1 Introducció

Els sistemes de disseny geomètric assistit per computador, històricament han centrat el seu treball en la modelització i la representació de la superfície dels objectes. El desenvolupament, però, de noves àrees de CAD, com poden ser la biologia, els sistemes mol.leculars i, principalment, la medecina i les ciències de la terra, ha implicat la necessitat de sistemes de modelatge i de visualització del volum intern dels objectes.

Cal que aquests sistemes cumpleixin tres característiques: les dades han de ser tan precises com sigui possible, per preservar la seva fidelitat amb la realitat, el sistema ha d'interaccionar amb l'usuari en temps real (o quasi real) i les dades han de ser importables des d'altres sistemes per tal d'analitzar-les de forma correlativa i poder comparar diferents conjunts de dades.

Els models més generals desenvolupats per a la representació de les superfícies, com poden ser el model de fronteres, els arbres CSG o els octrees, són difícilment expotables per modelar volums molt heterogenis, com es veurà en l'apartat 3. Per tant, és necessari treballar en noves formes de representació de dades volumètriques.

El Modelatge de Volum *(Volume Modeling)* consisteix en la definició d'una estructura o model matemàtic que permeti de manipular i visualitzar eficientment les propietats del volum dels objectes. Aquestes no han de ser necessàriament homogènies, ni l'objecte ha d'estar delimitat per una frontera contínua. Aquest tipus de modelatge, històricament, es defineix com un problema lateral a la *Visualització de Dades Científiques* (ViSC: Visualization in Scientific Computing). El terme de ViSC es defineix per primera vegada l'any 1987 [McCormick87], en el que s'exposen les primeres conclusions dins d'aquest camp.

S'entén com Visualització de Volum *(Volume Rendering)* el subconjunt d'aplicacions de ViSC relacionades amb la visualització de dades tancades en un volum.

Cal esmentar la confusió de nomenclatura que existeix en tota la bibliografia, ja que s'utilitza el terme de Visualització de Volum per denominar tant la seva visualització com el seu modelatge computacional i la seva manipulació. En l'àmbit del nostre treball, en canvi, entenem com a Modelatge de Volum, la representació i l'operació amb les dades i com a Visualització de Volum, la seva visualització en pantalla pròpiament dita.

Les aplicacions de modelatge i visualització de volum, generalment, consten d'una sèrie detapes des de la captació de les dades fins a la seva visualització. Dacord amb la bibliogarfia analitzada, sha sintetitzat un diagrama de blocs per a una aplicació genèrica tal com mostra a la figura 1.1).

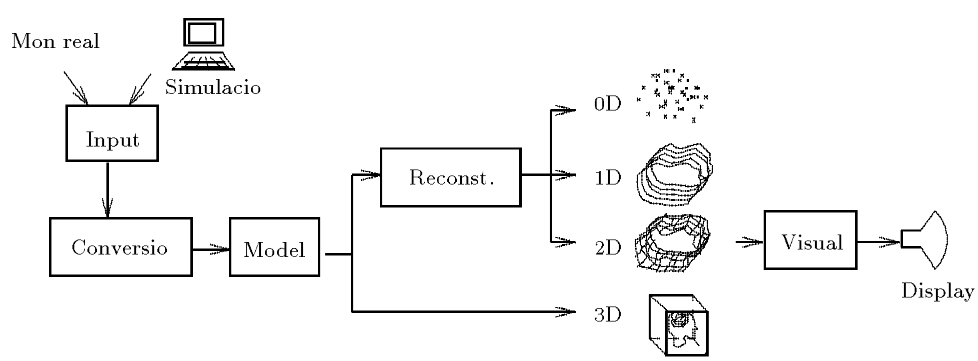

Figura 1.1 *Diagrama de blocs genèric.*

Les dades d'entrada poden ser captades de la realitat, o bé poden haver estat simulades mitjançant un computador. Usualment, les dades obtingudes han de ser processades a fí i efecte d'obtenir un model tri-dimensional. La informació continguda en el model es pot visualitzar mitjançant: punts (0D), rectes o contorns (1D), superfícies (2D) i directament volums (3D). En els tres primers casos, és necessari realitzar un procés de reconstrucció que a partir del model volumètric (normalment voxels), defineixi l'estructura 0D, 1D o 2D que es desitja visualitzar. En qualsevol cas, a partir d'aquesta estructura, es computa el procés de visualització (EPA i Shading) per a generar la imatge final. Les diferents tècniques de visualització ofereixen flexibilitat a l'usuari per a interpretar i analitzar les dades volumètriques.

Lobjectiu d aquest report és presentar una anàlisi de lestat del modelatge i la visualització de volum. En el següent apartat, sestudien les tècniques d'input més utilitzades i el stipus de dades que se n'extreuen. Els diferents esquemes de representació de dades volumètriques trobats en la bibliografia s'estudien detalladament a lapartat 3 i els algorismes de conversió entre models, en lapartat 4. A continuació, a lapartat 5, sanalitzen les tècniques de visualització. Sestudien també els tipus daplicacions en els que és necessari un tractament de volum (apartat 6) i les diferents arquitectures que shan dissenyat per optimitzar el temps de resposta dels sistemes de volum (apartat 7). Finalment, a lltim apartat, s'exposen les conclusions i se n'extreuen les principals línies de recerca en el modelatge i la visualització de volum.

2 Input

El mòdul d'entrada d'un sistema de modelatge de volum consisteix fonamentalment en un procés de recollida de dades per a la seva posterior codificació com a esquema de representació de volum. Com es pot veure en la figura 1, depenent de l'aplicació, aquest procés es realitza de dues maneres diferents: captant dades del món real o simulant dades per computador. El primer cas correspon a aplicacions en les que es pretén representar amb un model computacional un objecte o un fenomen **real**. L'objectiu principal d'aquestes aplicacions és **l'exploració de dades**. Un exemple d'aquest cas és, en la medicina, el modelatge de parts de cossos humans existents, com el cervell [FaZ89]. La captació es fa mitjançant dispositius que permeten de mesurar diferents paràmetres en determinats punts de l'espai, en un instant donat o bé, de forma continua, al llarg d'un interval de temps. Les característiques de les dades depenen, entre altres factors, de les prestacions dels aparells de mesura.

La simulació de dades, en canvi, s'utilitza quan es vol crear un model computacional d'un model **teòric** previament dissenyat amb l'objectiu **d'analitzar** el seu comportament i **verificar** la seva validesa. La simulació per computador d'un model matemàtic de dinàmica de fluïds [LLL89][MKD90], n'és un exemple. Les dades són generades directament per el software de simulació. Determinades aplicacions barrejen les dades de simulació amb les dades reals [MNR91].

En l'apartat 2.1, es descriuen les característiques de les dades, en l'apartat 2.2 les tècniques més relevants de captació de dades del món real.

2.1.- Característiques de les dades

Cal distingir dos aspectes diferents en la caracterització de dades: el referent a la distribució espaial dels punts on es prenen o es simulen les mesures, i el relacionat amb els valors mostrejats a cada punt.

Pel que fa a la distribució dels punts, les característiques més relevants són:

— El nombre de punts mostrejats

— La dimensió de les malles

— El tipus de malla

El nombre de punts mostrejats varia en funció del camp d'aplicació. Així, les dades mèdiques de volum suposen habitualment de l'ordre de 2 millions de punts [SFF89], mentre que les dades geològiques acostumen a formar paquets de 137 millions de punts.

Pel que fa a la dimensió de les malles pot ser bi- o tri-dimensional o fins i tot quatri-dimensional [Ros89] en entorns dinàmics. És frequent, sobretot en l'àmbit de la medicina, que un volum (3D) sigui mostrejat a través d'una successió de seccions transversals (2D), que, en conjunt, formen una malla 3D.

Es poden diferenciar dos principals tipus de malles de punts [Wil91]: les regulars i les irregulars. Les primeres són formades per punts repartits de forma uniforme definint cel.les rectangulars (2D) o paral.lepipèdiques (3D) totes idèntiques. Les malles irregulars es divideixen en estructurades i no estructurades. Les malles estructurades, també anomenades curvilínies [SpK90], són formades per hexaedres de diferentes mides i es poden interpretar com el resultat de l'aplicació d'una deformació no lineal a una malla regular. Les malles no estructurades, entre les que es troben les tetraèdriques són formades per punts distribuits de forma irregular. La definició de cel.les i les relacions de conectivitat entre elles no és implícita en la distribució dels punts. Els avantatges comparats dels diferents tipus de malles són descrits a [WiG90]. Cal destacar en particular, que les malles regulars cúbiques són les que presenten millors prestacions per a ubicar punts i interpolar valors.

Quan la modelització de dades de volum requereix d'una malla regular de punts i les dades captades formen un malla irregular, es fa un pre-procés de les dades d'entrada per tal de crear a partir d'elles una malla regular.

Pel que fa als parametres mostrejats a cada punt, les principals característiques en són:

— El nombre de paràmetre per punt

— El tipus de paràmetre

— La dimensió

El nombre de paràmetres per punt és lligat a l'aplicació. Es parla generalment de dades **monovaluades** quan només hi ha un paràmetre per punt i de dades **multivaluades** en cas contrari. En el camp de la medicina, les dades són habitualment monovaluades, mentre que en aplicacions com la geologia poden haber fins a 30 paràmetres per punt [Elv92]. Els paràmetres mesurats són variats: temperatura, densitat, tenor en diferents components químics, pressió, velocitat..etc. Un criteri per a classificar-los és la seva dimensió: els uni-dimensionals (escalars) com la densitat o la temperatura, els bi-dimensional, com la velocitat o l'acceleració o els tri-dimensionals com els moments d'inercia.

El volum total de les dades depén tant del nombre de punts com del nombre i tipus dels paràmetres associats. Les dades mèdiques ocupen entre 20 i 40 Megabytes mentre que les de meteorologia i sísmica [WoL88] són de l'ordre de Gygas i les de física d'alta energia arriben als Teras.

2.2.- Tècniques de captació de dades

Els dispositius de captació enregistren **senyals** que serveixen per a mesurar directa o indirectament determinades **propietats** o **paràmetres**. Algunes de les característiques més relevants dels aparells de mesura són doncs [Phi92][MaG87]:

— El tipus de senyal que enregistren

— El tipus de paràmetre que aquests senyals permeten mesurar

— El nombre de senyals que poden enregistrar simultàniament

— Si mesuren en continu, al llarg d'un interval de temps o només en un instant donat.

— La seva resolució o nombre de punts que mostregen.

— L'escala a la que treballen

— La precisió de les mesures

— Si són tècniques passives que mesuren efectes naturals, o al contrari actives, que produeixen una alteració artificial del entorn per tal d'efectuar la mesura.

— Si exposen l'entorn on mesuren a l'acció d'agents químics o de radiacions nuclears, o bé si són al contrari no contaminants. Aquest aspecte és especial relevant en el camp de la medicina.

Els dispositius de captació de dades que s'exposen a continuació s'utilitzen fonamentalment en el camp de la medicina. Aquestes tècniques tenen, en gran part, el seu origen en el descobriment dels raigs X per Rontgen, en 1895, i, en 1938, dels raigs X per llesques. Les tècniques més antigues com els ultrasons o la radiografia han estat adaptades per a donar com a sortida un senyal digital en lloc d'una fotografia o d'un senyal de video. La llista que segueix no és exaustiva, conté una relació dels mecanismes de captació més relevants.

CT *(Computer Tomography):* El senyal mesurat a cada punt és la variació del coeficient d'atenuació de raigs X, paàmetre relacionat amb la densitat. Aquesta tècnica és especialment adaptada a sòlids d'alta densitat, com els ossos.

MRI *(Magnetic Resonance Imaging):* Mesura la densitat i mobilitat de protons en un cos. El senyal enregistrat és l'energia emesa pels protons en el seu desplaçament, la qual és convertida directament en una imatge. No emet radiacions ja que l'excitació dels àtoms d'hidrogen és produida per ones de radiofreqüència. És una tècnica adient per a teixits tous, amb un alt contingut d'aigua. La *Ultrafast Magnetic Resonance* és una tècnica més recent amb la que es poden adquirir de l'ordre de 20 imatges per segon. Per això, s'adecua millor que la MR convencional a organs dinàmics com el cor. La **MRA** *(Magnetic Resonance Angiography)* és una variant dels MR que permet d'adquirir imatges de vasos sanguins coronaris i cerebrals per tal detectar possibles taponaments.

PET *(Positron Emission Tomography).* Es basa en la utilització de traçadors associats a emissors de positrons. La distribució espacial dels traçadors es pot establir enregistrant les reaccions dels positrons amb els electrons que es troben. Serveix per a mesurar processos metabòlics. Té una resolució molt baixa.

SPECT *(Single Photon Emission Tomography:)* És també basada en l'utilització de traçadors. Capta com a senyal el nombre de raigs gamma emesos. Serveix per a l'estudi de fluïds, especialment el fluxe de sang i la seva estructura.

Aquestes quatre primeres tècniques retornen, a diferencia de les segueixen, seccions planes transversals dels cossos, a partir de les què es pot reconstruir el volum 3D.

UI *(Ultrasound Imaging):* Es basa en la detecció de les reflexions d'ones acústiques que permeten de calcular profunditats. Dóna per sortida una imatge 2D. Com la MR, presenta el gran avantatge de ser inoqüu. Per això s'utilitza per a la visualització de fetus (ecografies).

TI *(Termographic Imaging):* Enregistra radiacions d'infrarojos emeses a través de la pell com a conseqüència de reaccions químiques.

MSI/MEG*(Magnetic Source Imaging o Magnetoencephalography)* Detecta, gràcies a un biomagnetòmetre, els canvis que es produeixen en els camps magnètics del cervell. El senyal enregistrat és un senyal elèctric que pot ser filtrat i mapejat en un model 3D procedent d'una MR.

Digitalitzadors 3D: En algunes aplicacions de neurocirurgia [Woh92], s'utilitzen juntament amb d'altres tipus d'entrada digitalitzadors electro-òptics 3D que compten amb LEDs (Light Emitter Diodes: diodes emissors de llum) en les extremitats de les pinces i dels fòrceps del cirurgia.

En altres camps com la geologia o la geofísica, s'utilitzen d'altres tècniques [She87] entre les que es troben la cristal.lografia de raigs X i el microscopi electrònic.

Alguns sistemes combinen dades captades amb diversos dispositius, per exemple la *multi-modal medical visualization* [HBP90b] que integra dades procedents de MRI i PET.

3 Modelatge de volum

3.1.- Introducció

Tal i com s'ha descrit en l'apartat 1, els sistemes de modelatge d'objectes, històricament modelitzen i representen dins del computador la superfície dels objectes. Un àmbit de recerca en el que es treballa actualment és el modelatge i la visualització del seu volum intern [Bru89].

El Modelatge de Volum consisteix en la definició d'un model que permeti d'operar (manipular i visualitzar) de forma eficient les propietats del volum dels objectes. Aquestes propietats no tenen perquè ser homogènies, ni l'objecte ha d'estar delimitat per una frontera contínua.

Les operacions que caldria realitzar amb aquest model estan analitzades de forma exhaustiva a [SFF89]. Les principals operacions que es poden sintetitzar en:

- Visualització: transformacions de visualització, eliminació de parts amagades i rendering (veure apartat 5).

– Transformacions geomètriques: escalat, rotacions, translacions, simetries [FDF90].
– Càlcul de distàncies, d'àrees i de volums (per exemple, mesuraments de pelvis [YYO92], volums d'unitat geològiques [SaC89])
– Seccions (per exemple, seccions per un bisturí o per un feix de radiacions en simulacions de cirurgia i tractament radiològic [HeU83] [BTT84], seccions planes per visualitzar [Lev90][VPR92]).
– Operacions booleanes: Unions, interseccions i diferències entre models volumètrics i amb altres models [BrN90].
– Operacions amb altres models [KYC90][Lev90b].
– Conversió a altres models (veure apartat 4).
– Animació (dinàmica de fluïds, simulació de cirurgia, visualització de òrgans dinàmics com el cor [FYZ85] [HiS89] [GeW92] [CrM92] [NiH92]).

