

Visualizzazione di immagini mediche 3D

I moderni dispositivi per l'acquisizione di immagini biomediche sono in grado di effettuare in tempi molto brevi acquisizioni di tipo volumetrico, cioè che definiscano una regione tridimensionale nello spazio. Questo tipo di dati provengono solitamente da modalità di acquisizione di tipo tomografico, in grado cioè di acquisire volumi di dati di tipo tridimensionale.

Nella maggior parte dei casi, il volume tridimensionale sarà costituito da una serie di fette parallele tra loro ed equidistanziate. Tali fette possono essere memorizzate in file DICOM separati, come avviene tipicamente nella TAC o nella MRI, oppure in file DICOM unici multifetta come è comune nella SPECT.

In ogni caso il volume dei dati darà formato da una serie di volumi elementari, detti voxel, rappresentati da parallelepipedi con base corrispondente alle dimensioni del pixel sulla singola immagine, ed altezza pari alla distanza di acquisizione delle fette. Spesso i pixel avranno forma quadrata, in quanto la risoluzione dell'immagine è di solito la stessa rispetto ai due assi di acquisizione sul piano dell'immagine. Molte modalità di acquisizione hanno invece una risoluzione lungo il piano di acquisizione diversa rispetto alla distanza tra due fette parallele acquisite, sia per ragioni inerenti alla fisica dell'acquisizione (si pensi alla Risonanza Magnetica dove lo spessore della fetta è legato all'intensità del segnale che si riesce ad acquisire) sia per motivi legati al tempo di acquisizione consentito (aumentando il numero di fette cresce in modo proporzionale il tempo di acquisizione). La risoluzione sui tre assi è di solito definita nei file DICOM. Dal punto di vista degli algoritmi di ricostruzione tridimensionale, il caso ideale è quello in cui il volume dei dati ha la stessa risoluzione sui tre assi cartesiani, in altri termini quando i voxel che costituiscono il volume sono, almeno in prima approssimazione, cubici. Nel caso di acquisizioni con risoluzione non omogenea la qualità del processo di visualizzazione tridimensionale può quindi essere fortemente degradata. Molti algoritmi di visualizzazione disponibili nella pratica clinica controllano quindi preventivamente che le caratteristiche del volume siano corrette ed in caso contrario non operano la ricostruzione.

Un caso peculiare è quello degli ultrasuoni 3D. In questo caso il volume dei dati è costituito da piani non paralleli dipendenti dal posizionamento dei piani di acquisizione solitamente a raggiera, nel caso di sonde 3D (a matrice o meccaniche) oppure approssimativamente paralleli nel caso di acquisizioni manuali del volume. In questo caso gli algoritmi di visualizzazione tridimensionale divengono più complessi.

Le fette costituenti possono essere visualizzate in sequenza e la ricostruzione tridimensionale può essere effettuata mentalmente dal radiologo. (MATLAB all'interno dell'image toolbox prevede il

tool *implay* che permette di esplorare un volume 3D di dati in modo interattivo). Questo approccio tuttavia richiede un tempo lungo visto che il volume può essere composto (come avviene ad esempio nella TAC) da centinaia di fette ed inoltre è spesso un processo mentale complesso che spesso non permette di sfruttare a pieno il contenuto informativo presente nelle immagini per fini chirurgici. E' quindi opportuno sviluppare algoritmi che consentano di ottenere delle rappresentazioni su un display 2D che siano in grado di rappresentare in modo credibile e aderente alla realtà il volume acquisito attraverso una singola rappresentazione, auspicabilmente navigabile, cioè rappresentabile da vari punti di vista dall'utente al fine di comprendere (grazie al parallasse di movimento) lo sviluppo 3D degli oggetti di interesse nel loro complesso.

Un approccio molto diffuso in radiologia è la visualizzazione **triplanar**. Nella visualizzazione triplanar dal volume dei dati vengono estratto tre piani tra loro perpendicolari attraverso una operazione di interpolazione. L'origine dei tre piani può essere definita in modo interattivo dall'operatore, che così può esplorare il volume dei dati. La visualizzazione triplanar è tipicamente disponibile su tutte le workstation per l'analisi di bioimmagini.

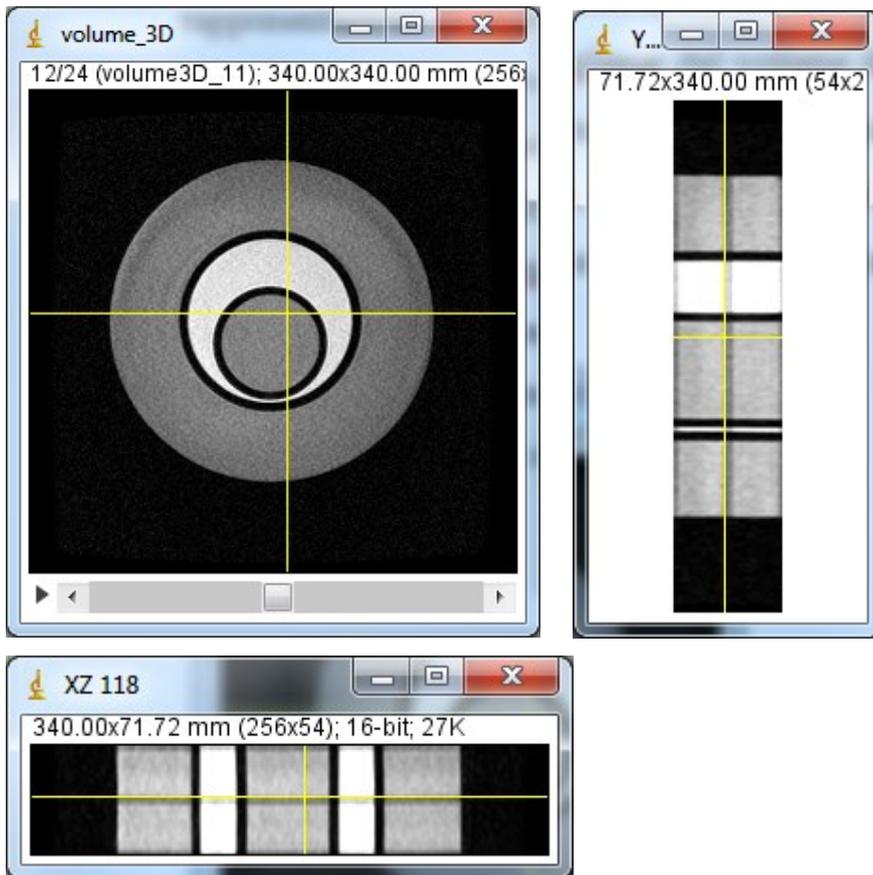


Figura 1: Triplanar view di un fantoccio ottenuta con imagej.

Esistono infine metodi per ottenere una visualizzazione su un display tradizionale 2D su di una unica immagine 2D in grado di fornire il senso di tridimensionalità. Per ottenere lo scopo, esistono varie tipologie di algoritmi, che sono suddivisi in due categorie: metodi diretti (**direct rendering**) ed indiretti (**indirect rendering**). Nei metodi diretti non c'è trasformazione del contenuto informativo delle immagini e la visualizzazione 3D è ottenuta richiedendo all'utente solo una minima interazione (come la scelta del punto di vista e poco altro). Nei metodi indiretti, invece, c'è un'elaborazione del contenuto informativo e spesso è richiesta all'utente una discreta quantità di lavoro e competenze per ottenere un risultato soddisfacente.

Di seguito sono descritte le metodiche più comunemente utilizzate.

Algoritmi proiettivi

Questa tipologia di algoritmi cade in quelle di cosiddetto indirect rendering pur avendo a che fare con metodiche decisamente semplici. Si tratta di algoritmi che imitano il principio alla base della produzione di immagini radiografiche. I dati di partenza sono costituiti dal volume rappresentato dall'insieme delle fette opportunamente distribuite nello spazio (in funzione della metodica di acquisizione).

