immagini migliori in cui gran parte delle false discontinuita' sono eliminate.

Object-based gradient shading

Tale tecnica calcola la normale associata ad ogni voxel (i,j,k), operando direttamente in object space su un intorno del voxel (i,j,k) [VanH89]. Sono

possibili due approcci diversi, dipendenti dalle caratteristiche dei dati volumetrici da analizzare.

Nel caso di spazi voxel-based classificati, il valore della normale puo' essere semplicemente approssimato esaminando i sei voxel adiacenti alle sei facce del voxel (i,j,k); Tale approccio e' denominato *binary gradient* poiche' viene considerata solo la presenza/assenza di voxel appartenenti alla stessa classe di (i,j,k) nelle posizioni adiacenti [Jen89]. Ogni voxel adiacente che non appartenga alla stessa classe di (i,j,k) genera una versore componente a direzione perpendicolare alla faccia di adiacenza, con origine (i,j,k), con verso orientato in direzione del voxel adiacente considerato e lunghezza unitaria. La normale e' quindi ottenuta come somma dei versori componenti, dando luogo a soli 26 possibili valori diversi per la normale approssimata. Le 26 possibilita' si riducono a 17 se si considerano solo le normali a prodotto vettoriale non negativo con la direzione di vista, cioe' quelle che individuano superfici visibili da un qualsiasi punto di vista. La qualita' delle immagini generabili con questo metodo non e' ottima: data la ridotta casistica di vettori generabili il passaggio da una superficie all'altra non e' graduale e numerose sono le false discontinuita'. Il metodo puo' dare migliori risultati ove siano considerati intorno piu' ampi, aumentando tuttavia il numero dei voxel da visitare e quindi il costo computazionale (Fig.5.8). La complessita' del metodo, $O(kV)$, e' dipendente dal numero V di voxel a cui e' applicato e dalla cardinalita' k dell'intorno di voxel considerato per il calcolo di ogni gradiente. Se applicato nell'ambito di un metodo proiettivo additivo BTF la complessita' che ne deriva, $O(kN^3)$, risulta estremamente gravosa in confronto a quella di altre tecniche. D'altro canto, le normali approssimate ottenute con tale tecnica risultano quelle piu' vicine al reale andamento della superficie dell'oggetto rappresentato. Sia per tale ragione, che per il fatto che ne viene richiesta l'applicazione su $O(N^2)$ voxel, tale metodo risulta uno dei piu' adatti ove si faccia uso di tecniche RT.

Nel caso invece di spazi cell-based (o di spazi voxel-based non classificati) la normale approssimata puo' essere ottenuta calcolando direttamente il gradiente $\Delta Value(i,j,k)$, considerando quale direzione della normale alla superficie passante per (i,j,k) la direzione in cui maggiore e' la variazione del valore associato ai voxel [VanH89]. In questo caso, la lunghezza del vettore ottenuto puo' essere interpretata come "surface detector", cioe' come stima delle probabilita' di passaggio di una superficie per tale punto (maggiore e' la lunghezza del vettore, maggiore e' la variazione di valore nell'intorno considerato e quindi maggiore e' la probabilita' che esista una superficie di discontinuita' tra sostanze differenti). Il gradiente potra' essere sia calcolato come interpolazione trilineare che come interpolazione tricubica; nel secondo caso, considerando piu' valori e combinandoli in modo piu' complesso, si genera una approssimazione piu' smooth al costo, tuttavia, di una piu' alta complessita' computazionale.