Els esquemes de representació més ampliament utilitzats per a la representació de sòlids rígids, com poden ser el model de fronteres i els arbres CSG [ReV82], són difícilment adaptables al modelatge de volum, sobretot si aquest és altament heterogeni. Aquesta adaptació és difícil per que el model matemàtic dels sistemes de representació de sòlids rígids està restringit per la següent propietat: la superfície del sòlid és contínua i separa els punts de l'espai que són del sòlid dels que no ho són. Per tant, és suficient emmagatzemar la superfície del sòlid per a tenir-lo definit. Es tracta, doncs, d'esquemes pensats per a la modelització de sòlids homogenis en el seu interior.

D'altra banda, els sistemes de captació i simulació de volum citats en l'apartat 2, retornen dades distribuïdes en malles de punts [SpK90]. Es pot considerar aquesta malla com la unió d'un conjunt d'objectes elementals (paral.lepípeds, hexaedres, tetraedres...), tots ells homogenis en el seu interior. Basant-se en aquesta hipòtesi, el model de Descomposició Cel.lular [ReV82] de sòlids rígids és directament implementable per a la representació de volum. L'única diferència és que, en aquest cas, cal emmagatzemar el valor de la/es propietat/s del volum que representa.

Més concretament, i per la seva fàcil gestió, les malles de punts més ampliament utilitzades són les regulars, definides com un conjunt de cel.les paral.lepipèdiques adjacents i de la mateixa grandària. Per tant, els models de Descomposició Cel.lular emprats són el d'Enumeració Espaial (Voxels), i, com a compactació d'aquests, el model d'Octrees.

De fet, en la bibliografia consultada, els únics sistemes de representació de volum citats són el model de Voxel i el d'Octrees, una descripció dels quals es pot trobar en l'apartat 3.2. Cal esmentar que aquests models són aproximats ja que estan construïts a partir d'un conjunt discret de punts mostrejats. Aquesta aproximació, lligada als propis sistemes de captació de dades, recorda l'aproximació realitzada quan s'utilitzen els models de Divisió Cel.lular per a representar superfícies de sòlids rígids [ReV82][BrN90].

Un dels desavantatges d'aquests models és l'elevada quantitat de memòria necessària per emmagatzemar, amb una precisió acceptable, la informació. Aquest fet provoca el desenvolupament d'algorismes de compactació en 3D, que són extensions d'algorismes de codificació d'imatges (2D). En l'apartat 3.3 s'analitzen aquests mètodes de compactació [Mur92].

Quan els sistemes de representació citats s'utilitzen per enmagatzemar volum realment tancat per una superfície, aquesta queda codificada com una propietat més dins d'una cel.la. Encara que existeixen algorismes que poden extreure una aproximació de la superfície a partir del model de voxels (apartat 4), existeixen aplicacions en les quals pot ser necessària la codificació explícita de la superfície. En aquest aspecte, s'han desenvolupat sistemes híbrids de codificació de la superfície i del volum (apartat 3.4).

Altres aplicacions, com l'adaptació de pròtesis en medicina, requereixen la integració de models geomètrics dissenyats amb sistemes de modelatge de sòlids (pròtesis) i dades volumètriques (l'ós on s'ha d'adaptar la pròtesi). En l'apartat 3.5 s'estudien sistemes mixtes que lliguen el modelatge de sòlids i el de volum.

Finalment, en l'apartat 3.6, s'extreuen les conclusions del capítol i s'exposen les principals línies de recerca obertes en aquest camp.

3.2.- Models de Voxels i Octrees

El model d'Enumeració Espaial o Voxels consisteix en un conjunt de cubs de la mateixa grandària distribuïts dins d'una malla regular fixa amb eixos paral.lels als coordenats [Req80]. La representació del model consisteix en una matriu de voxels, per cadascun dels quals s'emmagatzemen els valors d'un conjunt de propietats. S'anomena resolució al nombre, n, de voxels existent en cada eix coordenat.

Com a caracterització d'aquest model es pot concretar que:

- És aproximat. Donat un volum a representar només s'emmagatzemen les seves propietats en un conjunt discret de cel.les. Aquets aproximació, però, és inherent als aparells de captació d'informació (apartat 2).
- Requereix una gran quantitat de memòria per al seu enmagatzemament. Si n és la resolució del model, l'espai utilitzat és de l'ordre de n^3. És per aquest fet que, encara que els aparells de captació de dades poden proporcionar més quantitat de dades, generalment es treballa habitualment amb una resolució de $n = 8$ [Kau88b].
- Les operacions sobre el model són de fàcil implementació [ReV82], si bé la seva complexitat respecte al temps de càlcul és funció de la seva resolució. Aquest és, doncs, un altre motiu pel qual es tendeix a treballar amb baixes resolucions.

Per tant, el model de voxels, per a la seva simplicitat, és extensament utilitzat. La implementació d'operacions sobre el model s'ha centrat en la seva visualització i un dels seus desavantatges és la manca d'operativitat en temps real [Rho90] i la seva poca precisió. Les dues condicions no són assolibles simultàniament donat que un increment de precisió en la visualització requeriria un augment de la resolució del model i, per tant, implicaria un augment del temps de càlcul de totes les operacions. S'investiguen tècniques d'acceleració d'aquests càlculs via *software* i via *hardware*.

A efectes de compactació del model i donat que poden existir voxels veïns amb propietats anàlogues, es poden utilitzar models amb cubs de diferents grandàries per codificar el volum. En un dels cubs del model s'agruparia un conjunt de voxels veïns. Aquest és el propòsit del model d'Octrees.

Un Octree és una estructura d'arbre octal jeràrquic generada per una subdivisió recursiva d'un univers cúbic finit. En aquesta estructura cada node és una fulla o té vuit fills. L'arbre divideix l'espai de l'univers en cubs interiors i exteriors a l'objecte. L'arrel de l'arbre representa l'univers (cub d'aresta 2^n). Aquest cub es divideix en vuit cubs iguals d'arestes $2^{(n-1)}$, anomenats octants. Si un octant està parcialment plé d'objecte, es subdivideix en vuit cubs ([Sam90],[Mea82] i [Bru89]).

L'avantatge d'aquest model en front del de voxels és que compacta la informació [Nav86], sense complicar les operacions [Sam90]. En [WiG92] es fa una anàlisi de la compactació de memòria obtinguda i es conclou que aquesta és més o menys efectiva depenent de l'aplicació, és a dir, del grau d'heterogeneïtat del volum.

3.3.- Mètodes de compactació d'informació

Donada la similitud entre imatges (conjunts de pixels) i models de voxels, s'està estudiant la possibilitat d'utilitzar tècniques de compactació de la informació similars a les emprades en tractament d'imatges.

De fet, totes les tècniques de filtratge d'imatges (Fourier, cosinus, wavelets ...) són directament ampliables a 3D. Aquestes tècniques consisteixen en transformar l'espai de voxels a un sistema de variables el més independent possible, tenint com a base una família de funcions ortonormals. El Teorema de Karhunen-Loève dóna la família òptima de funcions ortonormals, però la seva implementació es molt costosa. Per aquesta raó s'utilitzen transformacions aproximades com la transformada de Fourier, la transformada del cosinus i, més recentment, les funcions de *wavelets*. Existeixen mètodes per l'obtenció ràpida de la transformada de Fourier i la del cosinus. La transformació mitjançant *wavelets* es genera per translacions i escalats diàdics d'una única funció predefinida [Mur92].

L'espai freqüencial és l'espai definit per la base de funcions ortonormals donada per una certa transformada. Per tal de comprimir la informació, s'eliminen els termes menys significatius de la base de l'espai freqüencial i, per tant, es pot codificar l'espai volumètric amb menys nombre de bits per voxel.

Una possible línia d'investigació és l'anàlisi de la viabilitat d'utilitzar la representació en l'espai freqüencial com un model per sí mateix. També caldria investigar els models mixtes voxels (octrees) - model freqüencial. També caldria analitzar la pèrdua d'informació en la conversió entre models (per exemple, voxels-wavelets), la compactació obtinguda i la complexitat de les operacions clàssiques (visualització, seccions, càlculs de volums,...) en l'espai freqüencial.

3.4.- Models híbrids superfície - volum

Moltes aplicacions, com l'estudi de capes estratosfèriques [Wat87][PuB92], tracten el volum tancat dins d'una superfície de complexitat variable. En els models de voxels o d'octrees, analitzats a l'apartat 3.2, aquesta superfície es representa amb un valor determinat d'una de les propietats que s'emmagatzemen en el voxel.

Per fer operacions amb aquestes superfícies, cal desenvolupar algorismes que les reconstrueixin de manera aproximada a partir dels voxels que la contenen (apartat 4.1). Aquests processos son costosos en temps de càlcul i, a més, un cop obtinguda la superfície sols es disposa d'un model de fronteres sense interconnexió directa amb el volum. En determinades aplicacions és necessari tenir accés als dos tipus de models (superfície-volum) simultàniament. Per exemple, en geologia [PuB92], per detectar bosses de petroli cal poder codificar les propietats interiors a una unitat de sedimentació i a la vegada la seva superfície.

Per a solucionar aquest problema s'estan investigant models híbrids que codifiquin simultàniament el volum i la superfície i les operacions a realitzar sobre ells. Aquesta àrea de recerca, però, encara està poc desenvolupada.

En [Bru89] es presenta una proposta de models híbrids basats en utilitzar models d'octrees de diferents tipus (diferents nodes terminals) per superfície i volum segons la complexitat de la frontera i el grau d'heterogeneïtat de les dades volumètriques.

3.5.- Models mixtes sòlids - volum

A moltes aplicacions es necessari interaccionar entre un model de volum (actualment, voxels) i un objecte dissenyat amb un sistema de CAD (generalment, model de fronteres). L'interacció entre els dos models no consisteix només en la visualització simultània del sòlid i el volum, sinó en altres operacions com la detecció del volum intersectat pel sòlid o l'adaptació del sòlid en el model volumètric. Exemples d'aquestes aplicacions són, en l'àmbit de la medicina, el disseny de pròtesis, els tractaments amb cons de radiació o la simulació d'operacions amb bisturí [KYC90].

Una primera estratègia per a solucionar-ho és codificar ambdós models en un mateix sistema de representació. Les solucions actuals consisteixen simplement en la conversió d'un model de representació a l'altre: *voxelitzar* el sòlid [KaS86][Kau87][Kau88a] o bé extreure isosuperfícies del model de voxels *(poligonalitzar)* [RKR87][LoC87][FKU77]. L'avantatge d'aquestes tècniques és que es treballa amb un únic model. Tenen, però, dos desavantatges: un increment de memòria i una pèrdua d'informació en la conversió entre models. L'increment de memòria és provocat per l'enmagatzemament del model mixte. La pèrdua d'informació és causada bé per la poligonització que sols reconstrueix un petit subconjunt d'informació continguda en el model de voxels , bé per la voxelització, que és una discretització de la superfície. Per evitar aquests inconvenients, caldria investigar models mixtes que representessin simultàniament les dades volumètriques i els sòlids. Aquesta línia és paral.lela a la descrita a l'apartat 3.4.

Un segon enfoc és implementar operacions que puguin treballar sobre dos models diferents (voxels i B-Rep). En aquest sentit, sols s'han desenvolupat algorismes de visualització per escenes mixtes [GMO89] [Lev90] [Fru91] [MiK92] [KYC90] (veure apartat 5.4). Caldria, per tant, estudiar altres operacions (com seccions, càlculs volumètrics,...) que treballessin amb ambdós models.

3.6.- Conclusions

El problema de modelatge de volum està poc formalitzat; no existeix una definició matemàtica precisa d'aquests tipus d'objectes, ni s'han analitzat les especificacions dels sistemes de modelatge de volum.

Si bé el model de voxel és el més utilitzat per la seva simplicitat i la seva versatilitat, això no obstant, presenta l'inconvenient de la seva elevada ocupació de memòria. Caldria trobar models alternatius que permetin de compactar la informació. Els treballs realitzats dins d'aquesta línia estan orientats cap a l'analisi de l'utilització d'octrees i cap a l'estudi de la viabilitat de l'aplicació de tècniques de filtratge.

Una altra línia de recerca és el desenvolupament de models híbrids volum-superfície necessaris en certes aplicacions, com per exemple, el modelatge de jaciments de petroli. També s'investiguen els sistemes de modelatge mixtes que suporten simultàniament models de volum i models de sòlid. Com es pot veure en l'apartat 5.3, existeixen tècniques de visualització mixtes [KYC90][Lev90b] però encara no s'han estudiat altres operacions tals com seccions, interseccions, detecció d'interferències i operacions booleanes.

4 Conversions entre models

En aquest apartat es diferencien dos tipus de conversions entre models: la conversió del model volumètric a altres models (reconstrucció) i la conversió de qualsevol model a un model volumètric *(3D scan conversion)*.

4.1.- Reconstrucció

La reconstrucció consisteix en el pas d'un model volumètric a un altre model. Aquest procés s'utilitza, generalment, per reduir l'espai ocupat per les dades volumètriques i per a tractar amb més detall les características de les dades.

El model volumètric es pot convertir en un model de dades 0D (punts), en un model de dades 1D (contorns) i en un model de dades 2D (superfícies) (veure figura 4.1)

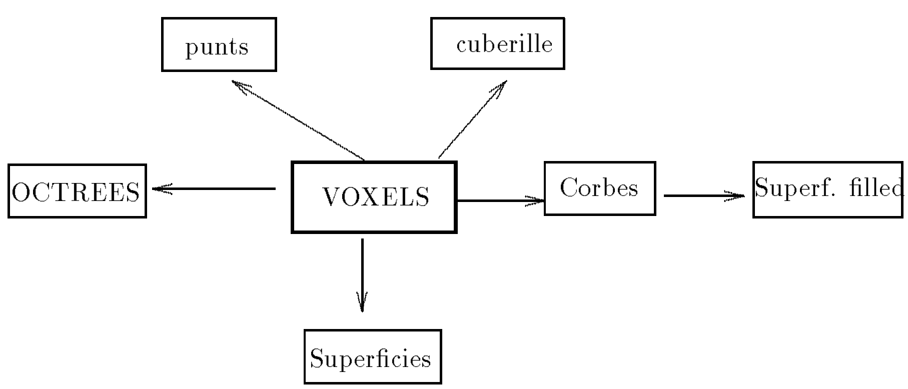

Figura 4.1 *Tipus de reconstruccions de voxels*

Reconstrucció de punts (0D)

El model volumètric es recorre directament i es considera cada voxel com un punt o com un conjunt de punts (els vuit vèrtexs del cub). S'associa a cada punt la informació, modificada o no, associada al voxel del què prové.

Reconstrucció de contorns (1D)

Per fer l'extracció de contorns a partir d'un model volumètric, cal que el volum estigui dividit en *serial sections* per extreure el contorn de cadascuna les seccions o llesques. És a dir, les dades d'entrada a la reconstrucció són un conjunt de llesques, i es dóna com a sortida un conjunt de contorns. Un contorn consisteix en la seqüència de punts que limiten el volum connectats per segments de línia, o per B-splines uniformes [Sung78].

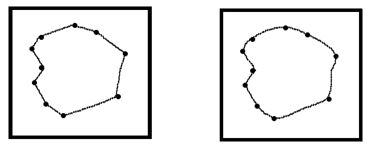

Figura 4.2 *Contorn aproximat per rectes i per B-splines*

Per extreure els contorns de cadascuna de les llesques , s'utilitzen tècniques de seguiment del contorn. Aquest seguiment pot ser:

- Manual [MaH76]: l'usuari defineix els punts de la frontera del volum i delimita la seva frontera.
- Semiautomàtic [PFM86]: s'utilitzen algorismes de *backtracking* i quan l'algorisme no sap per on seguir qüestiona a l'usuari el camí correcte.
- Automàtic [FVL86]: l'algorisme demana un primer punt interior al contorn per a poder aplicar un algorisme de germe.