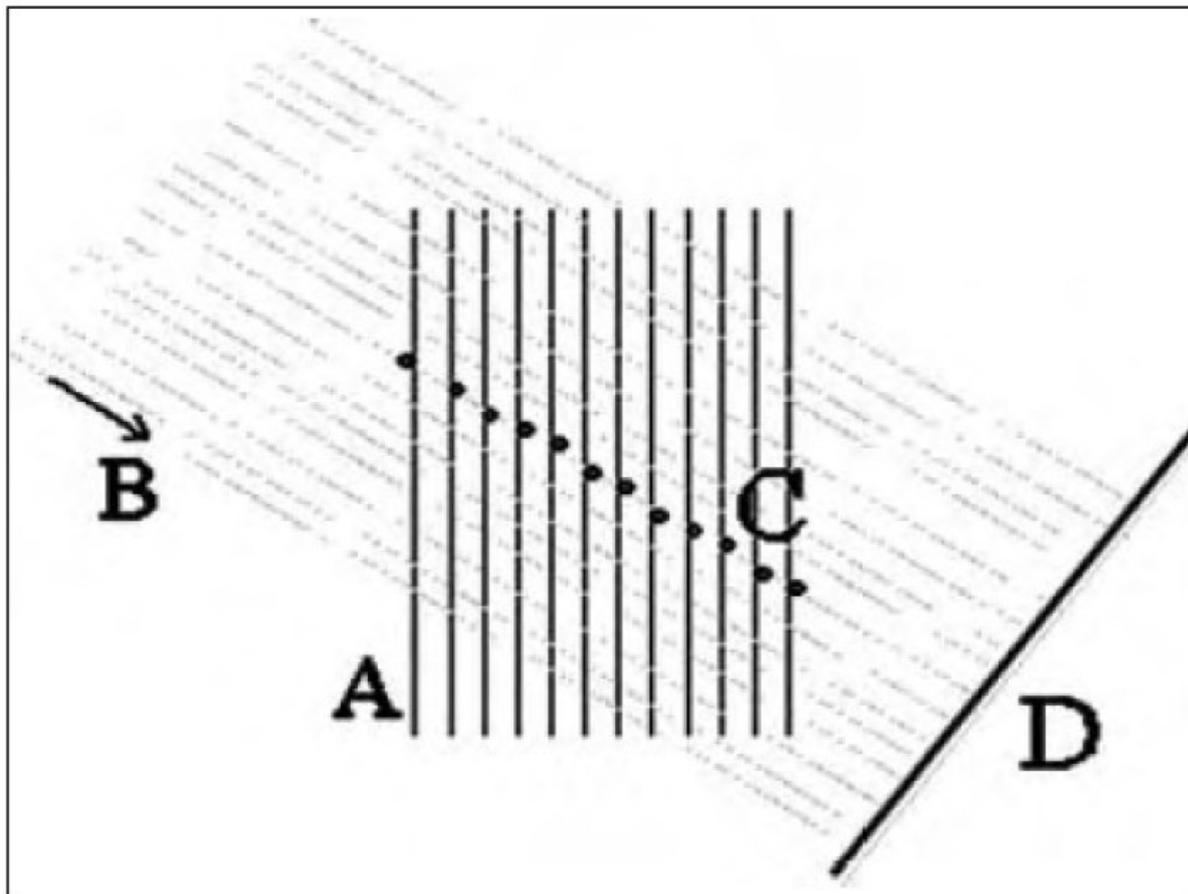


Figura 2. Principio di funzionamento dell'algoritmo proiettivo

Stabilita una direzione di proiezione si individuano una serie di raggi paralleli alla direzione di proiezione che attraversano il volume (le fette). In particolare ogni raggio di proiezione attraverserà tutte le fette in un certo numero di voxel (C). Per ogni raggio avremo quindi una serie di valori corrispondenti all'intensità di segnale del voxel attraversato in ogni fetta. Nell'algoritmo di tipo MIP (Maximum Intensity Projection – Proiezione a Massima Intensità) tra tutti i valori viene scelto quello di valore massimo, che viene riportato sull'immagine proiettata (D) che costituisce la proiezione desiderata. Nell'algoritmo RaySum, tutti i valori ottenuti vengono sommati ed il valore risultante viene riportato sull'immagine proiettata. Altri algoritmi utilizzati sono la proiezione ad intensità minima (mIP), in cui viene riportato il minimo tra i valori e la proiezione ad intensità media in cui viene riportata la media dei valori. Un'ulteriore variante prevede che i raggi di proiezione non siano paralleli ma partano da una sorgente e si propaghino con un certo angolo (proiezione prospettica come nelle immagini rx).

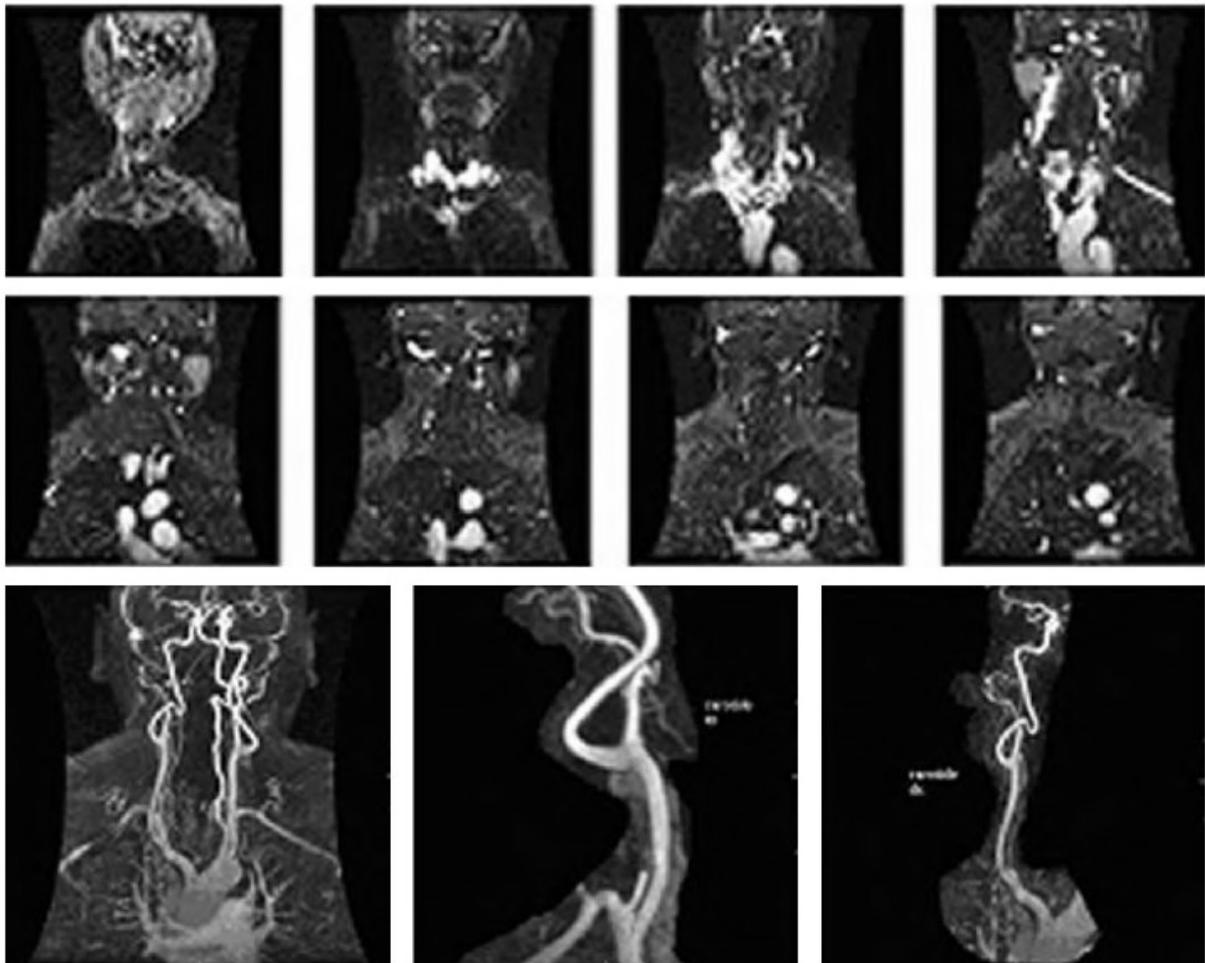


Figura 3. Ricostruzione di carotidi da immagini MRA con algoritmo MIP.

Un esempio di tecnica proiettiva è la visualizzazione angiografica MR, dove viene utilizzata la

tecnica MIP a raggi paralleli. Nella Fig 3 sono rappresentate le fasi di una ricostruzione 3D ottenuta con la tecnica sopra esposta. In alto sono visualizzate 8 delle 72 immagini che costituiscono l'acquisizione di partenza. Si tratta delle sezioni indicate con A in figura Fig 2, e cioè di sezioni parallele che coprono la parte superiore del tronco, il collo e la parte inferiore della testa. In basso sono visualizzate tre diverse ricostruzioni MIP. La prima è una ricostruzione globale con una direzione di vista perpendicolare al piano delle sezioni acquisite, le altre due sono dettagli della carotide sinistra e destra ottenute dall'immagine globale attraverso un diverso punto di vista (e rimuovendo manualmente alcune porzioni del volume). Algoritmi di tipo proiettivo sono implementati nelle workstation ad uso diagnostico e in molti programmi di analisi dell'immagine biomedica liberamente utilizzabili, quali Invesalius, MIPAV, OSIRIX o imageJ.

Gli algoritmi proiettivi hanno come vantaggio principale la loro semplicità, in quanto non richiedono operazioni complesse da parte dell'utente. Sono anche estremamente veloci. Lo svantaggio è che il loro uso è limitato ad immagini in cui l'anatomia d'interesse è costituita da tessuto con un elevato contrasto, rispetto a tutti gli altri, e che nelle vicinanze non ci debbono essere strutture con simili valori di intensità che possano pregiudicare la comprensione dello sviluppo 3D attraverso la proiezione dell'intero volume. Il tipico settore d'uso delle tecniche proiettive è l'imaging vascolare con contrasto, in cui la presenza del mezzo di contrasto nel sangue fornisce un alto contrasto positivo (MRA) o negativo (CT).