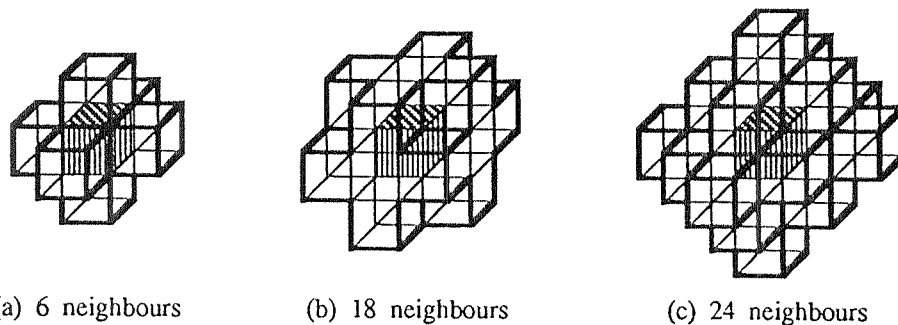


Fig.5.8: Intorni visitabili per il calcolo della normale approssimata.

6. Software commerciale per visualizzazione di modelli volumetrici

Riportiamo in questo paragrafo una breve descrizione dei prodotti commerciali per visualizzazione di dati volumetrici. Per una trattazione delle caratteristiche generali del software di visualizzazione scientifica e per un confronto delle varie realizzazioni si rimanda ad altro lavoro presente nella stessa collana [Pal90].

VoxelView

Recente software commerciale specializzato alla visualizzazione interattiva di dati volumetrici, prodotto da Vital Images Inc. (USA) per le workstation Silicon Graphics IRIS-4D.

Il sistema è dotato di una piacevole interfaccia utente grafica che ne rende estremamente semplice l'uso. Accetta in ingresso modelli volumetrici su grigliati di qualsiasi dimensione (compatibilmente con la disponibilità di memoria). Di tali modelli è possibile visualizzare singole sezioni 2D o viste tridimensionali ottenute per visualizzazione proiettiva con i diversi tessuti evidenziati in trasparenza. Sia i parametri di vista che il controllo della funzione di opacità, associata ai valori contenuti nel volume dei dati, sono gestiti interattivamente dall'utente. È possibile inoltre rendere più visibili le superfici di isovalore sia impiegando le classiche tecniche di illuminazione e shading che isolando dal resto del volume i voxel corrispondenti a tali superfici e visualizzandoli separatamente.

VoxelView può inoltre generare immagini stereo e fornisce gli strumenti per creare e salvare sequenze in animazione (ottenute per rotazione della camera) e per generare istogrammi utilizzabili per l'analisi dei dati e per la definizione degli intervalli o soglie di interesse nella generazione di immagini.

VoxelView permette di generare immagini in tempo reale da volumi considerevoli di dati (ad es. 256^3), con velocità di attraversamento e visualizzazione dei dati volumetrici che vanno dai 300K voxel/sec su workstation Personal Iris, modello di fascia bassa, ad un massimo di 20M voxel/sec sulle più potenti workstation Silicon Graphics dotate di motore grafico VGX. Le notevoli prestazioni del sistema sono dovute ad una stretta integrazione del SW con la sofisticata architettura del sottosistema grafico hardware Silicon Graphics. Il sistema richiede non meno di 16Mbyte di memoria centrale.

Va inoltre riportato che Silicon Graphics rilascia gratuitamente come software di libero uso sulle workstation Power Series il sistema VoxelLab, sistema che fa uso della medesima interfaccia utente di VoxelView e mette a disposizione un sottoinsieme delle funzionalità dello stesso.

Chap Volumes

Pacchetto di Volume Rendering sviluppato dalla PIXAR (USA) per la sua architettura PIXAR Image Computer, una macchina parallela specializzata nella elaborazione di immagini ed in grado di operare anche come

coprocessore grafico. Soluzione abbastanza costosa, in quanto richiede un hardware non economico che a sua volta necessita di una workstation Unix da impiegare come front end.