Si les propietats separades pel contorn són molt contrastades, s'utilitzen primer tècniques de *thresholding* per determinar l'àrea tancada pel contorn i després s'utilitza un mètode de seguiment del contorn automàtic. En canvi, si les propietats no són molt contrastades, és necessari recórrer a mètodes manuals, i per tant es necesita un editor per extreure interactivament els contorns.

Encara que el model de dades 1D permet rotacions ràpides, té l'inconvenient que es veuen els forats entre les llesques i les imatges de contorns de convolució són difícils d'interpretar inclús utilitzant colors i *depth shading*.

Reconstrucció de superfícies (2D)

Per tal de visualitzar i manipular les dades 3D, molts algorismes generen isosuperfícies a partir del model volumètric, ja siguin voxels, conjunts de llesques o octrees. Una isosuperfície és una superfície 3D tal que el seu volum té una propietat que cumpleix una certa condició. Per a trobar una isosuperfície en el volum de dades, s'han desenvolupat molts algorismes que segueixen fonalmentalment una de les dues estràtegies següents:

- Tècniques de *tiling:* Aquests tipus d'algorismes tenen com a precondició que el model volumètric pugui dividir-se en llesques. Els algorismes de *tiling* comencen trobant els contorns de cada llesca, ja sigui de forma manual, semiautomàtica o automàtica. En un segon pas, es suavitzen els contorns generats a cada llesca i la superfície es genera unint els contorns de llesques consecutives mitjançant *patches* o *ribbons* ("enrejolat"). Les primitives o *patchs* usades en l'enrejolat poden ser triangles, rectangles i, en el cas de tenir contorns definit per B-splines uniformes, splines cardinals [Sung78] o splines [LCC89].

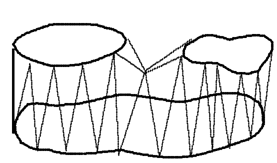

Figura 4.3 *Problema de branching*

Hi ha tècniques de *tiling* que optimitzen algunes característiques de la superfície generada. Keppel [Kep75] maximitza el volum tancat per la superfície, en canvi Fuchs [FKU77] minimitza l'àrea de la superfície.

Un problema inherent a la construcció per *tiling* consisteix en l'enrejolat de les bifurcacions o *branching*. Aquest problema vé donat quan un contorn d'una llesca es divideix en dos o més contorns en una llesca adjacent (veure figura 4.4). Es pot solucionar de forma manual [ChS78] o de forma automàtica [GaD82][NHK85][Boi85][AOU87].

En treballar amb superfícies "enrejolades", s'aconsegueixen visualitzacions realistes amb color, diferents fonts de llum i reflexió especular, es poden usar talls o transparències per veure l'interior del volum i es redueix el volum de les dades actives. En canvi, la detecció del contorn és difícil en estructures complexes i la precisió de la superfície depén de l'exactitut dels contorns. En general, és un procés costós en temps i no és gaire detallista.

Les àrees d'aplicació de les tècniques de *tiling* són la teràpia de radiació, el disseny d'implants ortopèdics.

- Generació de superfícies: A partir d'un model volumètric de voxels o octrees, s'han implementat diferents algorismes per a reconstruir superfícies 3D directament, sense reseguir el contorn de cada llesca.

Es poden aplicar els mètodes convencionals de triangularitzar un conjunt aleatori de punts (corresponents a voxels) en l'espai [BoF81][O'Ro81], però s'han desenvolupat tècniques més específiques i més adients a les dades volumètriques. Entre aquestes tècniques, hi ha les que segueixen un algorisme de germe (a partir d'un punt de la frontera de superfície i la manera de trobar el següent punt de la frontera, es construeix la isosuperfície) [Art79][HeL79], i hi ha les tècniques que aproximen la superfície localment en cada unitat volumètrica [LoCl87].

En aplicar un algorisme de germe, Artzy [Art79] modela la superfície d'un objecte com un graf dirigit i redueix el problema de la detecció de la superfície a un problema de recorregut d'un arbre. Es defineix un digraf G, tal que els seus nodes corresponen a les cares frontera dels voxels de la superfície i els subgrafs de G connexes corresponen a les diferents superfícies connexes de l'escena. Cada node del graf està connectat amb els nodes que corresponen a cares adjacents a ell, de forma que a cada node entren dos arcs i en surten dos. Així, per a tot node, existeix un arbre binari arrelat al node, tal que l'arbre s'extén a totes les components connexes del node arrel. La superfície es genera a partir d'un node del graf i el recorregut de l'arbre arrelat al node.

Per obtenir un model de *cuberille*, es recorre el model discriminant quins voxels són frontera de la superfície dels que no ho són. Hi ha la possiblitat d'eliminar de cada voxel frontera, les cares que són internes a la superfície. Així, s'obté una superfície formada per les cares límit dels voxels frontera. En el cas de no eliminar aquestes cares, s'obté una superfíce "gruixuda" formada per un conjunt de voxels pels què passa la frontera. Concretament, Herman i Liu [HeL79] van desenvolupar un algorisme per a generar un model *cuberille* format per un conjunt de voxels, seguint un procés de germe a partir d'un punt entrat per l'usuari. A partir d'un voxel frontera troba els voxels veïns que són possibles candidats a formar part de la frontera i escull d'entre ells el que té màxim gradient. Si l'esculllit no és de la frontera, s'inicia un procés de *backtacking* per anar a buscar un altre candidat. Per una altra banda, s'han desenvolupat algorismes per a obtenir el model de *cuberille* mijançant una tècnica de recorregut del model i, a nivell local de cada node, es determina si és de la superfície o no.

De forma local a cada unitat volumètrica, *Marching Cubs* [LoC87] crea una representació poligonal d'isosuperfícies a partir d'un array 3D de dades. Aquest algorisme utilitza com entrada un array de *slices* 2D, encara que es pot simular aquest array a partir d'un conjunt de voxels. Mitjançant l'estratègia de *divide-and-conquer*, l'algorisme crea triangles per aproximar la superfície en un cub format per vuit pixels (quatre d'una llesca i quatre de la llesca adjacent). Per assegurar una bona qualitat d'imatge, es calculen les normals de la superfície calculant les normals a cada vèrtex de cada triangle. L'algorisme segueix els següents passos:

1.- Es llegeix com a dades d'entrada quatre llesques adjacents.

2.- Es construeix un cub a partir de quatre pixels de dues llesques adjacents (un cub pot tenir diferents escales a cada dimensió).

3.- Es fa el test d'intersecció entre la isosuperfície i el cub, classificant cada vèrtex del cub respecte a la superfície.

4.- Amb el resultat del test anterior, es construeix un índex, anomenat també configuració del cub (veure figura 4.4), que té un rang de 0 a 255. Aquests 256 casos es poden reduir, segons [LoC87] a 15 cassos, per les simetries i complementacions dels diferents talls en el cub.

5.- Mitjançant l'índex, s'accedeix a una taula, tal que per a cada possible configuració del cub, indica quines arestes del cub són tallades per la superfície. Al llarg de cada aresta del cub en la què hi hagi un canvi de signe, s'interpola linealment el punt de tall de la superfície, usant les densitats dels vèrtexs.

6.- Es calcula la triangularització dels punts de tall (tan sols és unir els punts de tall formant triangles). Un cop es tenen els triangles se'ls hi aplica els 24 coeficients corresponents a les tres matrius del *pipeline* de visualització i les tres matrius de rotació (segons la configuració del cub) i complementant (segons quins vèrtexs del cub són dins i quins són fora) per arribar a una de les 256 possibles configuracions.

7.- Es calcula la normal de la superfície a cada vèrtex de cada triangle.

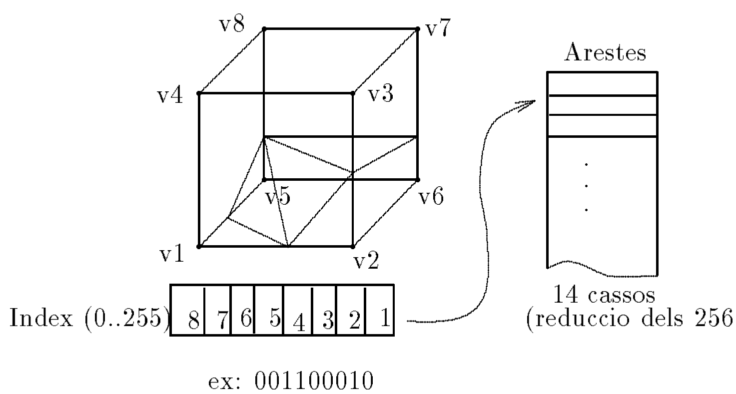

Figura 4.4 *Configuració d'un cub en l'algorisme de Marching Cubes*

Com a sortida d'aquest algorisme, es té un conjunt de triangles, amb una normal associada a cada vèrtex de cada triangle.

Per a evitar càlculs innecessaris en l'exploració d'un cub, sols s'exploren tres noves arestes a cada cub, ja que excepte els cubs del marge, totes les altres interseccions amb les altres arestes ja han estat prèviament calculades en els cubs veïns.

L'algorisme de *Marching Cubes* té ambigüitats en alguns casos en els que la configuració dels vèrtexs del cub respecte la superfície no és suficient per determinar les arestes que intersecciona la superfície [Dur88]. Aquestes ambigüitats (veure figura 4.5) han estat solucionades [Whi90], revisant el nombre d'aproximacions.

Aquest algorisme s'ha adaptat també per a xarxes estructurades [ShN89] i per a xarxes no estructurades [MHC90][RVB90].El mètode de *Marching Cubes* també s'ha implementat per a particions de l'espai en forma de tetraedres *(Marching Tetrahedra)* [Ups86].

L'algorisme de *Dividing Cubes* [CLL88] és molt similar a *Marching Cubes* però la seva sortida és diferent per fer més eficient la rasterització. Després de fer el test per saber si la superfície intersecta amb el cub, es subdivideix el cub, en cubs més petits. Cada subcub es reclassifica de nou respecte la superfície i tots el que són intersectats es tornen a subdividir, o bé es donen com a sortida. La quantitat de subdivisions depén ded la resolució desitjada. Un cop es tenen els cubs a la resolució desitjada, es calcula la normal a la superfície en el centre de cada cub.

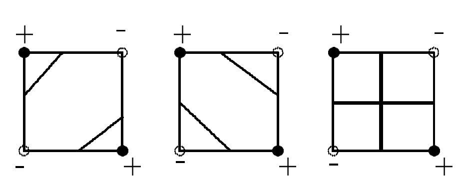

Figura 4.5 *Cassos ambigus en l'algorisme de Marching Cubes*

Tant el mètode de *Marching Cubs* com el mètode de *Dividing Cubes*, per aproximar la normal a la superfície, és a dir, per aproximar el gradient $g = (g_x, g_y, g_z)$ en un punt (x_0, y_0, z_0) amb densitat $f(x_0, y_0, z_0)$, utilitzen la diferència central al llarg dels tres eixos coordenats:

$$g_x = \frac{f(x_0 + a, y_0, z_0) - f(x_0 - a, y_0, z_0)}{2a}$$
$$g_y = \frac{f(x_0, y_0 + b, z_0) - f(x_0, y_0 - b, z_0)}{2b}$$
$$g_z = \frac{f(x_0, y_0, z_0 + c) - f(x_0, y_0, z_0 - c)}{2c}$$

on (a, b, c) són les dimensions del voxel. Degut a la utilització d'aquestes fòrmules, s'han de llegir quatre llesques i no sols dues, per a poder determinar la superfície.

Un altre mètode que calcula la superfície basant-se en les densitats és la interpolació local presentada per Webber [Web90]. A cada voxel s'aproxima una superfície biquadràtica utilitzant els 3x3x3 veïns. És un mètode costós però és molt precís en l'aproximació de la superfície en objectes biquadràtics.

Per a generar isosuperfícies a partir d' *octrees*, Wilhems implementa dos tipus d'algorismes:

- Una adaptació del mètode de *Marching Cubes* a *octrees* en la que es fa un recorregut de l'arbre i l'aproximació local de la superfície. Aquesta adaptació no permet aprofitar les interseccions prèviament calculades.
- Un mètode de construir i recórrer l'*octree* per a poder reduir el nombre de cel.les buides explorades i poder aprofitar l'informació ja calculada. Aquest mètode crea en una fase inicial un *octree* tal que cada node intermig conté el valor mínim i màxim contingut en el subarbre que està arrelat en ell. Les fulles de l'arbre contenen directament vuit cel.les i un punter a la taula de dades, al valor mínim dels 27 valors definits per les vuit cel.les. El volum s'enmagatzamena en una taula 3D per simplificar la localització de veïns. Per reconstruir la isosuperfície, s'atravessa l'*octree* amb el valor de *threshold*, explorant només les branques de l'arbre que contenen part de la superfície. Quan es visita una fulla, s'explora cadascuna de les cel.les i es generen els polígons corresponents.

Tots dos mètodes usen una taula de *look-up* per generar els polígons: la taula conté 256 entrades (una per a cada possible configuració) i cada entrada descriu quines arestes de la cel.la conté interseccions i com s'han d'unir els polígons. S'utilitza una segona taula per a tractar els cassos ambigus, ja esmentats en la descripció de *Marching Cubes*. Cada cas ambigu de la primera taula conté un punter a la secció corresponent a la segona taula.

S'ha comprovat que aquests dos mètodes acceleren la formació de la superfície degut que es redueixen el nombre de cubs a explorar gràcies a l'estructura d'octrees.

4.2.- Voxelització

En aquesta secció es decriu la conversió de qualsevol model a voxels o a octrees. El model a convertir pot ser un conjunt de polígons, rectes, corbes parametritzades, superfícies, etc.

Cal destacar que tots els algorismes de *scan-conversion* per a obtenir un model d'octrees són aplicables per obtenir un model de voxels, i a l'inrevés, ja que sols canvia el nivell de subdivisió en el que s'arriba. En alguns algorismes d'octrees és inevitable passar per un models intermitjos que en el cas de voxels són innecessaris. Les diferències principals es poden observar en les figures 4.6 i 4.7.

El tipus de les dades d'entrada determinen el tipus d'algorisme de construcció. Les dades d'entrada poden ser:

- Seccions (quadtrees): A partir d'una sèrie de seccions d'un objecte 3D es defineix l'octree. Aquestes seccions poden transformar-se primer a quadtrees i després es realitza el pas de quadtrees a octrees [YaS83]. En el cas de construir un model de voxels, no és necessari passar pel model de quadtrees.
- Siluetes: El model volumètric es reconstrueix a partir de les imatges del objecte 3D projectades [Pot85]. Aquesta projecció és l'ombra de l'objecte en ser il.luminat des d'un cert punt. Es construeix l'octree de

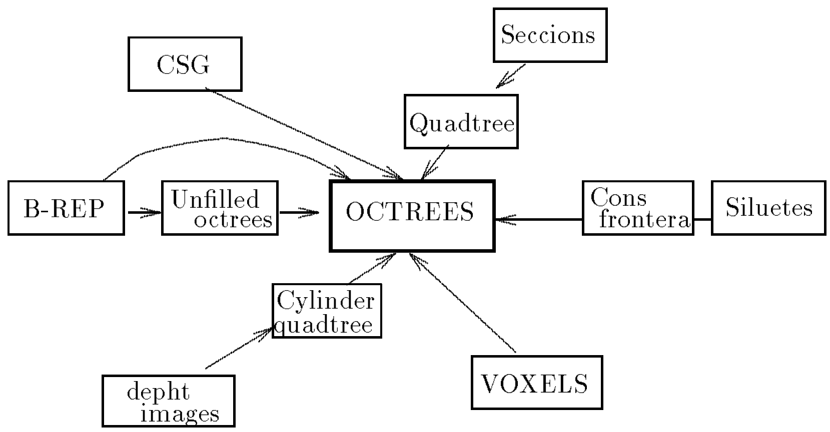

Figura 4.6 *Tipus de construccions d'octrees*

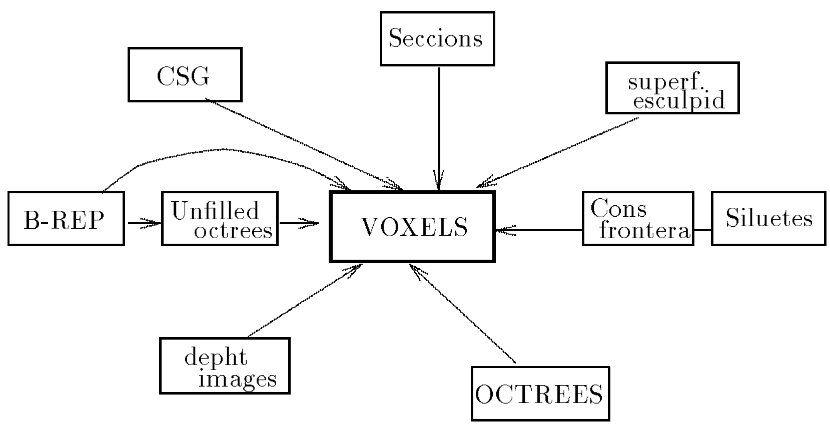

Figura 4.7 *Tipus de construccions de voxels*

cadascun dels cons de projecció. S'interseccionen tots els octrees formats a partir de les diferents vistes per formar el model final.