* Adattato da V Positano. Postprocessing (Cap. 3). In Risonanza Magnetica del Cuore e dei Vasi. M Lombardi e C Bartolozzi (Editors). Springer-Verlag Italia, 2004. pp. 55-72.

Direct Volume Rendering

Nel linguaggio corrente il termine "volume rendering" è spesso utilizzato in senso generico, comprendendo oltre alle elaborazioni tridimensionali di volume vere e proprie anche le elaborazioni tridimensionali di superficie o gli approcci a proiezione quali il MIP. In senso più rigoroso, per elaborazione tridimensionale di volume o volume rendering si intende un processo che fornisce una rappresentazione tridimensionale di un volume di dati senza utilizzare processi di elaborazione intermedi impliciti o espliciti, e quindi preserva l'informazione contenuta nei dati volumetrici. In pratica non si effettua il processo di segmentazione delle immagini come descritto e come necessario per la metodica di rendering descritta nel paragrafo successivo.

A volte viene utilizzata la locuzione "direct volume rendering" (rendering volumetrico diretto) per caratterizzare il fatto che la visualizzazione tridimensionale opera direttamente sul volume dei dati senza utilizzare rappresentazioni intermedie. Uno dei vantaggi fondamentali nell'uso di questo

metodo è che tutte le informazioni contenute nei dati vengono mantenute durante tutte le fasi di ricostruzione.

Uno svantaggio delle tecniche volumetriche è invece dato dalla grande mole di dati da manipolare nelle varie fasi dell'algoritmo; tale inconveniente porta anche a tempi di elaborazione elevati. Inoltre la produzione di immagini tridimensionali di buona qualità è più laboriosa rispetto a quanto avviene per algoritmi più semplici come il MIP o il surface rendering (paragrafo successivo). Infine il più grande limite di queste tecniche (rispetto a quelle che vedremo nel paragrafo successivo) è dato dal fatto che esse si basano sull'intensità dei voxel e non sulla loro natura e pertanto: possono essere visualizzate strutture non appartenenti all'anatomia d'interesse, poiché queste due hanno simili valori d'intensità, l'immagine può risultare rumorosa ed i bordi delle anatomie non definiti poiché non c'è praticamente mai un passaggio netto in termini di intensità da una struttura all'altra.

Caratterizzazione del volume dei dati

Il volume di dati in ingresso ad un algoritmo di direct volume rendering può essere visto come un insieme di voxel (cioè di volumi elementari) adiacenti tra loro.

La figura 4 illustra lo schema di base di un algoritmo di direct volume rendering: un volume di dati definito da una matrice di voxel viene illuminato da una o più sorgenti luminose. I raggi luminosi provenienti dalle sorgenti vengono riflessi o diffusi dai voxel del volume in base alle loro caratteristiche. Alcuni raggi riflessi viaggiano in direzione di un osservatore e vanno a formare una immagine che rappresenta il volume 3D visto da un osservatore da un certo punto di vista. Il punto fondamentale in un algoritmo di direct volume rendering è quindi la caratterizzazione dei voxel del volume rispetto ad una sorgente luminosa incidente. L'informazione che possediamo è il livello di grigio, o il valore di intensità di segnale, associato al singolo voxel. Questo valore sarà legato in modi dipendente dalla tecnica utilizzata per l'acquisizione del volume dei dati al tipo di tessuto che il voxel rappresenta. Solitamente ad un voxel vengono associate due caratteristiche fondamentali: l'opacità, cioè la quantità di luce che può attraversare il voxel, e lo shading, cioè il modo con cui il voxel riflette o diffonde la luce che lo colpisce.

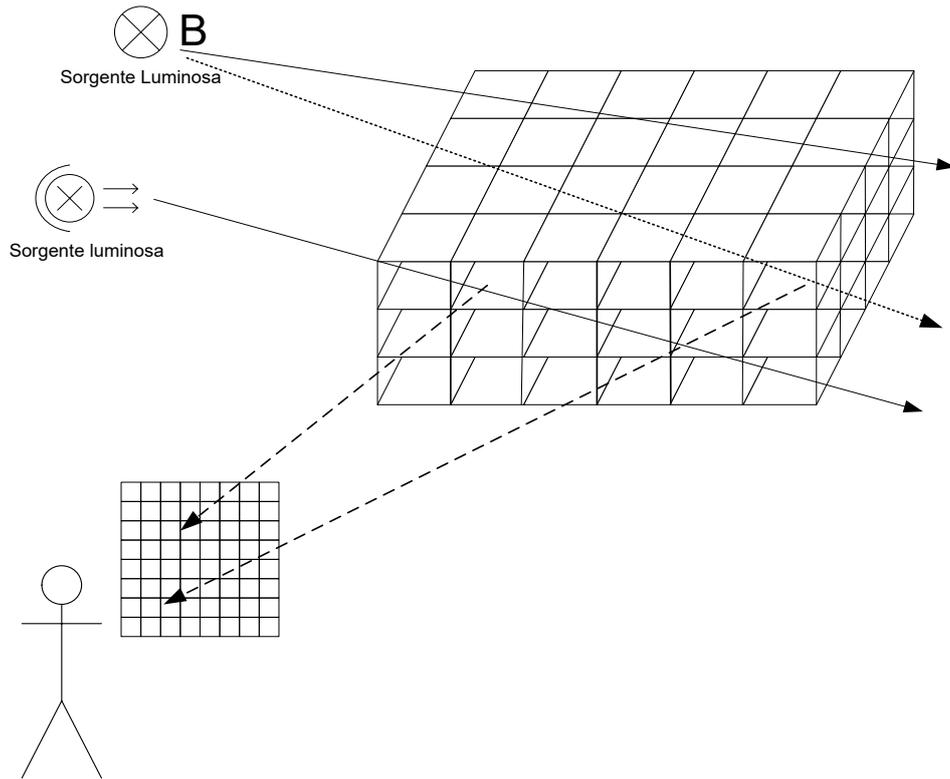


Figura 4: Schema di base di un algoritmo per il direct volume rendering

Funzione di Opacità

L'opacità del voxel è una quantità variabile tra 0 (il voxel è completamente trasparente, e quindi risulterà invisibile) e 1 (il voxel è completamente opaco e quindi nasconderà completamente i voxel dietro di lui). I valori intermedi rappresentano condizioni di semitrasparenza. La corrispondenza tra valore di intensità del voxel e grado di trasparenza può essere decisa dall'utente per ottenere l'effetto grafico voluto. Se ad esempio abbiamo delle immagini TAC cerebrali, ponendo a zero l'opacità dei tessuti molli avremo il rendering esclusivo della scatola cranica, se invece poniamo l'opacità dei tessuti molli uguale ad esempio a 0.5 osserveremo i tessuti molli in trasparenza e dietro ad essi le ossa. Gli strumenti che permettono di ottenere la visualizzazione volumetrica sono di solito forniti di una interfaccia grafica che permette all'utente di aggiustare la funzione di opacità nel modo più opportuno ed esistono anche valori pre impostati tabellati che l'utente può selezionare in modo automatico da una lista predisposta per visualizzare le strutture più comuni e di maggiore interesse.

In figura 5 è mostrato l'effetto della modifica della funzione di opacità. Il volume dei dati è costituito da due cubi, il primo (grande) con valore di intensità pari a 50 ed il secondo interno al primo (più piccolo) con opacità pari a 100. Il due grafici descrivono due possibili scelte della funzione di opacità. Sull'asse delle ascisse ci sono i possibili valori di intensità del voxel, sull'asse

delle ordinate il valore della funzione di opacità corrispondente al valore di intensità sull'asse delle ascisse. Nel caso (A) viene assegnata una opacità nulla ai voxel con intensità pari a 50 e un valore di opacità unitario ai voxel con valore 100. Viene quindi visualizzato solo il cubo più interno. Nel caso (B) ai voxel del cubo esterno viene associato un valore di opacità non nullo (pari a 0.2 circa). I voxel del cubo esterno risultano quindi semitrasparenti, mentre i voxel del cubo interno sono completamente opachi. E' importante notare che la funzione di opacità può essere modificata dall'utente in modo interattivo osservando i risultati della modifica sullo schermo. Questo consente, diversamente di osservare organi o tessuti diversi nella stessa sessione di lavoro.