Il pacchetto e' costituito da una libreria di funzioni grafiche utilizzabili come chiamate di funzioni da programmi in C o tramite un interprete di comandi. Le funzionalita' presenti permettono di leggere dati volumetrici di input (slice del volume di dati), effettuare una serie di operazioni sulle singole slice (classificazione, matting, interpolazione di dati mancanti, resizing) e la visualizzazione per proiezione dell'intero volume (con rotazione, estrazione di superfici, shading). Operazioni di filtraggio e classificazione dei dati possono essere effettuate isolando le zone di interesse ed individuando, tramite uso di look-up-table ed in interazione con l'utente, i materiali costituenti (ogni voxel puo' essere classificato come mix di vari materiali componenti); i dati vengono classificati associando ad ogni voxel, in funzione del rispettivo valore, i rispettivi valori di colore, opacita' ed indice di rifrazione. E' possibile usare anche tecniche statistiche di classificazione (generazione di istogrammi). Rendering diretto dei dati volumetrici classificati, con rotazioni qualsiasi del volume stesso sugli assi cartesiani; zone del volume, o particolari materiali, possono essere isolati e visualizzati separatamente. Sono prodotte immagini che visualizzano il volume come corpo opaco monocromatico, come corpo parzialmente trasparente o che ne visualizzano l'andamento superficiale applicando tecniche di shading.

Application Visualization System (AVS)

Sofisticato ed innovativo sistema di visualizzazione originariamente sviluppato dalla Stellar Computer [Ups89] ed attualmente commercializzato dalla Stardent (USA), costituisce uno dei primi veri ambienti di visualizzazione, e non solo uno strumento di rendering, fornendo la possibilita' di integrare al processo di visualizzazione le fasi di simulazione e di trasformazione dei dati. E' dotato di una interessante interfaccia a finestre. Fornisce dei moduli, applicabili sia a volumi 3D che a singole immagini 2D, per la generazione in real time di superfici di iso-valore, per la visualizzazione di sezioni del volume, per la generazione di viste back-to-front del volume (con risposta quasi interattiva, circa un frame per secondo), per la visualizzazione high-quality del volume (visualizzazione prospettica e modelli di illuminazione). AVS, attualmente fornito come parte del software di sistema (UNIX), opera su SuperWorkstation Stardent e si situa quindi nella fascia alta sia in termini di prestazioni che di costo. Sembra previsto a breve il rilascio del sistema per piattaforme non Stardent.

NCSA Visualization Software

Il National Center for Supercomputing Application (NCSA) dell'Universita' dell'Illinois ha sviluppato due interessanti pacchetti di visualizzazione scientifica, originariamente denominati Image e DataScope. Tali pacchetti permettono di analizzare interattivamente dati (griglie 2D di valori floating point) visualizzandoli sia in forma tabulare che come immagini, eventualmente in animazione; operano su hardware Apple Mac II e sono attualmente commercializzati da una nuova societa', la Spyglass Inc.,

sotto la nuova denominazione commerciale Spyglass Trasform e Spyglass View ed ad un costo estremamente contenuto (circa \$400 ciascuno).

La Spyglass ha inoltre annunciato un nuovo prodotto per la visualizzazione di modelli volumetrici, denominato Spyglass Dicer. Il sistema fornisce solo semplici tecniche di visualizzazione slice-by-slice, con generazione di slice ad orientamento qualsiasi, date le non alte prestazioni computazionali della piattaforma hw. Punti di forza del pacchetto saranno l'integrazione con il software Spyglass ed Apple in generale, il basso costo sia del sistema che dell'hardware e la nota semplicità di uso dei sistemi Apple.

Wavefront Data Visualizer

Prodotto recentemente proposto dalla Wavefront per l'analisi interattiva di dati su griglie tridimensionali sia regolari che non strutturate. Particolarità del sistema è appunto la capacità di visualizzare dati definiti su griglie non strutturate, senza necessità di operazioni di resampling ed allo scopo di mantenerne invariate le caratteristiche informative. I dati possono essere visualizzati definendo sezioni qualsiasi del volume; è gestita anche la visualizzazione di insiemi di particelle.

Il sistema opera su workstation Silicon Graphics, HP/Apollo, IBM e DEC.