- Voxels/Octrees: Es pensa en l'octree com una compactació de la informació guardada en el model de voxels. El pas d'octrees a voxels tan sols és subdividir l'octree fins el nivell d'un voxel.
- Imatges en profunditat: Una imatge en profunditat és un conjunt de pixels tal que cada pixel té associada la profunditat del punt de l'objecte que representa. Es té una imatge en profunditat per cada vista de l'objecte. A partir d'un conjunt de vistes es pot determinar quin és el model volumètric final [Con85].
- Model fronteres: un sòlid representat en model fronteres es descomposa recursivament. L'algorisme desenvolupat per realitzar el *scan-conversion* es basa en la divisió successiva de l'espai i la repartició de la informació de la frontera en els cubs successius [BrN90]. Hi ha una variació d'aquest algorisme que passa per un model intermig d'octrees, anomenat *unfilled octree* on sols estan definits els nodes grissos mínims. A partir d'aquesta informació i mitjançant un algorisme de germe es determinen les propietats de cada node.
- Model CSG: Un arbre CSG es converteix a voxels (octree), transformant cada primitiva de l'arbre CSG a voxels (octree) i avaluant cada operació de l'arbre CSG com una operació entre voxels (octrees) [LeR82].
- Superfícies esculpides: Una superfície parametritzada es pot descriure com una funció d'un, dos o tres paràmetres, que varien entre 0 i 1. Per a definir el model volumètric es pot avaluar directament punts de la superfície a cada punt informatiu del voxel. Aquest mètode és poc eficient i lent ja que, inclús usant la regla de Horner, l'avaluació a cada punt és costosa. Un altre mètode consisteix en biseccionar repetidament la superfície fins que el segments s'incloguin dins d'un voxel. Però, per fer un

algorisme realment eficient, Kaufman utilitza els paràmetres amb diferents passos d'epsilon. En [Kau87] es descriuen mètodes per a convertir corbes, superfícies i volums paramètrics a un model de voxels, en concret el model de voxels definit en el Cub (veure apartat 7.2).

5 Visualització

5.1.- Introducció

La visualització és el procés que permet de donar una representació gràfica del model computacional. És especialment important en el camp del modelatge de volum per la seva utilitat per a l'exploració de dades i la simulació.

El procés de visualització es composa de tres parts principals:

– Transformacions geomètriques, escalat i retallats

– Eliminació de Parts Amagades (EPA)

– Càlcul de la intensitat a assignar a cada element visible (Shading)

Les tècniques emprades per a dur a terme aquests tres processos depenen esencialment de la representació geomètrica escollida per modelar les dades.

Les tècniques de visualització de sòlids rígids i de superfícies esculpides no són adequades per els models de volum, encara que, com es veurà més endavant, algunes d'elles s'hi poden adaptar. Els apartats següents es centren en la visualització d'aquests models: en primer lloc (apartat 5.2) es descriuen aspectes generals d'aquest procés, a continuació, s'analitzen més detalladament les tècniques d'EPA (apartat 5.3) i de Shading (apartat 5.4).

5.2.- Tècniques de visualització dels models de volum.

Com s'ha esmentat al punt 2, cada unitat de volum té associada una informació que pot ser **monovaluada** (un sol paràmetre) o **multivaluada** (més d'un paràmetre) [Kau90]. La representació d'un paràmetre consisteix en la visualització de les unitats volumètriques que cumpleixen una de les següents condicions:

– Tenir associat un valor o un conjunt de valors determinats del paràmetre.

– Tenir associat un valor comprès en un rang o conjunt de rangs fixats.

– Cap condició. Per tal de mostrar l'evolució espaial del paràmetre dins del volum, es visualitzen totes les unitats de volum. Aquesta evolució pot ser de tot el volum, d'un subvolum o bé d'una secció del volum.

També es pot voler visualitzar més d'un paràmetre simultàniament.

La figura 5.1il.lustra el procés de tractament del model per a la seva visualització. Com es pot veure, pot ser necessari aplicar un preprocès de filtratge del model, el resultat del qual és que cada unitat de volum té com a valor del paràmetre analitzat bé un valor **binari**, bé un valor **comprès entre 0 i 1** [Lev91].

El procés de visualització consta de sis etapes, tot i que sovint algunes d'aquestes etapes són omeses, o realitzades conjuntament, o bé en diferent ordre. En primer lloc, es realitza l'extracció, explícita o no, de les unitats volumètriques que compleixen la condició seleccionada. Aquesta extracció es fa habitualment aplicant filtres *(thresholding)*.

Després, es calculen unes entitats geomètriques representatives (punts, rectes, o polígons) incloses en les unitats volumètriques seleccionades. Per tant, la següent etapa rep com entrada el que anomenarem **model de visualització** i que pot estar constituït per:

– 0D (punts)

– 1D (rectes o corbes)

– 2D (superfícies o plans)

– 3D (unitats de volum)

Aquesta tercera etapa consisteix en l'aplicació de transformacions geomètriques (traslació, rotació, escalat, ...) per tal d'expressar el model de visualització en el sistema de coordenades de l'observador. A continuació, es realitza l'eliminació de parts amagades i després els càlculs d'il.luminació i realisme *(shading)*. Aquests dos processos depenen fortament del tipus del model de visualització. L'última etapa consisteix en la rasterització.

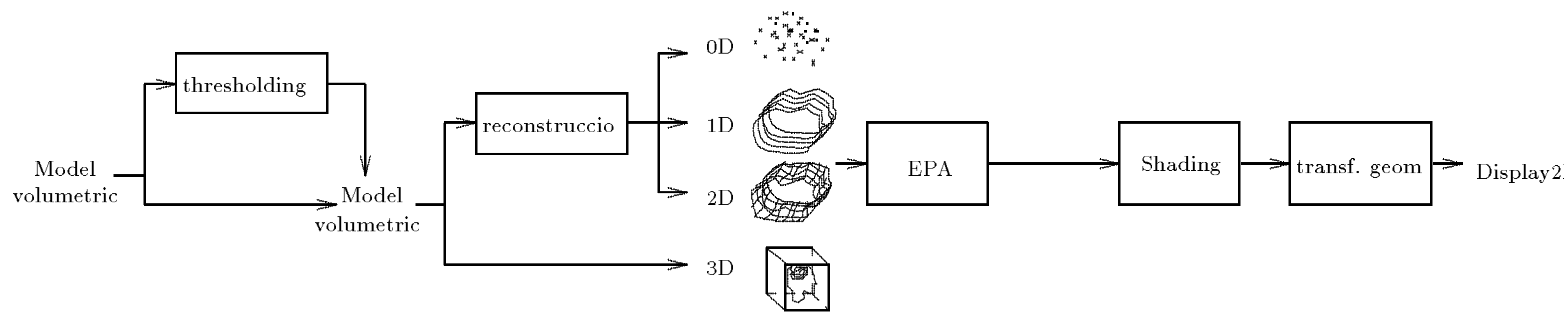

Figura 5.1 *Etapes del procés de visualització.*

Els algorismes de visualització es diferencien segons l'estratègia utilitzada en les diferents etapes del procés. Per a poder comparar els diferents algorismes de visualització, s'evaluen els següents paràmetres:

– el temps de càlcul emprat per a generar la imatge.

– la memòria ocupada.

– la versatilitat del mètode, és a dir, la facilitat amb la què es poden modificar paràmetres d'entrada.

– la precisió amb la què s'adequa el model de visualització al model de volum.

– la qualitat de l'imatge.

– la possibilitat de paral.lelitzar el procés.

5.3.- Tècniques d'eliminació de parts amagades

L'eliminació de parts amagades (EPA) consisteix en, donat un punt de vista, determinar la visibilitat dels objectes de l'escena en la imatge final. Això implica l'ordenació en profunditat, sigui explícita o no, dels objectes del model per determinar les parts visibles en la imatge final.

Els mètodes d'Eliminació de Parts Amagades per als sòlids representats per la seva frontera formada per cares planes es classifiquen, segons [SSS74], en funció de l'espai en el què treballen:

– Algorismes d'espai objecte: com els de Appel [App67], Galimberti i Montanari [GaM69].

– Algorismes d'espai imatge: entre els que destaquen el de Ray-Casting, els basats en el principi d'escombrat línia a línia, el de Z-buffer.

– Algorismes mixtes, que treballen parcialment en espai objecte i en espai imatge: en particular el de BSP [FAG83].

Per a visualitzar les superfícies esculpides es pot aproximar-les primer per cares planes i aplicar després un dels mètodes citats anteriorment. Existeixen a més, algorismes específics per visualitzar quàdriques [Mah72] i bicúbiques [LCW80] segons el principi d'escombrat línia a línia, així com generalitzacions de l'algorisme de Ray-Tracing.

Com s'ha esmentat en l'apartat 5.2, la dimensió de les dades d'entrada dels algorismes d'EPA condiciona l'estratègia emprada pels algorismes per definir les parts visibles. Les tècniques d'EPA en escenes de sòlids rígids i de superfícies esculpides es podran aplicar si el model de visualització és bidimensional i algunes d'elles es podran adaptar en el model tridimensional.

D'altra banda, hi ha models de visualització que porten implícita l'ordenació espaial de les unitats de volum, com és el cas dels models 3D i dels models 2D d'octree i cuberille. Aquesta ordenació permet l'aplicació d'algorismes que recorrin el model *back-to-front* (BTF) o *front-to-back* (FTB). En projeccions paral.leles l'ordre del recorregut del model és el mateix per a FTB com a BTF, però en una projecció perspectiva aquest ordre pot canviar.

Els algorismes trobats en la bibliografia que tracten dades 0D (punts o partícules) consideren que tot punt és visible, és a dir, no utilitzen cap tècnica d'EPA, i per donar certa visió espaial recorren a l'ús d'un *shading* adient al tipus de dades [Bli82],[Ree83] (veure apartat 5.4).

Les dades 1D d'entrada als algorismes d'EPA acostumen a ser un conjunt de contorns que representen la frontera de cada *slice*. Aquesta frontera es representa per un conjunt de punts connectats per rectes o per corbes (Foto 3). A partir de la informació d'aquesta frontera, es pot fer directament una eliminació de parts amagades (descrita en el programa TROTS [VeP77]). El model de visualització unidimensional també pot ser un conjunt de contorns en els tres plans ortogonals xy, yz, xz, tals que es combinen per a generar la imatge (implementat en el programa ISOSRF [WrH79]).

Per a realitzar l'EPA en models de visualització bidimensionals (plans o superfícies), es poden aplicar els algorismes convencionals per a sòlids representats per la seva frontera formada per cares planes o per a superfícies esculpides, tal com s'ha esmentat anteriorment.

Concretament, pel model *cuberille* s'han implementat dos tipus d' algorismes:

- Recorregut FTB o BTF del *cuberille* el.liminant per *culling* les cares no visibles dels voxels frontera (Foto 4 i Foto 5).
- *Z-Buffer: (o brute force)* [HeL79] que no té en compte l'ordre de recorregut. A cada pixel, s'associa una profunditat i una brillantor que, inicialment, seran infinit i zero respectivament. De cada voxel es projecten les seves cares visibles (és a dir, les que no s'eliminen per *culling*). Cada cara té associades la seva profunditat i la seva brillantor i, per a cada pixel en el què es projecta la cara, es compara la distància del pixel i la de la cara. En cas que la distància del pixel sigui més gran, se li assigna com a nova distància i com a nova brillantor, les de la cara.

S'han implementat algorismes sobre *octrees* que recorren l'arbre BTF en un determinat ordre segons la posició de l'observador que permeten aplicar directament l'algorisme del Pintor [Mea82][GWW86]. Per recorreguts FTB és necessari l'ús d'informació adicional. Una tècnica per disminuir la memòria auxiliar utilitzada en el cas de projeccions isomètriques és la utilitzada als G-Octrees [MFK90].

La descomposició espaial dels models volumètrics porta implícita l'ordenació espaial de les unitats de volum segons la posició de l'observador. Aquesta ordenació permet l'aplicació d'algorismes que recorrin el model *back-to-front* (BTF) o *front-to-back* (FTB). En projeccions paral.leles l'ordre del recorregut del model és el mateix per a FTB com a BTF, però en una projecció perspectiva aquest ordre pot canviar.

Els mètodes d'EPA en models de visualització 3D són modificacions de mètodes convencionals d'escenes on els sòlids es representen per a la seva frontera. Concretament, en la bibliografia trobada referent a la visualització de volum, els algorismes d'eliminació de parts amagades són espai imatge i per a models de voxels.

Per una posició de l'observador donada, cal determinar tots els objectes, o parts dels objectes, visibles. Els objectes són opacs, translúcids o transparents, ja sigui pel seu comportament amb la llum o bé per decisió de l'usuari. Per tant, la determinació de la visibilitat dependrà dels tipus d'objectes del model volumètric. Si hi ha objectes o parts de superfícies que són transparents, el volum intern és visible, mentre que si la frontera és opaca, sols serà visible la frontera i no es visualitzarà el volum. En aquest sentit, quan la frontera és opaca, es pot considerar com un mètode 2D en quan es projecten les cares d'incidència.

Els algorismes per a determinar la visibilitat i eliminar les parts amagades segueixen tres classes d'estratègies, segons la dimensió de la finestra de pantalla des d'on s'analitza l'escena (0D si és a nivell de pixel, 1D a nivell de scan-line, 2D a nivell de pantalla). Segons la bibliografia trobada, en l'àmbit de visualització de volum s'han desenvolupat algorismes de 0D, també anomenats de *image order* i algorismes de 2D, anomenats de *object order*. Les tècniques de *image order* determinen, per a cada pixel de la imatge, quins objectes o unitats del model es projecten. Les tècniques de *object order* recorren el model 3D i el projecten en la imatge plana, segons la posició de l'observador.

Els mètodes de *object order* comporten evitar el *aliasing* inherent a tota projecció de punts individuals, ja que una unitat de volum, per coherència del model, es projecta en pixels veïns.

5.3.1.- Mètodes de object-order

Els mètodes de *object order*, anomenats també de projecció o de *Z-buffer*, projecten cada unitat de volum en la imatge 2D, analitzant quins pixels afecta i quina contribució té a l'opacitat de cada pixel (Foto 7).

Segons com es travessi el model, els algorismes es classifiquen en dues grans classes:

- *back-to-front* : El model es recorre des de darrera cap endavant pintant primer el què està més lluny de l'observador i desprès el què està més aprop, és a dir, són versions volumètriques de l'algorisme del pintor ([CHR85], [Far83], [Wes90], [Mea82], [WiG91], [FGR85], [GWW86], [LaH91]). Aquest tipus de recorregut implica la generació de la imatge per capes successives que permeten donar informació a l'usuari de parts dels volum que en la imatge final quedaran amagades.
- *front-to-back* : El model es travessa de davant cap endarrera i un pixel es pinta només en el cas que no hagi estat pintat amb anterioritat, o bé no hagi arribat al seu grau màxim d'opacitat [UpK88], [HBP90a]. Per tant, es necessita associar a cada pixel una profunditat i una opacitat. No és necessari recórrer tot el model degut a que la condició de fi és que l'opacitat de tots els pixels sigui màxima. Aquest tipus de recorregut esdevé molt ràpid quan es visualitzen objectes opacs.