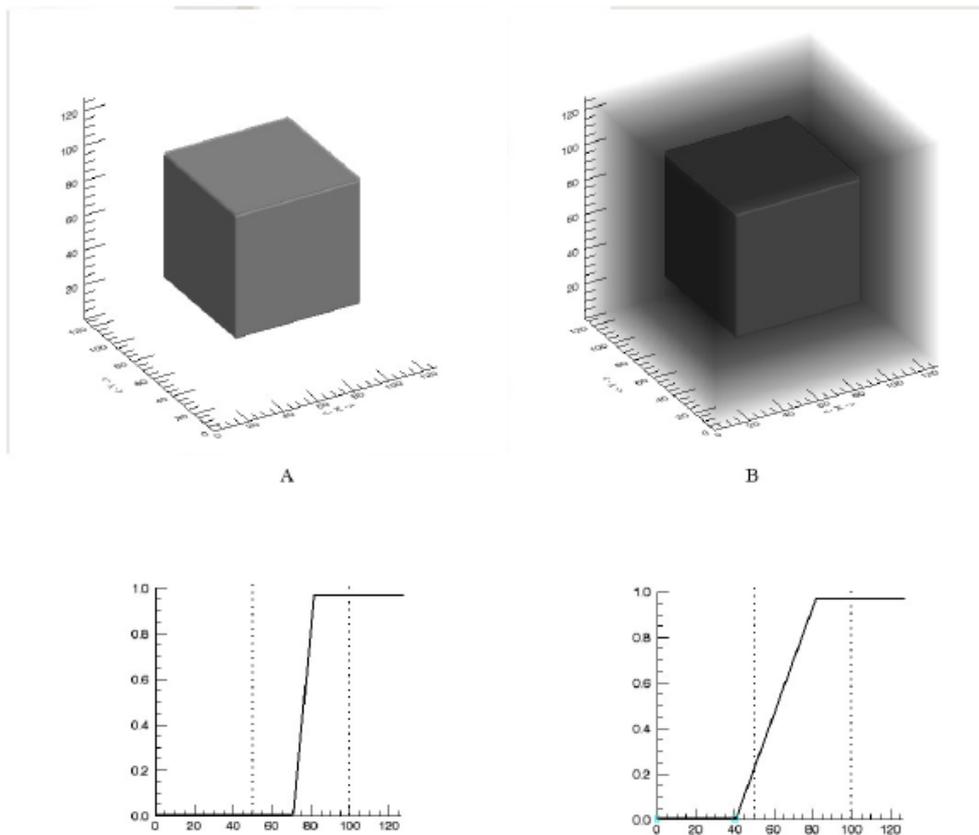


Figura 5: Effetto della funzione di opacità sulla visualizzazione volumetrica con volume rendering diretto.

Funzione di shading

La funzione di shading associa ad ogni voxel i parametri che definiscono come il voxel stesso risponde alle sorgenti di illuminazione che vengono utilizzate. I principi di base del calcolo della funzione di shading sono quelli illustrati nel caso del surface rendering (descritto nel paragrafo successivo): si tratta di calcolare l'angolo tra la luce proveniente da una sorgente ed una superficie

che descrive l'oggetto da visualizzare e quindi calcolare la luce riflessa e diffusa in base all'angolo di incidenza ed alle proprietà che vengono assegnate alla superficie in esame. Quello che differenzia il caso del rendering volumetrico è che, visto che si opera direttamente sul volume dei dati, non è definita alcuna superficie. Bisogna quindi estrarre una "superficie virtuale" dai dati che viene utilizzata solo per il calcolo delle condizioni di illuminazione. Il metodo più semplice è quello di calcolare il cosiddetto gradiente del volume dei dati. Il gradiente si calcola per ogni voxel facendo la differenza tra il valore del voxel stesso ed il valore dei voxel adiacenti. Quindi nelle regioni del volume dove l'intensità dei voxel è costante avremo un valore basso o nullo di gradiente (che significa nessun tipo di interfaccia quindi nessuna riflessione o rifrazione dei raggi di luce), mentre nelle regioni dove sono presenti delle discontinuità avremo valori più alti. Scegliendo la mappa di gradiente opportunamente filtrata come superficie virtuale per il calcolo dell'illuminazione simuliamo l'effetto della riflessione della luce sui bordi degli oggetti. La figura 6 mostra l'effetto della funzione di shading: nel caso A dove si tiene conto solo della funzione di opacità, la forma del cubo interno è poco definita. L'utilizzo della funzione di shading (B) ottimizza la visualizzazione del volume. In generale la posizione e l'intensità delle sorgenti luminose possono essere modificate dall'utente.

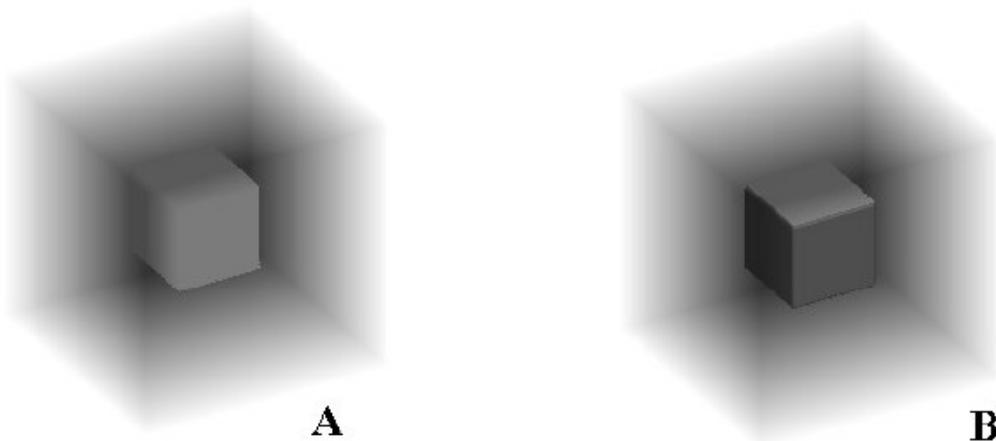


Figura 6: Effetto dell'introduzione della funzione di shading (B) nell'algoritmo di volume rendering diretto rispetto al caso in cui viene applicata solo la funzione di opacità (A).

Esistono numerosi altri approcci possibili per il calcolo dei parametri di shading nel volume rendering diretto, che possono essere anche molto complessi. In generale i parametri di shading sono gestiti in modo automatico dal programma che opera la visualizzazione. In altri termini, l'utente non deve definire la funzione che regola lo shading o al massimo può scegliere tra un numero limitato di funzioni predefinite.

Ray Tracing

Quanto descritto nei paragrafi precedenti deve essere applicato a tutte le direzioni incidenti sul punto di vista.

Nell' procedura di ray-tracing si segue l'approccio inverso rispetto a quanto avviene nel meccanismo naturale della visione: l'occhio dell'osservatore, o meglio la cosiddetta camera virtuale che riprende la scena, viene utilizzata come sorgente di raggi e viene verificato il modo in cui questi raggi intercettano i voxel che rappresentano gli oggetti che ci interessa visualizzare. Per quanto riguarda l'illuminazione, questa viene simulata tenendo conto della posizione relativa del raggio incidente proveniente dalla camera virtuale, della direzione delle sorgenti luminose direzionali e diffuse e dell'immagine di gradiente estratta dal volume dei dati. Il problema fondamentale è che (rispetto ad altri metodi, in particolare quelli descritti nel paragrafo successivo) la quantità di dati da analizzare è molto grande, e pari al numero di voxel che formano il volume dei dati. Ad esempio, un volume di dimensioni 256x256x256 (256 fette parallele ognuna con risoluzione 256x256 pixel) conta quasi 17 milioni di voxel. Gli algoritmi di rendering volumetrico diretto richiedono quindi tempi di calcolo molto alti e l'utilizzo di calcolatori molto veloci e costosi (soprattutto anni fa).

Algoritmi “surface rendering”

L'approccio a superfici (surface rendering) usa primitive bidimensionali, solitamente triangoli o più in generale poligoni, per la rappresentazione di un oggetto tridimensionale. In pratica la superficie dell'oggetto che si vuole visualizzare è approssimata con un certo numero di poligoni adiacenti tra loro. E' l'approccio più diffuso per la visualizzazione di oggetti e scenari 3D nei CAD, nei videogiochi ed in tutte le applicazioni di tutti i giorni.

Poiché sono i cambiamenti delle proprietà ottiche dei materiali a riflettere i raggi luminosi, i quali permettono la visualizzazione e la comprensione della scena, è vantaggioso modellare porzioni con caratteristiche ottiche uniformi identificando le loro interfacce (e tralasciando il volume interno).

Algoritmi ottimizzati allo scopo, che spesso utilizzano schede video dedicate, visualizzano i poligoni che delimitano le interfacce tra gli oggetti (o parti di essi) dando così una rappresentazione realistica dell'oggetto. Opportuni algoritmi simulano gli effetti di diffusione e riflessione della luce (shading), aumentando l'illusione di tridimensionalità. Naturalmente, maggiore è il numero di poligoni utilizzati, maggiore risulta l'accuratezza della ricostruzione. La Figura 7 mostra la rappresentazione a triangoli dei ventricoli destro e sinistro del muscolo cardiaco (destra) e la conseguente immagine tridimensionale come viene visualizzata sullo schermo. I dati utilizzati provengono da immagini di risonanza magnetica.