7. Bibliografia

- [Aman87] J. Amanatides, A. Woo, "A Fast Voxel Traversal Algorithm for Ray Tracing", *EUROGRAPHICS '87*, G. Marechal (Ed.), Elsevier Science Publisher, 1987, pp.3-10.
- [Asc90] F. Ascani, F.R. Guarguaglini, L. Moltedo, "Una classificazione di metodi di rappresentazione per ambienti di visualizzazione scientifica", *Collana P.F. Sistemi Informatici e Calcolo Parallelo*, N.1/13, 1990.
- [Bri86] S. Bright, S. Laffin, "Shading of Solid Voxel Models", *Computer Graphics Forum*, 1986, No.5, pp.131-137.
- [Chen85] L.S. Chen, G.T. Herman, R.A. Reynolds, "Surface Shading in the Cuberille Environment", *IEEE C.G.&A.*, December 1985, Vol.5, No.12, pp.33-43.
- [Chen89] L. S. Chen, M. R. Sontag, "Representation, display and manipulation of 3D digital scenes and their medical applications", *Computer Vision, Graphics and Image Processing*, No.48, pp.190-216, 1989.
- [Chri78] H. Christiansen, T. Sederberg, "Conversion of complex contour line definition into polygonal element mosaics", *Computer Graphics*, Vol. 12, No.3, pp.187-192, 1978.
- [Dre88] R. Drebin, L. Carpenter, P. Hanrahan, "Volume Rendering", *A.C.M. Computer Graphics*, Vol.22, No.4, August 1988, pp.65-74.
- [Free74] H. Freeman, "Computer processing of line-drawing images", *Computing Surveys*, Vol.6, pp.57-97, 1974.
- [Fren89] K. A. Frenkel, "Volume Rendering", *Communication A.C.M.*, April 1989, Vol.32, No.4, pp.426-435.
- [Frie85] G. Frieder, D. Gordon, R.A. Reynolds, "Back-to-Front Display of Voxel-Based Objects", *IEEE C.G.&A.*, January 1985, Vol.5, No.1, pp.52-60.
- [Fole82] J.D. Foley, A. Van Dam, "*Fundamentals of Interactive Computer Graphics*", Addison Wesley., Reading (MA), 1982.
- [Fuji86] A. Fujimoto, T. Tanaka, K. Iwata, "ARTS: Accelerated Ray Tracing System", *IEEE C.G.&A.*, April 1986, Vol.6, No.4, pp.16-26.
- [Glas89] A. S. Glassner, "*An Introduction to Ray Tracing*", Academic Press, 1989.
- [Gor85] D. Gordon, R.A. Reynolds, "Image Space Shading of Three-Dimensional Objects", *Computer Vision Graphics and Image Processing*, No.29, 1985, pp.361-376.
- [Hall89] R. Hall, "*Illumination and Color in Computer Generated Imagery*", Springer-Verlag., NewYork, 1989.
- [Hem87] D. C. Hemmy, T. R. Lindquist, "Optimizing 3-D Imaging Techniques To Meet Clinical Requirements", *N.C.G.A. Conference 1987*, Philadelphia, March 22-26, Vol.III, pp.69-80.