Les projeccions paral.leles tenen l'avantatge de conservar la coherència entre cel.les de forma que totes les cel.les projectades tenen la mateixa forma geomètrica. Wilhems proposa usar la coherència dels volums regulars i la projecció paral.lela per accelerar el procés de visualització [WiG91]. La projecció de cada cel.la és la mateixa per a totes les cel.les segons un mateix punt de vista, però trasladada a diferents llocs de pantalla. Per tant, es pot fer un preprocès que determini, segons la direcció de l'observador, la forma de la cel.la projectada i quan es projecti cada cel.la, usar la geometria precalculada amb una certa traslació segons l'ocupació espaial de la cel.la. Si la projecció fós perspectiva, en canvi, l'algorisme es complica considerablement.

Per tal de donar una aproximació de la imatge s'utilitza la tècnica de *progressive refinement* [LaH91] que consisteix en tenir diferents nivells de precisió en el model, o bé que l'algorisme de visualització accedeixi al model a diferents nivells de precisió. Així, a baix nivell de precisió es poden generar imatges interactivament i, en un cert moment, es poden produir successivament imatges més precises des d'un punt de vista donat, per exemple, quan l'usuari volgui examinar amb més cura una imatge que l'interessi.

Com a conclusió, els mètodes d'*object order* ocupen poca memòria ja que sols necessiten l'unitat de volum tractada a cada moment i els pixels en els què es projecta (que com a màxim serà el tamany de la pantalla). El fet de treballar localment amb una unitat de volum fa que siguin mètodes fàcilment paral.lelitzables en màquines de processadors amb poca memòria local (veure apartat 8).

5.3.2.- Mètodes de image-order

En els algorismes de *ray-casting*, es llenca un raig des de la posició de l'observador cap a cadascun dels pixels de la imatge. Per a cada pixel s'exploren quins voxels són atravessats pel raig, calculant quina opacitat aporten al pixel (veure apartat 5.4).

Hi ha diferents algorismes que utilitzen la tècnica de ray-casting però es diferencien esencialment en la forma en què fan el *shading* ([TuT88], [Lev88], [MCP91], [Sab88], [DCH88], [UpK88], [Lev90a]) i es veu detalladament en el següent apartat (Foto 14-15).

L'ordre de recorregut del model pot ser FTB o BTF, encara s'utilitza més freqüentment el recorregut FTB degut a que el nombre de dades a explorar és menor. En recórrer un model FTB, es pot determinar el fi de l'exploració del raig quan l'opacitat acumulada en aquell pixel és 1 , ja que això indica que el raig ha intersectat un objecte opac, o bé ha recorregut prou distància a través d'un objecte semi-transparent.

Per tal d'accelerar aquests tipus d'algorismes, s'aprofita la coherència espaial de les regions amb voxels buits [Lev90a], codificant-ho mitjançant l'enumeració espaial amb octrees [Mea82].

Una altre estratègia per a acelerar el procés de ray-casting en projeccions paral.leles és l'emprada per Upson i Keeler [UpK88] en la què segons la pendent del raig es determina quines tres de les sis cares del cub penetre el raig. La probabilitat que un raig interseccioni una cara donada perpendicular a un eix de coordenades és inversament proporcional a la pendent del raig respecte l'eix. Així, el test de penetració del raig pot ser minimitzat. Les interseccions del raig amb el conjunt de plans paral.lels poden ser generades en tres llistes diferents i fusionar-les al final del càlcul. De tal manera, que la generació d'interseccions és ràpida però la fusió és lenta. Aquest mètode genera totes les interseccions al llarg del raig quan pot ser que no es necessitin totes.

Les tècniques de *image order* són mètodes eficients per visualitzar escenes amb objectes translúcids o transparents. La seva vectorització és difícil degut a que requereixen memòria global compartida o bé tenir replicat el model a cadascun dels processadors (veure apartat 8). Un problema adicional és l'aparició de *aliasing* en les imatges generades per aquests mètodes que fa necessari un postprocés per a solucionar-lo.

5.4.- Shading

El shading és el cinquè pas del procés de visualització descrit a la figura 5.1 i consisteix en el càlcul de la intensitat a assignar als elements visibles. Com la eliminació de parts amagades, el shading depén fortament del esquema de representació de les dades. Per les escenes formades per superfícies esculpides o per sòlids homogenis representats per la seva superfície, els algorismes de shading es basen en models d'il.luminació que pretenen simular amb el més gran realisme possible, la intensitat en cada punt tal i com és percebuda per l'observador. D'accord amb Hall [Hall90], aquests models es poden classificar en tres grups:

- Empíric [Gou71] [Pho75], que modela tres il.luminacions, ambient, difusa i especular en les superfícies, basant-se, com el seu nom indica, en equacions empíriques, i sense tenir en compte les interreflexions entre superfícies.
- Transicional [Whi80], en el que es basa l'algorisme de Ray Tracing, i que consisteix en el càlcul de l'arbre de reflexions i refraccions especulars de la llum en la seva trajectòria des de que és emesa fins que és percebuda per l'observador.
- Analític [GTC84], en el què es basen els algorismes de radiositat, i que fa un balanç global de l'energia emesa i percebuda en totes les superfícies d'un entorn tancat.

La problemàtica és diferent en el shading de models de volum o **discrete shading**. L'objectiu del shading ja no és aconseguir *realisme* en la percepció de les escenes, sinó facilitar la comprensió de les dades, resaltant-ne determinats aspectes, afavorint la percepció de la profunditat, o marcant contrastes entre determinades propietats [FAO91]. El color, l'opacitat, els coeficients de reflexió i refracció que, en els models de sòlids i de superfície, són caracterísques *objectives* intrínseques dels objectes modelats, són en canvi, en els models de volum, paràmetres *subjectius* de la visualització, que cada algorisme relaciona de manera diferent, sovint a criteri dels usuaris, amb els valors associats a les unitats volumètriques.

Existeixen nombroses taxonomies dels algorismes de shading. Els classificarem aquí en funció de les dades d'entrada que reben després de l'eliminació de parts amagades i esmentarem d'altres criteris en la taula de resum 5.1.

1. Punts:

Els models d'il.luminació per a punts es basen en la teoria física del transport d'ones *(Radiative Transport Theory)* [Cha60]. L'equació de la dispersió de la llum es deriva d'un balanç d'energia entre els extrems d'un cilindre diferencial. L'equació estableix que la diferència d'intensitat entre els dos extrems del cilindre (primer terme de l'equació 5.1) és igual a la diferència entre l'energia emesa per el cilindre (tercer terme) i la que aquest absorbeix (segon terme):

$$\frac{-1}{(k\rho)} s \nabla_x I(x,s) - I(x,s) + \frac{1}{4\Pi} \int_{\|\bar{s}\|=1} p(s,\bar{s}) I(x,\bar{s}) d\bar{s} = 0 \tag{5.1}$$

En realitat, l'energia emesa es composa de fet de dos termes: el de la dispersió de la llum i el atribuïble a que les partícules poden ser elles mateixes fonts de llum. La formulació de l'equació 5.1 suposa però, que les partícules no són fonts.

Per a resoldre l' equació, Blinn [Bli82] fa les següents hipòtesis:

- una sola font de llum
- medi de densitat uniforme
- medi de baix albedo, és a dir que el ratio entre la llum reflectida pel medi i la llum incident és petit
- dispersió simple: no es consideren les reflexions d'ordre superior.

El model de Blinn, concebut inicialment per la visualització dels anells de Saturn representats per un conjunt de partícules de gel, es pot aplicar també a la il.luminació de núvols i de superfícies amb pols.

Kajiya i Von Herzen [KaV84] proposen un model d'il.luminació alternatiu amb dispersió multiple, per a medis amb alt albedo. Així mateix, presenten un algorisme de ray-tracing basat en aquests dos models i consistent en dues fases. En un pre-procés es calcula una matriu d'intensitats, o contribucions de la font de llum a cada punt del model. Després, s'integren al llarg de cada raig les brillantors dels punts, calculades bé segons el model de baix albedo, bé segons el d'alt albedo. En un article posterior [KaK89], Kajiya generalitza el seu model i l'aplica a la visualització de textures de pell i cabell.

Reeves [Ree85] proposa un model d'il.luminació molt simple per a escenes formades per partícules puntuals. Cada partícula, és considerada com un emissor pur d'una intensitat associada de forma estocàstica. No hi ha dispersió de la llum. El shading consisteix en un traçat de raigs. Per cada raig es sumen les intensitats de tots els punts (partícules) que aquest intersecta.

2. Corbes i segments de rectes

Quan el model de visualització està format per línies, segments de rectes o corbes, el shading es redueix al *depth cueing*, o atenuació amb la distància que permet de pintar els punts dels segments amb una intensitat inversament proporcional a la seva distància a la font de llum [VMW83].

3. Models cuberille, octree i de voxels opacs

L'objectiu de la visualització és, en aquest cas, la representació de la superfície d'un o més objectes. Cal distingir dos casos:

- quan la superfície passa per els voxels frontera però no coincideix amb les cares dels voxels.(figura 5.2(a))
- quan la superfície està formada per les cares visibles dels voxels visibles com en el cas del model cuberille.(figura 5.2(b))

El cas (a) correspon als mètodes de reconstrucció descrits a l'apartat 3 [LoC87], [Web90]. Un cop obtinguda la superfície el shading aplicat pot ser qualsevol dels mètodes per a superfícies citats més amunt. De fet, s'utilitza habitualment el model empíric amb suavitzat de Gouraud o Phong. El càlcul de la normal, tal i com s'ha indicat en l'apartat 3, depén del gradient d'aquest valor en el veïnatge del voxel. Això és degut a que els valors (o nivells de grisos) associats als voxels frontera representen aquí el tant per cent d'ocupació dels voxels i per tant són indicadors de la orientació de la superfície. Per això, aquest shading és sovint referenciat a la bibliografia com **Gray-Level Gradient Shading**. Quan la superfície es reconstrueix per

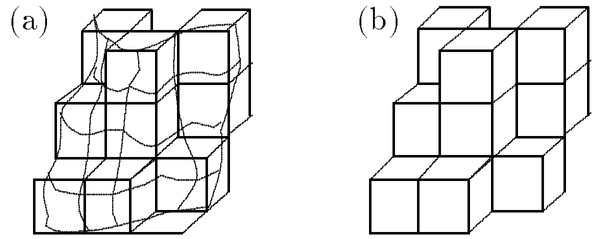

Figura 5.2 *Models de superfície*

derivació de trossos biquadràtics, com en l'algorisme de Webber, es referencien com **Local Interpolation Gradient** [Web90].

Quan la superfície està formada per les cares dels voxels visibles, el model d'il.luminació utilitzat és una adaptació del model empíric de Lambert que contempla només la reflexió difusa i el *depth cueing*. Segons Chen, Herman, Reynolds i Udupa [CHR85], la intensitat en un punt P de la superfície es calcula com:

$$I(P) = (I_{max} - I_{min}) * f_d * factor\_normal(P) + I_{min}$$

on,

- I(P) és la intensitat en el punt P expressada com una adressa entera [0-255];
- I_{max} i I_{min} els valors màxim i mínim de intensitat considerats, també enters i habitualment fixats a 235 i 30 respectivament ;
- f_d el factor d'atenuació, igual a $\frac{d}{2R}$, essent d la distància del punt a la llum i R el radi de l'esfera englobant a la superfície de l'objecte.
- factor_normal(P) és el terme que, en el model de Lambert per a superfícies, és igual al cosinus de l'angle que formen la normal a la superfície en P i la direcció d'observació (figura), i que, en el model *cuberille*, pot expressar-se de diverses maneres segons l'algorisme. El que diferencia cada mètode és precisament com realitzen aquest càlcul.

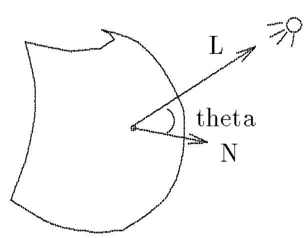

Figura 5.3 *Factor-normal en el model de Lambert*

Aquesta formulació, vàlida quan es representa la superfície d'un sol objecte, es pot generalitzar quan hi ha més d'un objecte, definint tants rangs d'intensitats (màxim i mínim) com nombre de superfícies hi ha. L'especificació dels rangs la realitza habitualment l'usuari i es pot modificar de forma interactiva en la majoria dels sistemes de visualització. La tècnica del color fals *(colored range method)* [Far83] per a models de voxels opacs, es basa en aquest principi: per tal de diferenciar objectes, l'usuari assigna un color a cada superfície o rang de valors a pintar. El color amb el que es pinten els voxels frontera és l'assignat ponderat amb un factor proporcional a la distància a l'observador.

S'exposen a continuació els principals mètodes de càlcul del factor-normal. Com es veurà, cap d'ells no depén aquí dels valors associats als voxels: són funció solament de la seva distribució espaial. Això es deriva del fet que els valors indiquen solament la presència o absència de la superfície. Els voxels interiors són completament plens de sòlids i els exteriors, en són buits. Per això, Levoy [Lev91] classifica aquests algorismes com de *binary shading*.

Cal distingir dos casos: quan les dades procedents de l'EPA són directament les cares visibles dels voxels, perque s'ha procedit a un recorregut ordenat de les unitats volumètriques, i quan es té un buffer de profunditats resultat d'un algorisme de Z-buffer. En el primer cas, es pot aplicar directament un suavitzat de Gouraud o de Phong a les cares dels voxels visibles o bé realitzar el càlcul del factor-normal de les següents formes:

- **Distance Only Shading** $factor\_normal(P) = 1$ En aquest cas el shading es limita a l'atenuació en profunditat i permet, per tant, d'obtenir imatges suaus, però que no tenen en compte la orientació de les cares, el que dificulta la percepció de les formes dels objectes a visualitzar. És un algorisme senzill i un dels de menor cost [CHR85][HeU81][VMW83]. L'algorisme del *colored range method* [Far83], esmentat més amunt, pertany a aquesta categoria.
- **Constant Shading** o **Block Constant Shading**: $factor\_normal(P) = cos(\frac{\theta}{n})^p$ on θ és l'angle que formen la normal a la cara visible del voxel a la que pertany el punt P, calculada en el centre d'aquesta cara, i la direcció d'observació. Aquest mètode millora la percepció de la forma dels objectes però en dóna una aparièncía artificial, com la d'un tauler d'escacs. Els factors p i n, empírics, serveixen precisament per a millorar-ne la suavitat (a [CHR85] $p = 0.6$ i $n = 2$).
- **Image Based Contextual Shading** : $factor\_normal(P) = \sum_{i=1}^{4} w_i cos(\frac{\theta_i}{n})^p$ on $i = 0$ correspon a la cara f_0 del voxel visible a la que pertany P, i $i = 1..4$ correspon a cadascuna de de les quatre cares, f_i, veïnes a la cara de P [HeU83][CHR85]. Els pesos w_i són calculats de forma empírica en funció de la orientació de les cares respecte de la llum i tenint en compte que es verifica $\sum_{i=1}^{4} w_i = 1$. Quan la normal a una cara forma un angle menor que 45 graus amb la direcció de la llum, aquesta cara és considerada crítica. Si la cara f_0 és crítica, w_0 és proporcional al nombre n_c de cares crítiques veïnes, a les que s'assigna pes 0. En aquest cas, el pes de les cares no crítiques és el mateix per totes elles i es calcula com: $\frac{(1-w_0)}{(4-n_c)}$. Es resolt el cas en que f_0 és no crítica simètricament.
- **Normal Based Contextual Shading** $factor\_normal = cos(\frac{\theta}{n})^p$ on θ és l'angle que forma un vector normal **aproximat** a la cara de P, $\bar{w}$, amb la direcció de la llum. La orientació de $\bar{w}$ es calcula a partir de la de les seves cares veïnes: existeixen 81 possibles combinacions de les orientacions veïnes, de les que es poden deduir 25 possibles orientacions de $\bar{w}$. Les imatges obtingudes no tindran per tant més que 25 matisos. Aquests es poden precalcular i guardar en una taula de lookup. Tanmateix, per a accelerar l'algorisme es pot associar a cada cara visible un codi d'orientació *(neighbor-code)*. El desavantatge és llavors l'increment de memòria requerit.
- **Congradient Shading** $factor\_normal = f(\bar{g})$ on $\bar{g}$ és el vector gradient. Aquest algorisme té dues versions: unidireccional i bidireccional, ambdues desenvolupades per a la seva implementació en l'arquitectura *Cube* [CKB90]. El mòdul de shading, integrat en l'arquitectura, treballa en paral.lel a la visualització i rep per entrada, l'un darrera l'altre, els voxels visibles, el seu color original c_f i la seva profunditat. En la versió unidireccional, es calcula per cada voxel visible una funció $\bar{N}_x$ de congradient horizontal simplificada i retallada entre un valor màxim i mínim $\frac{H}{2}$ i $\frac{-H}{2}$. H és el tamany del espai congradient escullit d'acord amb uns criteris heurístics descrits a [CKB90]. $\bar{N}_x = z_{i+1,j} - z_{i-1,j}$ on $\bar{N}_x$ és un índex a una taula de lookup bidimensional que conté H matisos per cadascun dels colors originals del voxels. L'organització d'aquests matisos en la taula es fa en funció de la direcció de la llum. Quan aquesta coincideix amb la direcció d'observació, el color original es centra en la taula. i, en cas contrari, es desplaça (veure figura 5.4). D'aquesta manera una modificació de la direcció de la llum només implica fer unes permutacions circular en la taula. La versió bidireccional és similar però treballa amb blocs de voxels en lloc de files.