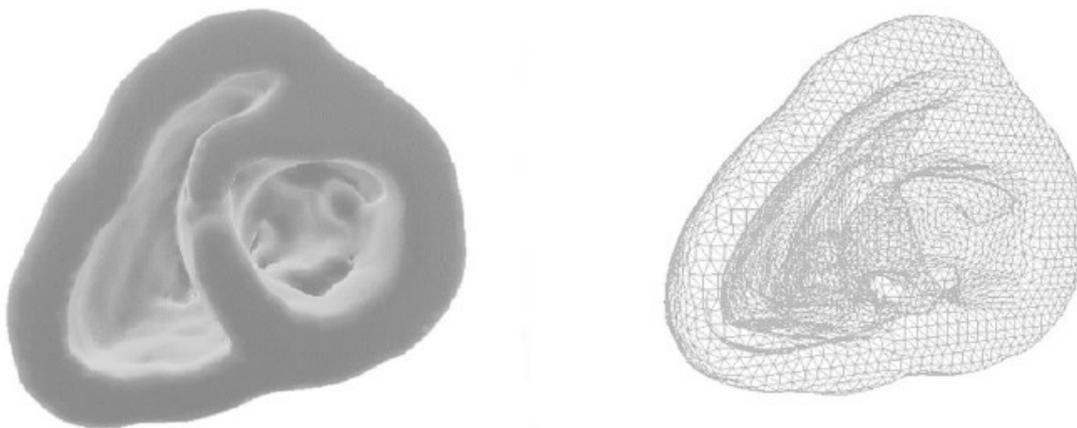


Figura 7: Rappresentazione 3D a superfici del cuore

Il vantaggio principale nella rappresentazione a superfici è la riduzione dei dati da elaborare ad una lista di poligoni ed all'elenco delle connessioni tra gli stessi, con conseguente riduzione dei tempi di calcolo. Inoltre l'approccio a superfici è estensivamente utilizzato in applicazioni commerciali a larghissima diffusione, quali ad esempio i videogiochi, e quindi sul mercato sono disponibili a costi

estremamente limitati periferiche video ad altissime prestazioni che supportano questo tipo di approccio.

Ray Tracing

Nel mondo reale, i raggi luminosi che partono da una sorgente intercettano gli oggetti, che gli diffondono e li riflettono secondo le loro caratteristiche. In dipendenza dalla natura e posizione delle sorgenti luminose e dalla natura e posizione degli oggetti alcuni raggi riflessi (o emessi come nel caso di sorgenti luminose) colpiscono l'occhio dell'osservatore, che ne ricava l'immagine tridimensionale. Nelle simulazioni al computer riprodurre questo processo non avrebbe senso, infatti simuleremmo il comportamento di un numero elevatissimo di raggi luminosi dei quali sono una piccolissima parte sarebbero utili al nostro scopo, che è quello di riprodurre l'immagine tridimensionale come sarebbe vista da un osservatore in una certa posizione. Gli algoritmi di *ray tracing* implementano il processo in modo inverso: utilizzano l'occhio dell'osservatore, o meglio la cosiddetta camera virtuale che riprende la scena, come sorgente di raggi e verificano il modo in cui questi raggi intercettano le superfici degli oggetti che ci interessa visualizzare. Per quanto riguarda l'illuminazione, questa viene simulata tenendo conto della posizione relativa del raggio incidente proveniente dalla camera virtuale, della normale della superficie interessata e della direzione delle sorgenti luminose direzionali rispetto alla superficie. Si tratta di algoritmi complessi una cui trattazione dettagliata esula dagli scopi di questo corso.

Una scena virtuale è costituita da uno o più oggetti ed è definita dalle coordinate dei vertici dei poligoni, da come i vertici sono tra loro uniti per formare i poligoni, eventualmente dalla normale ai poligoni, eventualmente dalle proprietà dei materiali ed altre informazioni utili a rendere più realistica la visualizzazione virtuale. Si riportano di seguito esempi di tali rappresentazioni in due formati di file comunemente utilizzati relativi allo stesso oggetto: un cubo dove ogni faccia è rappresentata con 2 triangoli.

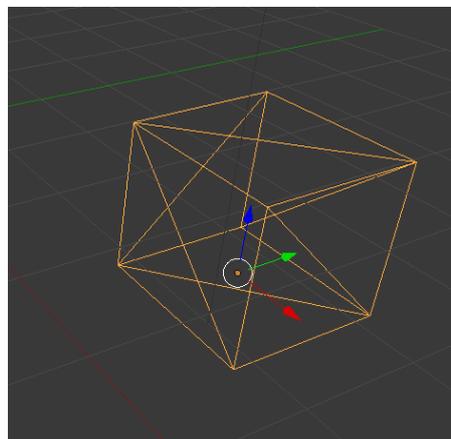


Figura 8: Superficie di un cubo descritta da una serie di triangoli (2 per lato per un totale di 12)

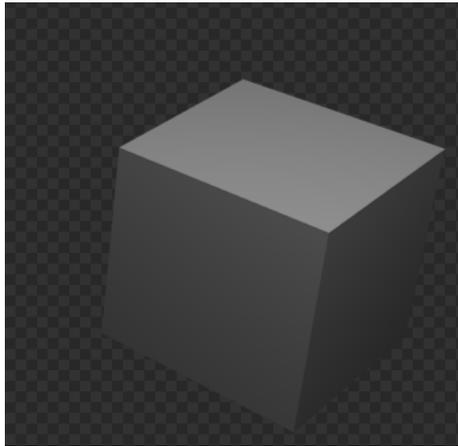


Figura 9: Rendering del cubo associando opportune proprietà al motore di rendering che permettono di ottenere una visualizzazione verosimile.

RAPPRESENTAZIONE DELLA SUPERFICE DELL'OGGETTO IN FORMATO STL

Questo formato per ciascun triangolo riporta le coordinate di ciascun vertice. E' il formato comunemente utilizzato nei processi di prototipazione rapida come la stampa 3D.

```
solid Exported from Blender-2.76 (sub 0)
facet normal -0.000000 0.000000 -1.000000
outer loop
vertex 0.816111 1.267592 -0.452698
vertex 0.816111 -0.732408 -0.452698
vertex -1.183889 -0.732408 -0.452698
endloop
endfacet
facet normal 0.000000 0.000000 -1.000000
outer loop
vertex -1.183889 -0.732408 -0.452698
vertex -1.183889 1.267593 -0.452698
vertex 0.816111 1.267592 -0.452698
endloop
endfacet
facet normal 0.000000 0.000000 1.000000
outer loop
vertex 0.816111 1.267592 1.547302
vertex -1.183889 1.267592 1.547302
vertex -1.183890 -0.732407 1.547302
endloop
endfacet
facet normal 0.000000 0.000000 1.000000
outer loop
vertex -1.183890 -0.732407 1.547302
vertex 0.816110 -0.732408 1.547302
vertex 0.816111 1.267592 1.547302
endloop
endfacet
facet normal 1.000000 -0.000001 -0.000000
outer loop
vertex 0.816111 1.267592 -0.452698
vertex 0.816111 1.267592 1.547302
vertex 0.816110 -0.732408 1.547302
endloop
endfacet
facet normal 1.000000 0.000000 0.000000
outer loop
vertex 0.816110 -0.732408 1.547302
vertex 0.816111 -0.732408 -0.452698
vertex 0.816111 1.267592 -0.452698
endloop
endfacet
```

Vincenzo Positano e Vincenzo Ferrari: Visualizzazione 3D delle immagini volumetriche

```
facet normal -0.000000 -1.000000 -0.000000
outer loop
vertex 0.816111 -0.732408 -0.452698
vertex 0.816110 -0.732408 1.547302
vertex -1.183890 -0.732407 1.547302
endloop
endfacet
facet normal -0.000000 -1.000000 -0.000000
outer loop
vertex -1.183890 -0.732407 1.547302
vertex -1.183889 -0.732408 -0.452698
vertex 0.816111 -0.732408 -0.452698
endloop
endfacet
facet normal -1.000000 0.000000 -0.000000
outer loop
vertex -1.183889 -0.732408 -0.452698
vertex -1.183890 -0.732407 1.547302
vertex -1.183889 1.267592 1.547302
endloop
endfacet
facet normal -1.000000 0.000000 -0.000000
outer loop
vertex -1.183889 1.267592 1.547302
vertex -1.183889 1.267593 -0.452698
vertex -1.183889 -0.732408 -0.452698
endloop
endfacet
facet normal 0.000000 1.000000 0.000000
outer loop
vertex 0.816111 1.267592 1.547302
vertex 0.816111 1.267592 -0.452698
vertex -1.183889 1.267593 -0.452698
endloop
endfacet
facet normal 0.000000 1.000000 0.000000
outer loop
vertex -1.183889 1.267593 -0.452698
vertex -1.183889 1.267592 1.547302
vertex 0.816111 1.267592 1.547302
endloop
endfacet
endsolid Exported from Blender-2.76 (sub 0)
```

RAPPRESENTAZIONE DELLA SUPERFICE DELL'OGGETTO IN FORMATO VRML (.WRL)

Questo formato elenca prima tutti i vertici (Coordinate) che descrivono la superficie poi la modalità per unire i vertici al fine di ottenere i triangoli (coordIndex). I '-1' nelle righe nella sezione coordIndex delimitano i poligoni. Questa seconda modalità non è ridondante mentre la precedente (STL) sì. Questo formato permette di inserire molte proprietà sia per i singoli oggetti che per l'intera scena.

```
#VRML V2.0 utf8
```

```
Transform {
  scale 1 1 1
  rotation 0 0 1 0
  scaleOrientation 0 0 1 0
  center 0.000 0.000 0.000
  translation 0.000 0.000 0.000
  children
  Shape {
    appearance
    Appearance {
      material
      Material {
        diffuseColor 0.8 0.8 0.8
      }
    }
  }
}
```

```
}  
}  
geometry  
IndexedFaceSet {  
  coord  
  Coordinate {  
    point [  
      0.816 1.268 -0.453,  
      0.816 -0.732 -0.453,  
      -1.184 -0.732 -0.453,  
      -1.184 1.268 -0.453,  
      0.816 1.268 1.547,  
      -1.184 1.268 1.547,  
      -1.184 -0.732 1.547,  
      0.816 -0.732 1.547  
    ]  
  }  
  coordIndex [  
    0, 1, 2, -1,  
    2, 3, 0, -1,  
    4, 5, 6, -1,  
    6, 7, 4, -1,  
    0, 4, 7, -1,  
    7, 1, 0, -1,  
    1, 7, 6, -1,  
    6, 2, 1, -1,  
    2, 6, 5, -1,  
    5, 3, 2, -1,  
    4, 0, 3, -1,  
    3, 5, 4, -1  
  ]  
}  
}
```

I due suddetti esempi si riferiscono alla descrizione di una scena virtuale in due formati di file che possono essere utilizzati per la memorizzazione e lo scambio di rappresentazioni di oggetti o interi scenari virtuali. I due file cubo.stl e cubo.wrl forniti agli studenti sono in formato ASCII e la visualizzazione dell'oggetto può essere ottenuta ad esempio con il software gratuito Blender.