- [Her79] G.T. Herman, H.K. Liu, "Three Dimensional Display of Human Organs from Computed Tomograms", *Computer Graphics and Image Processing*, Vol.9, , 1979, pp.1-21.
- [Kepp75] E. Keppel, "Approximating complex surfaces by triangulation of contour lines", *IBM Journal Res. Development*, Vol. 19, pp.1-21, 1975.
- [Jack85] D. Jackel, "The graphics PARCUM system. A 3D memory based computer architecture for processing and display of solid models", *Computer Graphics Forum*, Vol 4, pp.21-32, 1985.
- [Jen89a] G.J. Jense, D.P. Hulismans, "Interactive Voxel-Based Graphics for 3D Reconstruction of Biological Structures", *Computer & Graphics*, Vol.13, No.2, 1989, pp.145-150
- [Jen89b] G. J. Jense, "Voxel-based methods for CAD", *Computer Aided Design*, Vol 21, No.8, pp.528-533, 1989.
- [Kaje84] J. Kajeya, B. Von Herzen, "Ray tracing volume densities", *Computer Graphics*, Vol 18, No.3, pp.165-173, 1984.
- [Kauf86] A. Kaufman, "Memory and processing architecture for 3D voxel-based imagery", *IEEE Computer Graphics & Applications*, Vol.8, No.11, pp.10-23, 1986.
- [Kauf88] A. Kaufman, "The CUBE workstation - A 3D voxel-based graphics environment", *The Visual Computer*, Vol. 4, pp.210-221, 1988.
- [Levo88] M. Levoy, "Display of Surfaces from Volume Data", *IEEE C.G.&A.*, May 1988, Vol.8, No.5, pp.29-37.
- [Lore87] W Lorensen, H. Cline, "Marching cubes: a high resolution 3D surface construction algorithm", *Computer Graphics*, Vol. 21, No.4, pp.163-170, 1987..
- [Meag82] D. Meagher, "Geometric modeling using octree encoding", *Computer Graphics and Image Processing*, Vol. 19, 2, pp.129-147, 1982.
- [Mont90] C. Montani, R. Scopigno, "Rendering Volumetric Data using the Sticks Representation Scheme", to be published on *A.C.M.Computer Graphics*, Vol.24, no.5 (?), 1990.
- [Pal90] P. Palamidese, "Classificazione del Software di Visualizzazione ed Animazione", *Collana P.F. Sistemi Informatici e Calcolo Parallelo*, N.1/14, 1990.
- [Rey87] R. A. Reynolds, D. Gordon, L.S. Chen, "A Dynamic Screen Technique for Shaded Graphics Display of Slice-Represented Objects", *Computer Graphics and Image Processing*, No.38, 1987, pp.275-298.
- [Rho90] M.L.Rhodes, "Computer Graphics in Medicine", *IEEE C.G.&A.*, March 1990, Vol.10, No.3, pp.20-23.
- [Sab88] P. Sabella, "A Rendering Algorithm for Visualizing 3D Scalar Field", *A.C.M. Computer Graphics*, Vol.22, No.4, August 1988, pp.51-58.
- [Same84] H. Samet, "The quadtree and related hierarchical data structure", *Computing Surveys*, Vol. 16, No.2, 1984.

- [Schl86] D.S. Schlussemberg, W. Smith, D. Woodward, "Three-Dimensional Display of Medical Image Volumes", *Proceedings of N.C.G.A.*, March 1986.
- [Smit89] A. R. Smith, "Volume graphics and volume visualization: a tutorial", PIXAR Internal Report, May 28, 1987.
- [Tie90] U. Tiede, K. Hoene, M. Bomans et alt., "Investigation of Medical 3D-Rendering Algorithms", *IEEE C.G.&A.*, March 1990, Vol.10, No.3, pp.41-53.
- [Triv85] S. S. Trivedi, "Representation of three-dimensional binary scenes", *NCGA Conference Proc.*, Dallas, pp.132-144, 1985.
- [Tuy84] H.K. Tuy, L.T. Tuy, "Direct 2D Display of 3D Obbects", *IEEE C.G.&A.*, October 1984, Vol.4, No.10, pp.29-34.
- [VanH89] T. Van Hook, "Volume Dislay Methods", *State of the Art in Data Visualization*, SIGGRAPH '89 Course Note, 1989, pp.VIII.1-13.
- [Ups88] C. Upson, M. Keeler, "V-BUFFER: Visible Volume Rendering", *A.C.M. Computer Graphics*, Vol.22, No.4, August 1988, pp.59-64.
- [Ups89] C. Upson et alt., "The Application Visualization System: A Computational Environment for Scientific Visualization", *IEEE C. G. & A.*, July 1989, pp.30-42.
- [West89] L. Westover, "Interactive Volume Rendering", *Chapel Hill Workshop on Volume Visualization*, Chapel Hill, North Carolina, May 1989, pp.9-16.