Quan la sortida de l'EPA és un Buffer de profunditats, s'aproxima la direcció normal a la superfície amb el gradient: $(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, 1)$ És l'anomenat **Gradient Shading, Depth Gradient**, **Distance Shading**, o **Z-Buffer Gradient**. D'acord amb [GoR85], l'estimació de les derivades es pot fer calculant,

- la diferència central: $\partial_c = \frac{1}{2}(z_{i+1,j}, z_{i-1,j})$ amb $2 \leq i \leq N-1, 1 \leq j \leq N$
- la diferència trassera *(backward)* $\partial_b = z_{i,j}, z_{i-1,j}$ amb $2 \leq i \leq N, 1 \leq j \leq N$

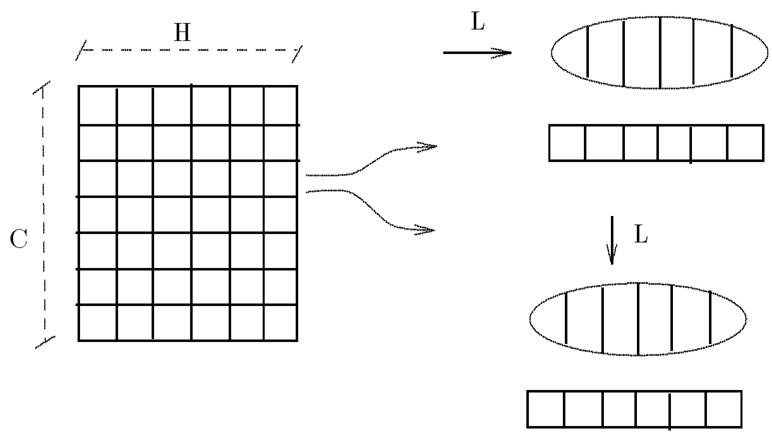

Figura 5.4 *Mètode de congradient*

– la diferència davantera *(forward)* $\partial_f = z_{i+1,j}, = z, z_{i,j}$ amb $1 \leq i \leq N-1, 1 \leq j \leq N$

o bé, ponderant ∂_b i ∂_f segons la expressió:

$$\frac{W_b \partial_b + W_f \partial_f}{W_b + W_f}$$

D'aquesta manera s'aconsegueix tractar més acuradament els casos en els què les diferències indiquen un canvi brusc produït per una discontinuïtat entre dues superfícies.

A [GoR85] s'analitzen els resultats obtinguts amb diferents valors per els pesos W_b i W_a. L'inconvenient més gran d'aquest mètode és, precisament, que pot donar errors en les discontinuïtats locals degudes a la discretització. A més, d'acord amb [CKB90] només permet de discriminar entre 16 valors de normal significatius. En canvi, el gran avantatge d'aquest model és el seu baix cost ja que opera directament en 2D. Bright i Laflin [BrL86] presenten dos mètodes alternatius de càlcul del gradient, ambdós coneguts com a **Gradient Interpolation Shading**, que resulten adients quan l'imatge presenta petites discontinïtats. En ambdóshas casos cal primer detectar aquestes discontinuïtats i a continuació interpolar el gradient per trams entre dues discontinuïtats successives. Tam i Davis [TaD88] proposen una variant d'aquest model, **Normal Averaging** que permet de millorar la qualitat de les imatges obtingudes analitzant un veïnatge més ampli de 3x3 punts de l'imatge.

4. Models de voxels i octree transparents o semi-transparents.

A diferència del cas anterior, aquí es pretén visualitzar directament el volum. Per això els algorismes de shading per a aquests models són anomenats *direct volum rendering* per oposició als de *isosurface rendering* de l'apartat anterior. Wilhems [WVG91] mostra els avantatges comparatius entre aquests dos enfocs: la visualització de superfícies dóna imatges més clares i entenedores i és habitualment de menor cost computacional. La visualització de volum, que tracta un nombre elevat d'unitats volumètriques és de cost elevat però permet d'apreciar l'interior del volum. En canvi, d'acord amb Höhne [HBP90] el shading de volums semi-transparents seria recomenable en el casos en els què les superfícies no es poden determinar de forma precisa. Alguns autors [Lev90b] [KYC90] combinen els dos enfocs i desenvolupen algorismes per a escenes mixtes, formades per volums semi-transparents i superfícies, generalment opaques. Aquest és un tema de recerca obert en aquest àmbit.

Classificació de materials dins del volum

Les unitats volumètriques d'entrada al shading tenen associat un valor continu, generalment comprés entre 0 i 1. Aquest valor és representatiu del contingut del voxel. En particular, si en el volum hi ha més d'un material, el valor pot tenir dues interpretacions possibles [DCH88] (veure figura 5.5):

– l'unitat de volum està parcialment ocupada per un tipus de material que és homogeni respecte del paràmetre.És a dir, la frontera d'aquest material passa per l'unitat de volum i el valor n' indica el tant per cent d'ocupació.

- l'unitat de volum està completament ocupada per un tipus de material que no és homogèni respecte d'aquest paràmetre. El valor indica el valor relatiu del paràmetre respecte dels seus valors màxim i mínim.

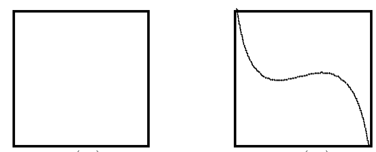

Figura 5.5 *Interpretacions del contingut d'un voxel: (a) ocupació total, densitat 50%. (b) ocupació 50%, densitat total*

Drebin, Carpenter i Hanrahan [DCH88] proposen un mètode probabilístic de classificació de material (veure figura 5.6). Si es coneixen els n materials continguts en el volum, el tant per cent del material i contingut en un voxel de valor I, $p_i(I)$, es pot expressar com:

$$p_i(I) = \frac{P_i(I)}{\sum_{j=1}^{n} P_j(I)}$$

essent, $P_i(I)$ la probabilitat, coneguda a priori, que el material i doni a un voxel el valor I.

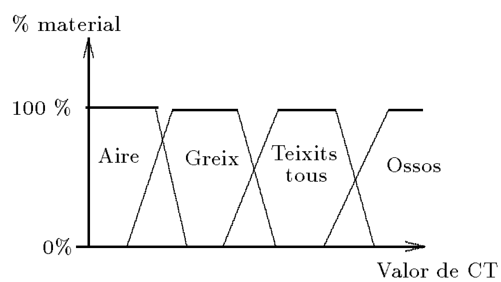

Figura 5.6 *Classificació de materials segons el mètode de Drebin i Carpenter*

L'aplicació d'aquest mètode requereix però, un coneixement previ del què hi ha en el volum, com és sovint en el camp de la medicina però no sempre possible en altres camps.

Models d'il.luminació

El shading de volums semitransparents es basa en la hipòtesi que la llum incident interacciona amb els voxels, que són parcialment transmissors i parcialment reflectors. La llum reflectida és la que contribueix directament a la intensitat percebuda per l'observador. La llum transmesa és la que incideix en els voxels situats a darrera. Les proporcions relatives de llum transmesa i llum reflectida són funció de la ocupació o densitat dels voxels i per tant depenen directament dels seus valors associats. Per això, un pas previ al shading, consisteix en calcular per cada voxel els paràmetres d'il.luminació. Aquest pas, anomenat de classificació s'analitzarà en l'apartat següent.

Recentment, Kreuger [Kreu90] ha proposat un model d'il.luminació alternatiu. Es considera que el volum és atravessat per un conjunt de partícules virtuals que incideixen amb una intensitat inicial donada. El model analitza la interacció de les partícules virtuals amb el volum procedint a un balanç d'energia inspirat en la teoria de transport d'ones de manera similar als models de Blinn [Bli82] i Kajiya [KaH84] per a punts presentats més amunt.

Càlcul dels paràmetres d'il.luminació

Un pas previ al shading, anomenat habitualment *classificació*, consisteix en assignar a cada voxel quatre paràmetres: les intensitats RGB i l'opacitat α, inversa de la transparència. Aquest pas es fa mitjançant *funcions de transferència* . Farrell [FaC89] anomena **model òptic** el format per l'espai de voxels i les funcions de transferència.

Diferents autors han proposat possibles funcions de transferència. Així, Upson i Keeler [UpK88], discuteixen els avantatges comparats de funcions de transferències linealment creixents amb el valor del paràmetre, que permeten de veure la gradació de valors dins del voxel, en front de funcions discontínues, que permeten de resaltar determinades propietats (figura 5.7).

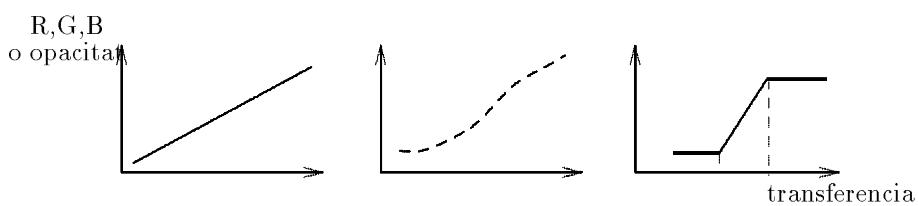

Figura 5.7 *Funcions de transferència*

D'altra banda, Levoy [Lev88] calcula les intensitats aplicant el model de Phong que contempla reflexions difuses i especulars, prenent com a valor de vector normal el vector gradient en el voxel: $\bar{g}(g_x, g_y, g_z)$, amb

$$g_x = \frac{1}{2}(f(x_{i+1}, y_j, z_k) - f(x_{i-1}, y_j, z_k))$$
$$g_y = \frac{1}{2}(f(x_i, y_{j+1}, z_k) - f(x_i, y_{j-1}, z_k))$$
$$g_z = \frac{1}{2}(f(x_i, y_j, z_{k+1}) - f(x_i, y_j, z_{k-1}))$$

com en el *gray-level gradient shading* . Pel que fa les opacitats la funció de transferència que proposa és:

$$\alpha(P) = \begin{cases} -1 & \text{si } g = 0 \text{ i } f(P) = f_v \\ 1 - \frac{1}{r}\frac{|(f_v - f(P)|}{g} & \text{si } g > 0 \text{ i } f(P) - r * g \leq f_v \leq f(P) + r * g \\ 0 & \text{en cas contrari} \end{cases}$$

on $P = (x_i, y_j, z_k)$ $g = \|\bar{g}(x_i, y_j, z_k)\|$, f_v és el valor de filtre per a la superfície que es vol pintar i r és el nombre de voxels o gruix de la zona de volum transparent.

Westover [Wes89] proposa una classificació basada en qualsevol valor associat als voxels: gradient, densitat o coordenades. Hi ha quatre taules de 256 entrades indexades amb el valor del voxel: taula d'emitància, de reflexió , d'opacitat i de modulació d'opacitat. La intensitat associada al voxel es calcula segons una adaptació del model empíric de Phong semblant a la utilitzada per Levoy [Lev88] i en la que la normal també és aproximada per el vector gradient. L'opacitat és:

$$\alpha = TO[idx_{opac}] * TMO[idx_{mod}]$$

on TO i TMO són respectivament les taules d'opacitat i modulació d'opacitat i idx_{opac} i idx_{mod} els índexs a aquestes taules.

Mètodes per a resaltar propietats

Per a resaltar propietats o determinades zones dins del volum en la visualització es pot, bé modificar de forma local i temporal els valors associats als voxels, bé modificar directament els valors d'opacitat associats. Un exemple del primer enfoc és el proposat per Drebin, Carpenter i Hanrahan [DCH88]. Defineixen un volum d'interés o *matte* format per voxels amb un valor associat predefinit. S'opera el matte amb el volum, voxel a voxel, per tal d'anular, reduir o al contrari augmentar els valors del voxel. El segon enfoc és adoptat per Ma, Cohen i Painter [MCP91] que permeten a l'usuari d'indicar gràficament un punt, centre de la regió (que es desitja resaltar i un radi d'influència. L'opacitat dels voxels *(seed)* s' atenua de manera que els voxels que es troben fora del radi d'influència són transparents (figura 5.8).

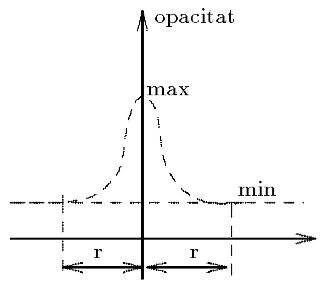

Figura 5.8 *Atenuació de la opacitat en funció de la ditancia al centre del la regió d'influencia matte*

Algorismes de shading

En l'apartat 5.3, s'ha vist que l'EPA per a models de voxels transparents o semitransparents consisteix bàsicament en: el ray-casting o un recorregut ordenat dels voxels, generalment FTB o BTF. Per a cada cas, existeixen algorismes de shading diferents que exposem a continuació.

– **Mètodes de shading basats en el ray-casting:**

Aquests algorismes tracen un raig en la direcció z per cada pixel de pantalla i en calculen la intersecció amb el volum. Habitualment es calculen punts d'intersecció espaiats a intervals regulars al llarg de raig. Per cadascun d'aquests punts, es calcula la seva opacitat i la seva intensitat per interpolació trilineal de les opacitats i intensitats associades als vèrtexs dels voxels als que pertanyen els punts (veure figura 5.9). La intensitat en el pixel associat a cada raig, s'obté integrant les intensitats dels punts d'intersecció al llarg del raig.

Així, d'acord amb Levoy [Lev88], la intensitat acumulada al llarg de n punts mostrejats al llarg d'un raig que passa pel pixel(i,j)és:

$$I_\lambda(i,j) = \sum_{k=0}^{n} \left[I_\lambda(x_i, y_j, z_k) * \alpha(x_i, y_j, z_k) \prod_{l=k+1}^{n} (1 - \alpha(x_i, y_j, z_l)) \right]$$

amb $\lambda = (r, g, b)$, essent $I_\lambda(x_i, y_j, z_k)$ i $\alpha(x_i, y_j, z_k)$ la intensitat i la opacitat, respectivament en el k-èssim punt del raig (i,j) (veure figura) i prenent per valors inicials en el fons *(background)* $\alpha(x_i, y_i, z_0) = 1$ i $I_\lambda(x_i, y_j, z_0) = IBackground_\lambda$.