La rappresentazione in memoria del calcolatore (memoria principale ma anche memoria sulla scheda grafica) della scena virtuale è un'altra cosa, ma segue metodiche simili per la rappresentazione dei dati a quelle del formato VRML.

La visualizzazione dello scenario 3D comporta la definizione di una o più sorgenti di illuminazione virtuali e di un punto di vista per una camera virtuale con le relative proprietà. La visualizzazione simula dei raggi luminosi emessi dalla sorgente che vengono riflessi dalle superfici (cioè dai singoli triangoli che le compongono) secondo le leggi dell'ottica (da qui l'importanza della conoscenza della normale al poligono) ed eventualmente arrivano all'osservatore. La somma di tutti i contributi dei singoli triangoli produce l'immagine finale. In realtà per evidenti motivi computazionali non vengono computate tutte le riflessioni, ma solo quelle che producono raggi visibili dall'osservatore.

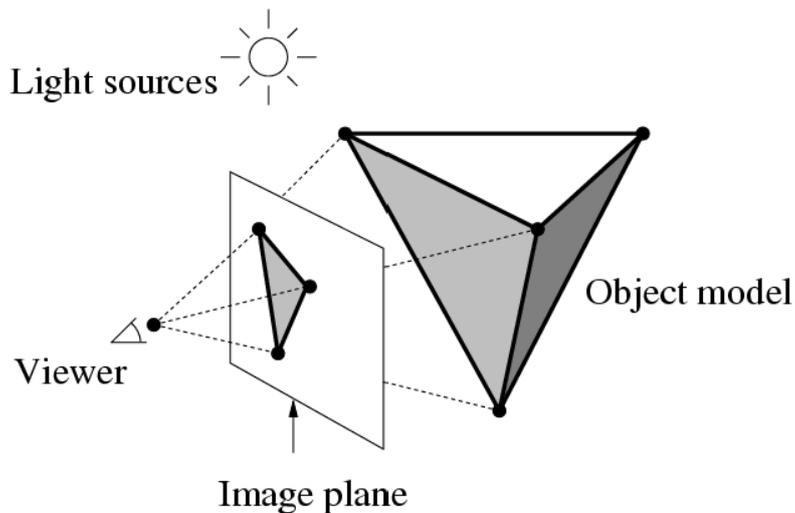


Figura 10: Principio di funzionamento di un algoritmo surface rendering

Metodi di ombreggiatura

Un altro aspetto fondamentale nella visualizzazione 3D tramite algoritmi a superfici è l'introduzione dell'ombreggiatura quale effetto per veicolare l'informazione tridimensionale. Si tratta di un processo fondamentale, in quanto gran parte delle informazioni sulla forma di un oggetto nello spazio sono fornite proprio dalla diversa illuminazione delle superfici che formano l'oggetto stesso. Nel mondo reale, la luce che colpisce gli oggetti viene riflessa o diffusa dagli oggetti stessi e va quindi ad illuminare altri oggetti che a loro volta la riflettono e la diffondono. Inoltre la presenza dell'aria causa fenomeni di diffusione della luce che concorrono alla creazione delle condizioni di illuminazione della scena. Nel mondo reale quindi nel momento in cui è presente una sorgente luminosa tutti gli oggetti risulteranno in un qualche modo illuminati. Nella simulazione dei processi di illuminazione fatta al computer sarebbe impossibile tener conto di tutti i processi che avvengono in natura. Di solito i processi di riflessione multipla non vengono considerati come non vengono considerati i processi di diffusione causati dall'atmosfera.

Di solito avremo una sorgente di illuminazione diffusa, cioè uniforme nello spazio, che consente di illuminare e quindi di rendere visibili tutte le superfici presenti. Avremo poi una o più sorgenti di illuminazione direzionali (spot) che migliorano l'illusione di tridimensionalità.

La figura 11 mostra l'importanza di una corretta illuminazione della superficie per una buona visualizzazione tridimensionale. Un cubo appoggiato su di un piano viene illuminato da una sorgente di luce diffusa (A) a cui viene aggiunta una sorgente direzionale (B). Si osserva come nel caso A la forma del cubo sia a mala pena intuibile, mentre risulta molto meglio definita in B. Infatti l'ombreggiatura da illuminazione diffusa, la quale fornisce un'illuminazione costante per tutte le

superfici di un oggetto, è poco realistica e non offre alcuna possibilità di differenziare le varie superfici. Per riuscire a realizzare un modello di ombreggiatura più realistico si deve perciò passare a considerare sorgenti direzionali e puntiformi.

La differenza sostanziale tra il caso (A) ed il caso (B) è che nel caso (B) si è tenuto conto dell'inclinazione della superficie dell'oggetto rispetto alla luce emessa dalla sorgente direzionale. In altri termini, ogni faccia del cubo ha una inclinazione diversa rispetto alla sorgente direzionale, verrà illuminata in modo diverso e quindi apparirà diversa dalle altre facce.

L'informazione riguardo l'inclinazione della superficie dell'oggetto in un punto rispetto alla sorgente è ricavata dalla normale dei vari triangoli che costituiscono la superficie. Per ogni triangolo quindi viene calcolato l'angolo tra la direzione della luce emessa dalla sorgente e la normale del triangolo stesso, ed in base a questo dato il triangolo viene colorato in modo opportuno.

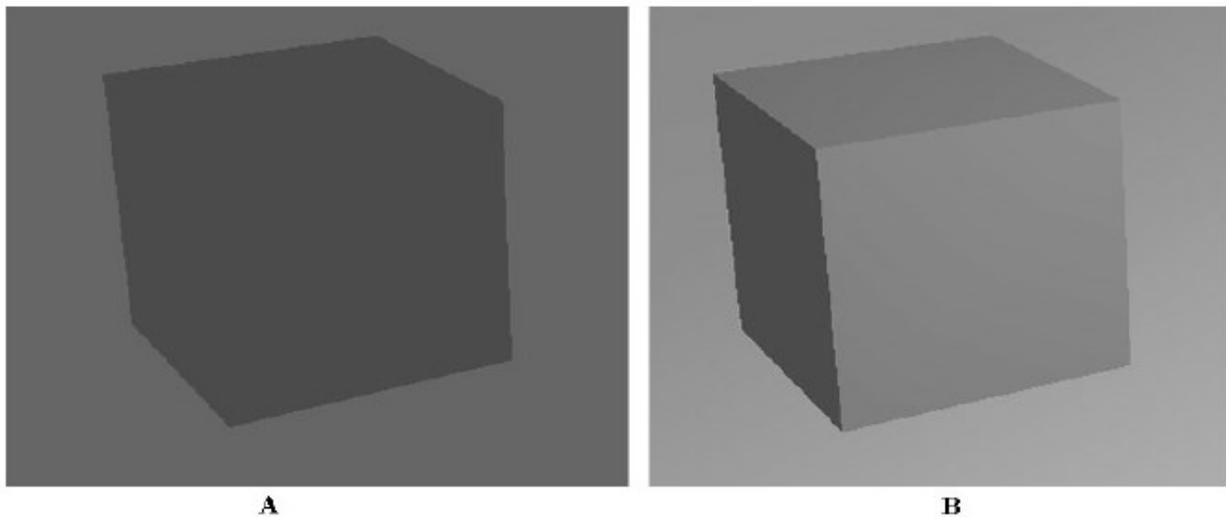


Figura 11: Effetto della funzione di ombreggiatura basata sulla normale al triangolo.

Infine, modelli di ombreggiatura più sofisticati distinguono tra *riflessione diffusa*, dovuta al fatto che una superficie opaca diffonde la luce ugualmente in tutte le direzioni, e *riflessione speculare*, dovuta al fatto che una superficie lucida provoca la creazione di riflessi.