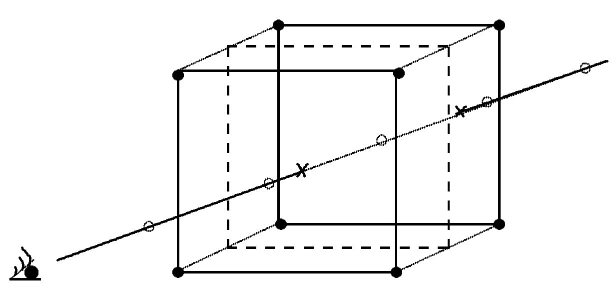

Figura 9.9 *Ray casting de volum*

Aquesta formulació es basa en un recorregut BFT per al càlcul dels punts d'intersecció. Upson i Keeler [UpK88] proposen un recorregut FTB, que presenta l'avantage que es poden parar els càlculs d'intersecció en quan l'opacitat acumulada és igual a 1. Un problema que planteja aquest mètode és el seu elevat cost computacional. Per això, un tema de recerca obert, que ha estat recentment tractat per alguns autors ([Lev 90], [YaK92], [DaH92]), és el de l'aprofitament de la coherència eapaial per a accelerar els càlculs d'intersecció.

L'algorisme de Sabella [Sab88], inspirat del de Kay i Von Herzen [KaH84], proposa un model d'il.luminació més general en el que els voxels poden ser emissors de llum *DE varying Density Emitters (DE).*

A la taula 5.1 estan citats d'altres algorismes de ray-tracing basats en en mateix principi.

- **Mètodes de shading basats en el recorregut ordenat del volum**

La principal diferència d'aquests mètodes amb els anteriors és que projecten ordenadament els voxels, FTB, BTF o en un ordre convenient quan es tracta d'un model octree, i la integració de les intensitats i opacitats es fa directament en espai imatge (figura 9.10), sobre un buffer d'intensitats: és l'anomenada*composició*. D'acord amb Wilhems [Wil90] el procés BTF dels voxels presenta l'avantatge de poder obtenir imatges temporals del interior del volum, i de veure com es va composant gradualment l'imatge, el que permet d'entendre millor el volum.

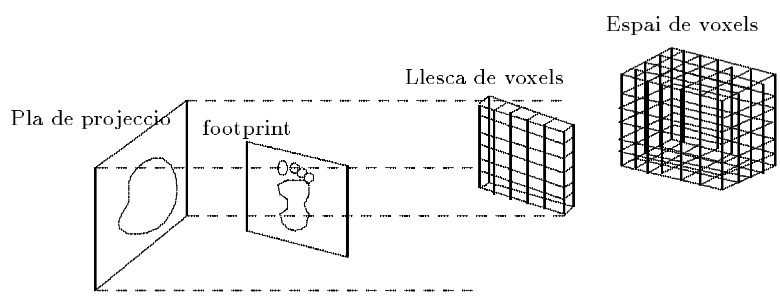

Figura 9.10 *Composició de projeccions de volum*

L'algorisme de Westover [Wes89][Wes90] que consisteix en 4 fases: en primer lloc es fa el procés de visualització que converteix els índexs enters (i,j,k) associats a cada voxel del volum en coordenades (x,y,z) d'observació: és el que l'autor anomena *screen mapping* . A continuació s'associa a cada voxel els valors RGB i α segons el procediment esmentat més amunt. La tercera fase de l'algorisme, anomenada de *reconstrucció*, consisteix en el càlcul de l'imprenta o *footprint* que deixa la projecció del voxel en el buffer d'imatge. És una funció continua que s'obté per convolució de les dades discretes del voxel projectat:

$$footprint(x,y) = \int_{\infty}^{\infty} h_v(x,y,w)dw$$

on $h_v()$ és el kernel de reconstrucció del volum i (x,y) estan expressades en termes relatius a la posició del centre del kernel de convolució. Per acabar, cal sumar de forma ponderada el footprint a l'imatge acumulada. La contribució d'un footprint a un pixel P és:

$$pes(u,v)_P = footprint(P_x - u, P_y - v)$$

on (u,v) són les coordenades absolutes de la projecció en espai imatge. Els footprints es poden calcular al principi de l'algorisme. Es crea una taula indexada amb el desplaçament de la projecció del voxel respecte del centre d'un pixel. La taula conté el pes associat a cada pixel en el veïnatge de la projecció.

En un recorregut FTB la suma d'intensitats i opacitats en un pixel es fa amb les fòrmules:

$$I_{acu} := I_{acu} + ((1 - \alpha_{acu}) * (I_n * \alpha_n))$$

$$\alpha_{acu} := \alpha_{acu} + ((1 - \alpha_{acu}) * \alpha_n)$$

on el subindex n indica el nou punti el subindex acu els valors acumulats. En un BTF les formules són:

$$I_{acu} := ((1 - \alpha_n) * I_{acu}) + (I_n * \alpha_n)$$

$$\alpha_{acu} := ((1 - \alpha_n) * \alpha_{acu} + \alpha_n)$$

L'algorisme de *cell-by-cell processing* proposat per Upson i Keeler [UpK88] *V-buffer* processa els voxels FTB i analitza la seva contribució a l'imatge final línia per línia, com en l'algorisme de *scan-line Z-buffer*. Per cada línia d'escombrat es calculen els segments *spans* projectats d'intersecció del voxel amb el plà d'escombrat. Els valors d'intensitat associats als pixels dels segments s'obtenen per interpolació trilineal dels valors associats als vèrtexs del voxel.

Com s'ha descrit en l'apartat anterior, Wilhems i Van Gelder [WiG91] desenvolupen un algorisme de projecció que aprofita la coherencia espaial en les projeccions paral.leles. Cada voxel és dividit en cel.les . L'assignació d'intensitat i d'opacitat amb les funcions de transferències es fa directament en els vèrtexs d'aquestes cel.les i la integració en profunditat es realitza tamb'e a nivell de cel.les.

Una línia de recerca oberta en aquest camp és l'aprofitament de la coherencia espaial per tal reduir el nombre i el tamany de les projeccions. Cal citar en aquest sentit, l'algorisme de Laur i Hanrahan [LaH91] que, basant-se en una estructuració jerárquica del model de voxels, permet de realitzar un *splatting* progressiu a diferents nivells de resolució.

D'altres algorismes d'aquesta familia són referenciats a la taula 5.1.

6 Aplicacions

En aquest apartat es mostren les diferents àrees d'aplicació del modelatge de volum trobades a la bibliografia i s'analitza la problemàtica específica de cadascun d'aquests camps.

Per a cada aplicació, es presenten les característiques pròpies de les dades que fan que algunes tècniques explicades al llarg d'aquest raport siguin més adequades per una àrea que per una altra. Així mateix, s'esmenta l'ús que cada aplicació fa del sistema que habitualment és un dels següents:

- Visualitzar i analitzar les dades volumètriques amb la possibilitat de comparar-les amb altres dades i explorar les dades amb diferents tècniques de visualització.
- Interaccionar amb les dades volumètriques per tal de modificar-les i mesurar-les.
- Simular moviment de partícules o fluïds per analitzar possibles efectes en el model i comportaments ideals d'aquest moviment. És a dir, estudiar la dinàmica de fluïds aplicada a una aplicació concreta. Aquesta prestació és possible si les dades volumètriques tenen associats propietats vectorials.

Les diferents àrees d'aplicació trobades en la bibliografia són:

- **Medicina :** Aquesta àrea és una de les més desenvolupades dins del modelatge de volum i, per tant, és el camp motor d'aquests sistemes (veure Fotos 16, 17, 18 i 19). És molt important la precisió de les dades, una bona qualitat d'imatge i facilitar les dades en temps real per poder diagnosticar correctament certes malalties. Tanmateix, l'exactitud i precisió de les dades és una de les raons per les què segons [Rho91] aquestes tècniques de modelatge de volum tenen dificultat per introduir-se en la pràctica habitual de la medicina. Un altre motiu de la seva poca utilització clínica és que tenen un cost massa elevat encara.

  Dins de les aplicacions mèdiques que utilitzen el modelatge de volum per analitzar les dades es troben: els diagnòstics de cirurgia (a nivell de neurocirurgia [VMW83], [SYM92], cosmètica [NOB88] i ortodòncia), les anàlisis radiològiques de fractures [FDM87], l'anàlisi de la morfologia de vasos sanguinis [BGS85][VPR92] i d'estructures vasculars complexes [CLH89]. Les aplicacions que necesiten interaccionar amb les dades volumètriques són aquelles que simulen operacions, tractaments de radiologia

[SuG78][Lev90b] (per exemple, per trobar la situació i orientació òptima pels rajos de radiació, de forma que es maximitzi l'efecte dels raigs en el tumor i es minimitzi aquest efecte en els òrgans veïns) i disseny i manufactures de pròtesis [GRW87][RKR87]. També s'han desenvolupat aplicacions que mesuren àrees del cos humà com [BBO91].

– **Ciències de la terra :** En aquestes aplicacions es combinen models empírics i simulacions numèriques per analitzar fenòmens geològics. Generalment, la informació és incompleta i, en alguns casos, contradictòria pel què fa a les dimensions, geometries i varietats de les unitats geològiques. A més les dades geològiques són molt complexes: existeix una forta interrelació entre les superfícies provocada per interferències entre processos geològics al llarg de la història. Així per exemple és freqüent voler analitzar una roca i desconèixer les propietats dels estrats que l'envolten. Un problema adicional és el gran nombre de variables a analitzar i a interelacionar en cada unitat geològica: temperatura, salinitat, pressió... Totes aquestes característiques impliquen que el manegament de les dades sigui molt complexe, i més si considerem que en aquest tipus d'aplicacions totes les escales són interessants, des del nivell microscòpic fins a nivell macroscòpic (Fotos 21 i 22).

  En geofísica s'han desenvolupat aplicacions d'anàlisi visual d'estrats com contorns de superfícies i visualitzacions semitransparents [WoL88], [Sab88] i [HiS89a]. En oceonografia s'estudien tant les capes estratosfèriques [Ros89a], [Ros89b], com la dinàmica de les corrents marines [FAO91]. En prospeccions petrolíferes s'analitza l'existència i el moviment de fluïds en les reserves subterrànies, i es calcula el seu volum [Wat87], [SaC89]. Dins de la meteorologia, s'estudia la capa d'ozó, les temperatures de la terra [Tre92], i d'altres fenòmens com les tempestes i els tornados [UpK88] i la circulació dels vents [CrM92][HiS89a].

– **Bioquímica i biologia:** És una de les àrees d'aplicació en les què s'ha estat treballant des de fa anys, ja que les mol.lècules complexes, com les proteïnes, són estructures difícils d'interpretar sense l'ajut del computador. S'usen tècniques de difracció de raigs X per construir mapes de densitats d'electrons en una xarxa tridimensional. Es fan reconstruccions a partir de dades procedents de microcopis electrònics i dades procedents de CT's. Una problemàtica adicional en aquest camp és l'escala microscòpica en la que s'ha de treballar (veure Foto 20).

  Dins de bioquímica s'estudia principalment el comportament de mol.lècules complexes [WNB91] [GoO89] [MHC90] i enzimes [MPS92] i dins de la biologia, l'estructura dels cranis i òrgans d'animals [ScM86] [SMW88].

– **Química:** Com entrada s'utilitzen les posicions exactes dels àtoms en una mol.lècula (en xarxes de 100x100x100). Fonamentalment, s'analitzen i es visualitzen les dades volumètriques, mesurant certes propietats com temperatures, pressions, magnetismes, potencials lipofílics i hidrofòbics. Els sistemes de modelatge de volum s'utilitzen per a realitzar predicions acurades de propietats de mol.lècules petites i simular els efectes de la introducció de noves components en una mol.lècula, el que evita haver de sintetitzar-les.

  Els sistemes de modelatge de volum s'utilitzen en la recerca de materials semiconductors en l'enginyeria elèctrica, en l'estudi de catalitzadors i en ceràmiques i superconductors a altes temperatures en la indústria. A [KKF89] es simula el transport del quantum electrònic en semiconductors aplicant teories de dinàmica de fluïds.

– **Arqueologia:** Dins aquest camp, es poden tenir dades molt diferents segons els dispositius d'input [Rei89]: des d'inspeccions topogràfiques i prospeccions geofísiques, fins a CT de mòmies.

  Una aplicació és la reconstrucció de runes, en la què també es treballa amb models de sòlids [Rei89]. Una altra possible aplicació és l'anàlisi de mòmies. Cal remarcar que en aquest cas és necessari operar amb el model per tal d'extreure les unitats volumètriques que contenen el guix que envolta la mòmia, per a poder identificar correctament els seus òssos [YYO92].

– **Mecànica de fluïds :** Sota aquest títol s'inclouen tot tipus d'aplicacions que treballen amb dades originades per simulacions computacionals de dinàmica de fluïds (CFD), encara que en els apartats anteriors, s'han referenciat concretament algunes d'elles per enmarcar-les en aplicacions concretes (circulació de masses d'aire, dinàmica de corrents marines,...). En aquestes simulacions es mesuren un gran nombre de propietats físiques que poden ser: escalars com la pressió i les temperatures o vectorials com la velocitat i la acceleració. Aquestes propietats es representen en forma de matrius tridimensionals. Les dades associades a cada voxel són cadascuna d'aquestes matrius. Per analitzar els resultats de les simulacions, s'utilitzen tècniques d'animació per a visualitzar les propietats vectorials [Ups86],[Gar90]. Simultàniament amb aquestes, es poden visualitzar també les propietats escalars [Rob85], [CrM92] i [GeW92].

7 Implementacions

Els sistemes de modelatge de volum han de manegar una gran quantitat de dades. Per aconseguir que puguin operar de forma interactiva en temps real o en temps quasi real és necessari recórrer bé al disseny d'algorismes que aprofitin al màxim les capacitats del software i del hardware, o bé al disseny d'arquitectures específiques. En la bibliografia trobada, s'intenta sobretot accelerar el procés de visualització, ja que és un dels processos més costosos que necesita una interacció amb l'usuari.

Els tipus d'arquitectura en els que s'implementen els sistemes de volum són:

- Arquitectures de propòsit general: són arquitectures convencionals sobre les quals es dissenyen algorismes desenvolupant diferents estratègies per accelerar les operacions volumètriques. Per exemple: Farrell, MIPG, ANALYZE, Pixar/Vicom, Pixel Planes.
- Arquitectures de propòsit específic: són arquitectures dissenyades especialment per a suportar models volumètrics (concretament, models de voxels *(voxel-encoded)*. Són arquitectures com Voxel Processor, Insigth, GODPA, Cube.

L'elecció d'aquestes arquitectures determina si l'acceleració es fa via software (arquitectures de propòsit general) o via hardware (arquitectures de propòsit especícific). Aquestes arquitectures s'analitzaran més detalladament en els apartats 7.1 i 7.2.

Les arquitectures poden ser seqüencials o paral.leles. Les paral.leles es poden analitzar segons la seva topologia multiprocessador o segons la seva organització de la memòria.

Per a classificar la topologia dels computadors paral.lels s'utilitza molt sovint la taxonomia desenvolupada per Flynn [Fly66], basada en el criteri de l'estructura de la màquina a nivell macroscòpic i en l'organització de les instruccions i de les dades en els processadors. Segons aquesta classificació, les màquines s'agrupen en:

- SISD : Singe Instruction Single Data
- SIMD : Singe Instruction Multiple Data
- MISD : Multiple Instruction Single Data
- MIMD : Multiple Instruction Multiple Data

Pels sistemes paral.lels de modelatge de volum, són útils les màquines que maneguen simultàniament moltes dades i, per tant, s'utilitzen arquitectures SIMD i MIMD. Per aquesta raó, s'expliquen més detalladament a continuació.

L'arquitectura **SIMD** està formada per processadors que executen les mateixes instruccions, però amb diferents dades. Els programes concurrents implementats sobre arquitectures SIMD són anomenats **homogenis**, perque els seus *threads* (o conjunt d'instruccions) són idèntics però s'executen amb diferents dades. En aquests programes, les expressions condicionals són de gran utilitat per a controlar els processadors actius i inactius en un cert instant de l'execució.