Visualizzazione con superfici di immagini volumetriche

Per ottenere la descrizione delle superfici di interfaccia dei vari oggetti (strutture anatomiche) è necessario identificarle attraverso il processo detto di segmentazione. Un errore nella procedura di segmentazione comporta inevitabilmente un errore nella rappresentazione 3D.

Se consideriamo come esempio un'immagine del cuore, avremo la necessità di identificare la parete cardiaca separandola dai ventricoli e dallo sfondo. Esistono moltissimi algoritmi di segmentazione

(che saranno passati in rassegna nel corso di Elaborazione delle Bioimmagini). Questi algoritmi appartengono a due famiglie, gli algoritmi di segmentazione a soglia e gli algoritmi di segmentazione a contorni. Nella seguente figura sono visualizzati i due possibili approcci su immagini di risonanza magnetica del cuore: in alto, il ventricolo sinistro è definito attraverso i contorni dell'endocardio e dell'epicardio. In basso, viene invece definito attraverso una operazione a soglia in cui la superficie del miocardio è interamente definita da una maschera sovrapposta all'immagine. Come visto in precedenza, i due approcci sono interscambiabili, nel senso che da una segmentazione a soglia si può ottenere l'equivalente a contorni e viceversa.

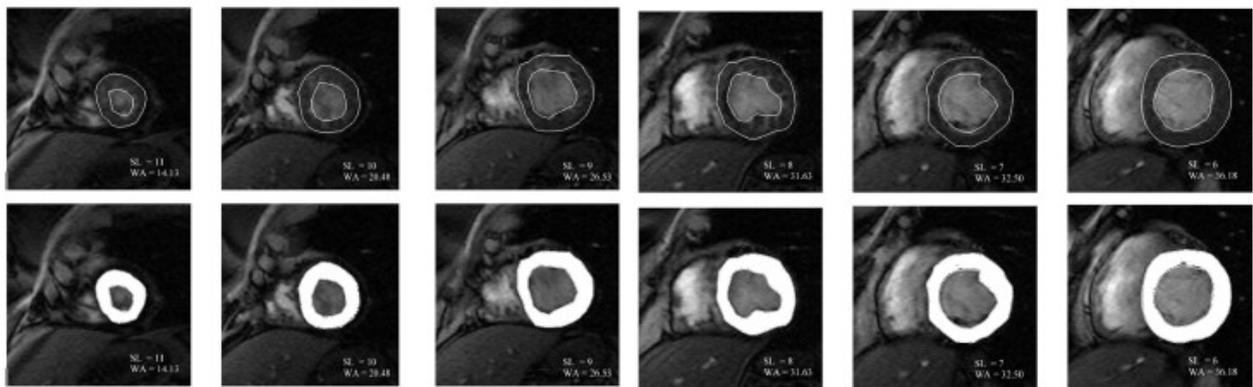


Figura 12: Approccio a contorni (in alto) e a soglia (in basso) per la segmentazione di immagini. L'esempio è riferito ad immagini cardiache con risonanza magnetica.

Nell'approccio a soglia applicato alla visualizzazione 3D, spesso l'operazione di segmentazione è di tipo interattivo. L'operatore deve definire due valori di intensità che identifichino il tessuto da analizzare rispetto al resto dell'immagine. Spesso il programma che viene utilizzato per la visualizzazione fornisce dei valori di riferimento che possono poi essere modificati per migliorare il risultato. È possibile osservare in modo interattivo l'effetto della modifica delle soglie sulle immagini sotto esame. L'approccio a soglia ha quindi il vantaggio di risultare decisamente semplice. Per tale motivo, per alcune strutture è anche possibile visualizzare con metodiche di *direct surface rendering*. In questo caso il software esegue una segmentazione automatica "sogliando" con valori definiti dall'utente tabellati per la struttura d'interesse e poi procede alla ricostruzione della superficie come descritto di seguito.

Lo svantaggio principale è che tutti i pixel dell'immagine vengono elaborati nello stesso modo indipendentemente dalla loro posizione spaziale e soprattutto alla natura della struttura d'appartenenza. Questo comporta la presenza di pixel "spuri" che possono compromettere la visualizzazione e l'interpretazione del contenuto delle immagini. Inoltre, se più oggetti hanno lo stesso valore di intensità verranno selezionati tutti e per identificare un singolo oggetto. Ulteriori

passaggi dell'utente, con ulteriori passaggi di segmentazione, potranno permettere di scindere gli oggetti.

L'approccio a contorni sfrutta le differenze in termini d'intensità che sono quantificate tramite il calcolo del gradiente di intensità attorno a ciascun voxel. Maggiori dettagli saranno forniti nel corso di Elaborazione delle Bioimmagini.

Ricostruzione delle superfici da volumi: Marching Cube

L'algoritmo *marching cube* opera su dati segmentati. In questo caso avremo una maschera tridimensionale, dove ogni elemento (voxel) del volume di dati tridimensionale conterrà il valore 1 se è stato selezionato nell'operazione di segmentazione a soglia e 0 altrimenti. Consideriamo per semplicità un'immagine bidimensionale come quella in figura 11, dove i pixel grigi indicano la regione segmentata ed i pixel bianchi la regione da escludere. La procedura descritta è puramente esemplificativa, nella pratica l'algoritmo è usato solo in 3D. La procedura consiste nel far scorrere lungo l'immagine un quadrato (A) con i quattro vertici che intersecano quattro pixel dell'immagine. Per ogni posizione del quadrato, si va a cercare nella tabella in basso che comprende 16 possibili configurazioni la configurazione corrispondente. Ad esempio il quadrato in posizione A corrisponde alla configurazione 3 (un pixel segmentato in basso a destra) e quindi tratteremo una linea come indicato nell'immagine di destra in figura. Il quadrato B corrisponderà alla configurazione 7 (due pixel segmentati a sinistra) e quindi tratteremo una linea verticale. Facendo scorrere il quadrato su tutta la figura otterremo una approssimazione del contorno della zona segmentata. Le configurazioni 11 e 6 non sono univoche, nel senso che dipendono dal contesto e quindi dalle configurazioni adiacenti. Esistono tecniche particolari che risolvono questo tipo di ambiguità e che non vengono approfondite in questa sede. Come si osserva, l'algoritmo è in grado di ricondurre una segmentazione "quadrettata" ad una rappresentazione di tipo continuo, in cui i bordi della zona di interesse sono abbastanza continui e non presentano angoli retti. Naturalmente la qualità della segmentazione sarà tanto maggiore quanto maggiore è il numero dei pixel utilizzati.

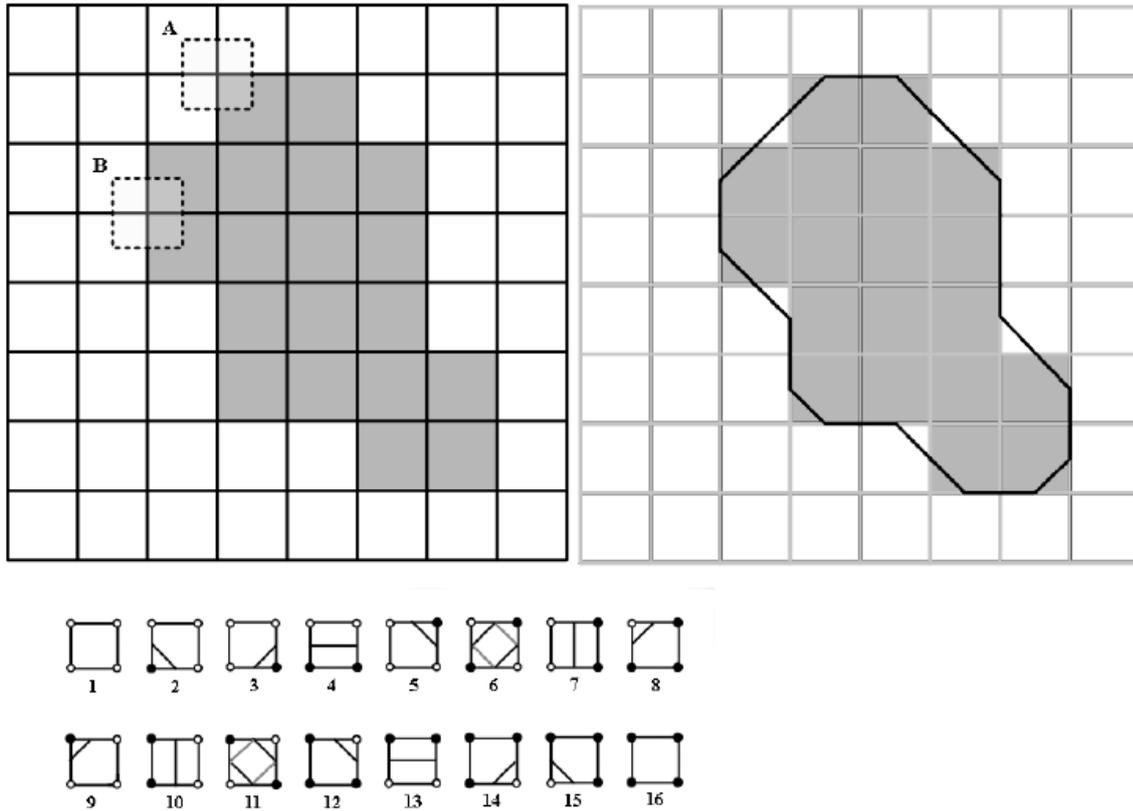


Figura 13: Esempificazione bidimensionale dell'algoritmo marching-cube.

L'algoritmo marching-cube usato nella pratica per la ricostruzione di superfici da volumi di dati è una estensione di quello illustrato in precedenza per immagini bidimensionali. Al posto di un quadrato che scorre su di una immagine bidimensionale intersecando quattro pixel avremo un cubo con otto vertici che interseca otto voxel del volume dei dati. Per ogni posizione del cubo dovremo scegliere una delle possibili 256 configurazioni (al posto delle 16 del caso bidimensionale). Per ognuna delle 256 configurazioni avremo non più una linea ma un triangolo orientato nello spazio. Le 15 configurazioni univoche sono visualizzate in fig. 12, le altre si ottengono dalle 15 attraverso riflessioni e rotazioni del cubo.

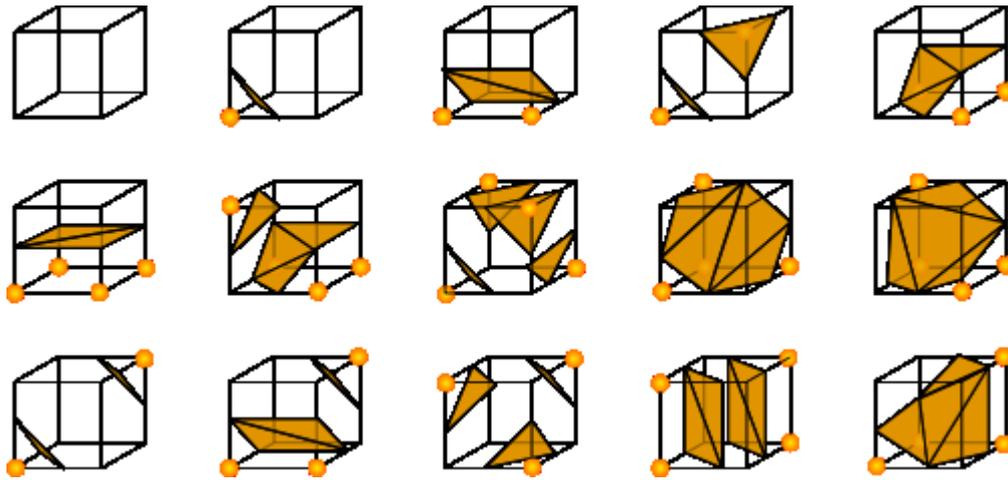


Figura 14: Configurazioni univoche nell'algoritmo marching cube (Wikipedia)

L'unione dei triangoli trovati facendo scorrere il cubo darà la superficie che definisce l'area segmentata. Inoltre l'algoritmo fornisce la *normale* ad ogni triangolo, cioè la retta perpendicolare al triangolo stesso. Questa risulterà utile nella definizione dell'illuminazione. Anche nel caso tridimensionale sono presenti casi ambigui che vengono risolti automaticamente. Molto spesso esistono anche procedure automatiche che consentono di ridurre il numero di triangoli necessari a descrivere la superficie mantenendo una buona qualità della rappresentazione. Questo può essere utile per semplificare e velocizzare le operazioni di visualizzazione.

Applicazioni nella visualizzazione di immagini biomediche

L'approccio a superfici è una modalità di visualizzazione abbastanza popolare nel campo delle immagini biomediche, soprattutto da quando la tecnologia sviluppata per i videogiochi ha messo a disposizione algoritmi di visualizzazione estremamente efficienti e periferiche hardware dedicate a costi estremamente contenuti. Chiaramente l'approccio a superfici può essere utilizzato solo per l'elaborazione delle immagini provenienti da modalità di acquisizione di tipo tomografico, in grado cioè di acquisire volumi di dati di tipo tridimensionale. Algoritmi di surface rendering vengono utilizzati negli ultrasuoni 3D, in particolare nell'imaging del feto. Nella medicina nucleare (SPECT o PET) algoritmi di surface rendering possono essere adoperati per la visualizzazione fine di organie. Il maggiore campo di applicazione riguarda le immagini MRI e TAC, che sono inerentemente topografiche e con risoluzione abbastanza alta da assicurare una buona qualità della visualizzazione tridimensionale.

L'utilizzo della visualizzazione 3D delle immagini è particolarmente utile per finalità terapeutiche oltre che diagnostiche. In particolare per la pianificazione chirurgica. Documentazione relativi a casi clinici reali sarà fornita dal docente.

Display per la visualizzazione tridimensionale

Per quanto elaborato sia un algoritmo di ricostruzione volumetrica, l'immagine risultante che viene visualizzata su di uno schermo non può per sua natura veicolare una informazione di tipo tridimensionale in modo completo. Alcune stazioni di elaborazione offrono allora la possibilità di visualizzare il volume ricostruito attraverso periferiche opportune, dette occhiali 3D, che incrementano l'illusione di tridimensionalità. Il principio di base è quello di fornire ai due occhi dell'osservatore due immagini leggermente diverse, imitando quello che avviene quando viene osservato un oggetto reale. Nel meccanismo della visione, infatti, se un oggetto si trova in un certo intervallo di distanze dall'osservatore, ai due occhi arrivano due immagini da prospettive diverse. Il sistema occhio-cervello integra le due immagini fornendo l'informazione di tridimensionalità (stereoscopia). Il senso stereoscopico non dipende soltanto dalla visione binoculare; ne è la prova il fatto che anche soggetti monoculari e soggetti binoculari quando si fissano oggetti ad una distanza superiore ai 30 metri, dato che a tale distanza si assume che i raggi luminosi siano pressoché paralleli, possono avere il senso della profondità. Tuttavia, la visione stereoscopica può essere molto utile nel veicolare una informazione di tipo tridimensionale. Esistono sostanzialmente tre tipi di occhiali 3D, anche detti occhiali VR (occhiali per realtà virtuale):

Occhiali con filtri ottici: Sono estremamente simili agli occhiali utilizzati nel cinema 3D. Si tratta semplicemente di occhiali in cui le due lenti sono sostituite da due filtri ottici (ad esempio rosso e verde) che fanno passare solo la luce di un determinato colore. Sullo schermo vengono visualizzate due immagini una in rosso ed una in verde sovrapposte, ognuna delle quali viene vista con un solo occhio per merito dei filtri. Analogamente esistono occhiali con lenti polarizzate che sfruttano la diversa polarizzazione della luce. Viene così creata l'illusione tridimensionale a livello cerebrale. Il vantaggio principale di questo tipo di occhiali è il costo ridottissimo e la possibilità di usare un qualunque calcolatore per la generazione dell'immagine.

I moderni occhiali con filtri ottici sfruttano la polarizzazione della luce e non il contenuto cromatico. Questa soluzione non limita pertanto la possibilità di visualizzare i colori.

Occhiali ad otturatori: Gli occhiali ad otturatori montano, al posto delle lenti, dei filtri a cristalli liquidi che, se polarizzati correttamente, diventano completamente neri impedendo all'occhio di vedere attraverso essi. Le due immagini vengono mostrate in sequenza sullo schermo alternando quella destinata all'occhio destro con quella destinati all'occhio sinistro. In contemporanea viene inviato un segnale agli occhiali in modo da oscurare l'occhio non interessato dall'immagine presente sullo schermo. Gli occhiali ad otturatori sono più complessi di quelli a filtri ottici e

richiedono la presenza sul computer di una interfaccia atta a pilotarli. L'immagine 3D che si ottiene è però più realistica, in quanto si possono utilizzare immagini a colori naturali e non virate in rosso o verde.

Occhiali LCD: Lo sviluppo degli schermi a cristalli liquidi ha permesso la costruzione di occhiali LCD, in cui le lenti sono sostituite da piccoli schermi a cristalli liquidi. Spesso negli occhiali LCD la visione è schermata ai lati in modo da impedire l'accesso della luce esterna. In questi dispositivi ad ognuno dei due schermi viene inviata una delle due immagini stereoscopiche. Si tratta di una soluzione ottimale con un costo che con lo sviluppo della tecnologia LCD è in continua diminuzione. Il problema fondamentale con questa soluzione è il fatto che il computer collegato agli occhiali deve essere in grado di fornire contemporaneamente due immagini diverse su schermi diversi, cosa che non è richiesta nei due approcci precedenti. Si tratta quindi di una tecnologia che fornisce prestazioni migliori a ad un costo molto superiore. Il suo uso è attualmente limitato ad applicazioni di realtà virtuale in chirurgia.

Maggiori dettagli sulle tecnologie alla base dei display per visualizzazione 3D possono essere trovati nel seguente articolo:

IEEE TRANSACTIONS ON BROADCASTING 57(2), JUNE 2011, PP. 362-371, DOI: 10.1109/TBC.2011.2130930

Three-Dimensional Displays: A Review and Applications Analysis

Nicolas S. Holliman, Neil A. Dodgson, Gregg E. Favalora, and Lachlan Pockett

L'articolo è scaricabile dalla rete dell'Università di Pisa al seguente link:

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5754618&tag=1