L'arquitectura MIMD, sovint referenciada com multiprocessador, està formada per processadors independents, de forma que cadascun d'ells pot executar les seves pròpies instruccions. Els programes concurrents sobre arquitectures MIMD s'anomenen **heterogenis**, ja que cada *thread* executa diferents instruccions.

L'accés als recursos compartits, com la memòria o fitxers, s'ha de sincronitzar amb molta precisió per evitar resultats inconsistents.

Cal esmentar que les arquitectures de propòsit general i les de propòsit específic poden ser tant de tipus SIMD com MIMD, depenent del tipus de processadors que les formin.

Per a classificar l'organització de la memòria en les arquitectures paral.leles SIMD i MIMD, Meuer i Stomaier [MeS92] diferencien tres casos: dades compartides, distribuïdes o virtualment compartides (veure figura 11.1). Cada cas comporta diferents problemes d'accés a memòria i de comunicació entre els diferents processadors.

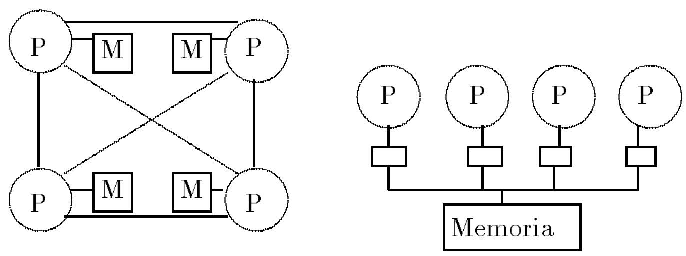

Figura 11.1 *Memòria distribuïda i memòria compartida*

Les arquitectures amb memòria compartida s'organitzen en N processadors i una memòria principal. Els processadors accedeixen a memòria mitjançant un bus. El coll d'ampolla d'aquests sistemes és l'ampla de banda del bus i l'alta probabilitat de col.lisions en memòria. Per evitar accessos innecessaris i possibles conflictes a memòria, cada processador té associat una memòria *cache*. Degut a la càrrega del bus, per tenir operativitat, el nombre de processadors en les arquitectures amb memòria compartida és limitat (de l'ordre de 10 o 12 processadors).

En les arquitectures amb memòria distribuïda, cada processador té associat un tros de memòria. En aquests sistemes, la comunicació entre processadors ha de permetre l'intercanvi de dades i la sincronització entre processos. El cost de la distribució de les dades depén fortament de la topologia dels canals de comunicació. Aquests tipus d'arquitectures permet tenir un nombre ilimitat de processadors, ja que no tenen el coll d'ampolla de la memòria compartida. Això no obstant, un nombre massa elevat de processadors pot sobrecarregar la comunicació i la sincronització entre processos. A vegades, es connecta un nombre petit de processadors (per exemple 8) a un *cluster* per accelerar la velocitat d'intercomunicació, encara que això comporta un alentiment en la comunicació amb processadors d'altres *clusters*. En ambdós casos, els algorismes de comunicació i sincronització han d'implementar-se dins de l'aplicació. Per evitar-ho, hi ha sistemes operatius que implementen aquests algorismes (memòria virtual compartida) [NiL91], fent molt més fàcil la paral.lelització d'una aplicació ja existent.

Un altre concepte important a considerar en les arquitectures paral.leles és la granularitat, que és la unitat màxima que es pot paral.lelitzar en una màquina. Es diu d'una màquina que és de grànol gros si, per exemple, pot executar blocs grans d'instruccions en paral.lel; en canvi, és de grànol fi si només pot operar bits en paral.lel.

Com s'ha esmentat al principi de l'apartat, segons la bibliografia trobada, les operacions dels sistemes de modelatge de volum que més s'intenten accelerar utilitzant arquitectures paral.leles són les seves etapes més costoses de la visualització. Com les etapes de visualització tenen una estructura seqüencial (és a dir, entre elles hi ha conflictes de rang-domini), es construeix un pipeline de tants passos com etapes té la visualització. Els passos de les etapes més costoses es paral.lelitzen. Altres nivells de paral.lelisme es poden aconseguir partint el display en diferents pipelines o extenent qualsevol del passos en una organització de multiprocessadors. La comunicació entre els diferents passos s'implementa amb diferents mecanismes (FIFO, memòria compartida,...). Habitualment, s'agrupen les operacions que treballen en coordenades de món (exploració del model i transformacions geomètriques) i les que treballen en coordenades de pantalla (epa, shading i rasterització) (veure apartat 5).

En els següents apartats s'estudien les arquitectures de propòsit general i les de propòsit específic desenvolupades en sistemes de modelatge de volum i s'analitzen els aventatges i inconvenients que comporten cadascuna d'elles.

Per avaluar les arquitectures es té en compte els següents paràmetres:

- Arquitectura: topologia i composició de la màquina. Determina la interacció entre el hardware i el software i les restriccions del model utilitzat, de *throughput* i de qualitat d'imatge.
- Estratègia: manera d'utilitzar l'arquitectura per realitzar diferents tasques per tal d'aconseguir el màxim rendiment.
- Model de dades: model volumètric suportat per la màquina.
- EPA, Shading i Antialiasing: algorismes implementats.
- Color: tonalitats de grisos (B/N) o de colors.
- Operacions: utilitats que ofereix la màquina per manegar el model de dades.
- Velocitat o rendiment de la màquina: nombre d'imatges per segon que permet la màquina.***No es molt clar
- Resolució de pantalla: nombre de pixels de la pantalla.
- Grau de paral.lelisme: capacitat de paral.lelisme de la màquina: granularitat i nombre d'operacions simultànies suportades.
- Memòria: Distribució de la memòria i capacitat de la màquina (mesurada en Kbytes o en Kvoxels)
- Status operacional de la màquina: estat de desenvolupament actual de la màquina: disseny, prototipus, prototipus testejat, màquina en funcionament o màquina comercial.

7.1.- Arquitectures de propòsit general

Els algorismes implementats en aquest tipus de sistemes incrementen el seu rendiment via software. Per tant, en general es plantegen algorismes sobre màquines comercials ja existents, o bé, per tal d'optimitzar els passos més costosos de la visualització de volum s'han dissenyat màquines de propòsit general que implementen via hardware operacions gràfiques molt concretes, o s'ajuden d'acceleradors gràfics. Aquestes arquitectures poden ser seqüencials com la màquina de Farrell, MIPG o ANALYZE o paral.leles, com és el cas de Pixar/Vicom, Standent Stellar GS, AT&T Pixel Machine, Silicon Graphics 4D i Pixel Planes. Totes les màquines esmentades a continuació treballen amb un model de voxels (veure taula 7.1).

La implementació de Farrell [FYZ85] reparteix les tasques entre un Host i una Workstation. El Host, IBM 370/4341, realitza les operacions més costoses (rotació 3D i el suavitzat) mentre que les operacions més sencilles (projecció, transparència) es fan a la Workstation, IBM 7350. Els algorismes implementats són un recorregut BTF com a mètode d'EPA i el *Colored Range method*, descrit a l'apartat 5.3, per al shading. Es redueix la càrrega computacional eliminant o simplificant les operacions per a formar una imatge, per això, encara que no és una màquina paral.lela s'aconsegueixen visualitzacions ràpides (10-15 seg/frame).

L'ANALYZE [HeR85][RoH90] consisteix en un conjunt de processos que es comuniquen entre si i s'executen en una Workstation sense cap tipus de concurrència. La imatge es situa en memòria compartida per facilitar un procés ràpid de la imatge. La màquina ANALYZE dóna suport a entorn de visualització flexible, anàlisi de volums de dades 3D i suporta una gran variació d'operacions interactives 2D, display 2D i anàlisi d'imatges. Per eliminar parts amagades utilitza un z-buffer i per fer el shading utilitza un normal-averaging gradient. S'aconsegueix una velocitat de 27 imatges precalculades per segon.

MIPG (Medical Image Processing Group Machines) [URB90] s'implementa sobre PC. El disseny de l'última versió (3D98) dóna prioritat a un cost baix i una qualitat d'imatge acceptable davant de la velocitat. S'intenta reduir la quantitat de dades manipulades, basant-se en el model cuberille. Utilitza un z-buffer per a fer l'EPA i usa diferents tipus de shadings de cuberille (basats en la distància, contextual, gradient i Phong). En tenir només una CPU, resulta ser una màquina lenta (1-3 minuts).

Dins de les màquines paral.leles, el Pixar/Vicom [LeP84] utilitza un Host i quatre SIMD (anomenades Chap) i es connecten entre ells mitjantąnt un bus (Sysbus). Per accelerar l'accés a memòria, aquesta s'organitza en tres nivells de jerarquia. Les SIMD actuen a nivell de pixel. S'utilitza un software anomenat Software ChapVolume que permet fer un recorregut dels voxels BTF. Aquest software també permet la composició d'imatges (desenvolupada per [DCH88]). S'aconsegueix una visualització d'un volum de 256x256x256 voxels en minuts.

Pixel-Planes Machines són una sèrie de màquines que realitzen visualitzacions ràpides mitjançant el paral.lelisme massiu per solucionar el problema del coll d'ampolla del colorejat dels pixels. L'última versió s'anomena Pixel-Planes 5 [FGH85]. L'arquitectura està formada per: un *frame buffer* (de propòsit especial i doble per accelerar el refresc de pantalla), unitats de *Renderer* (SIMD's amb memòria local. Cadascun d'ells processa 128x128 pixels i utilitza el suavitzat de Gouraud), Processadors Gràfics (P.G.) (encarregats de la gestió de les dades. A cada processador gràfic se li assigna una estructura gràfica anomenada PixelPHIG amb les dades. El processador ordena les primitives en diferents trossos que corresponen a diferents parts de pantalla i, després, assigna cada tros a un *renderer* diferent) i una Host Workstation (que interacciona amb l'usuari i serveix per editar imatges). Tots aquests elements estan connectats mitjançant un *Token Ring* que treballa a 20 Mwords/seg. La resolució és de 1280x1024 i gràcies al doble buffer s'obté una velocitat de 24 frames/seg.

Stardent Stellar GS usa hardware de propòsit específic en la rasterització operant blocs de 4x4 pixels en paral.lel [UFL89]. Via software explota la velocitat de comunicació entre processadors vectorials mitjançant memòria compartida i un bus de banda ampla. Aconsegueix visualitzacions de volums de 256x256x70 voxels en 15 frames per segon.

AT&T Pixel Machine utilitza una xarxa regular *(mesh)* de més de 64 processadors en la què cada processador s'encarrega d'un tros de frame buffer. L'usuari té accés a llibreries de software per a la generació d'escenes 3D. L'EPA i el shading es fan amb un ray-casting. Cada processador de la xarxa traça els rajos des dels seus pixels [PoH89].

7.2.- Arquitectures de propòsit específic

Les arquitectures de propòsit específic utilitzen hardware especialitzat per accelerar al màxim els processos més costosos dels sistemes de volum.

Les màquines de propòsit específic més relevants en la bibliografia són: Cube (Kaufman), Insigth (Meagher), Voxel Processor (Goldwasser), PARCUM, 3DP4. En aquest apartat s'expliquen amb més detall les tres primeres arquitectures.

Cube

L'arquitectura Cube [Kau88] utilitza el paral.lelisme massiu per processar interactivament dades volumètriques a alta resolució.

Consta de tres processadors per carregar, manipular i visualitzar el conjunt de dades 3D (veure figura 11.2). L'unitat de paral.lelisme és un *beam* o conjunt, fila o columna de voxels. El bus que comunicar els tres processadors pot transmetre un *beam* en paral.lel i s'anomena *Voxel Based Write Bus* (VBWB). Mitjançant aquest bus, s'implementen les projeccions paral.leles. Cada voxel d'un *beam* es col.loca en un processador del bus VBWB. Tots els processadors que tenen un voxel opac accedeixen al bus simultàniament i posen, bit a bit, la profunditat del seu voxel, guanyant el més proper a l'observador. Aquesta acció té un cost temporal $O(log(n))$, on n és el nombre de voxels del *beam*. El bus implementa també transparències i plans de tall inabilitant els processadors de la regió de transparència o de tall abans de fer la determinació de visibilitat. El VBWB està format de n processadors per a una escena de profunditat n.

L'organització de memòria s'anomena *Cubic Frame Buffer* (CFB) i consta de n mòduls de memòria de n^2 voxels cadascun. Cadascun dels voxels adjacents en qualsevol dels eixos coordenats està emmagatzemat en un mòdul diferent. Concretament, tot voxel (x, y, z) està situat en el mòdul k, tal que $k = (x+y+z)mod(n)$, i (x, y) senyalen l'ubicació del voxel dins del mòdul k. D'aquesta forma no es produeixen conflictes en l'accés paral.lel a memòria i es garantitza un ràpid accés a les dades. D'altra banda, es pot modificar el CFB directament, sumant, esborrant o enmascarant voxels.

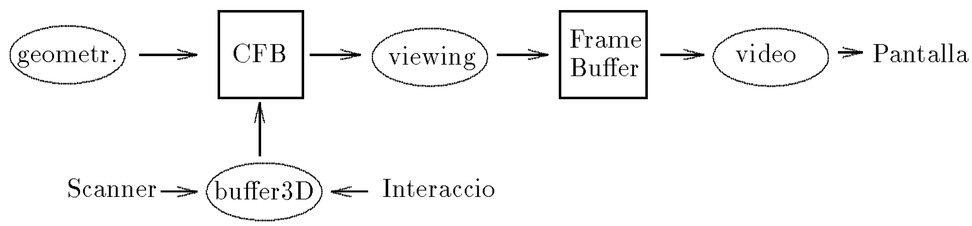

Figura 11.2 *Arquitectura Cube*

Les unitats funcionals són:

- *3D Geometry Processor :* converteix objectes 3D en la representació basada en voxels. Els objectes 3D poden estar formats per segments de línia, per polígons plans, corbes cúbiques,...
- *3D Frame Buffer Processor:* suporta l'interfície interactiva amb l'usuari i també l'entrada de dades provinents de *scanners*. Utilitza operacions de *bitblt* i mapejat de voxels *voxblt* per manipular i processar subcubs del CFB.
- *3D Viewing Processor:* genera projeccions 2D a partir de les dades del CFB. Utilitza la tècnica de *congradient* per realitzar el shading.

L'usuari, en temps real, pot alterar el color dels objectes de l'escena, modificar la transparència dels diferents materials i seccionar el volum. S'aconsegueixen imatges en temps real usant dos nivells de paral.lelisme: un nivell bast per interaccionar amb l'usuari i un nivell més fi per accelerar la visualització (processant un conjunt de voxels simultàniament). Permet una resolució de 512x512 pixels i el model pot ser de 512x512x512 voxels amb una informació de 8 bits per voxel.

Insight

L'arquitectura Insight [Mea85] permet la manipulació, l'anàlisi i la visualització d'informació mèdica de forma interactiva, utilitzant un model d'*octrees* per tenir un ràpid accés a memòria. La compressió que dóna aquest model permet una visualització ràpida però necesita d'un preprocés. A més, es redueixen el nombre de voxels a visualitzar al nombre de voxels d'interés (VOI) mitjançant processos de selecció (com plans de tall, *thresholding*,...)

L'arquitectura consta d'una Unitat de Procés de Sòlids i una Worstation. La unitat de Procés de Sòlids està formada per una cinta, un disc, un sistema microcomputador (M68010) i un processador de hardware especialitzat *(Solid Engine)*. La Workstation consta de dos perifèrics d'entrada-sortida (tableta 2D i *trackball* 3D), un CRT gràfic, un CRT alfanumèric i un teclat.

El *Solid Engine* és un processador modular hardware, és a dir, és un suport hardware per mòduls implementats amb processadors. Cada mòdul correspon a funcions de processament de sòlids (per exemple, hi ha un mòdul per a generar imatges i n'hi ha un altre per manipular i anàlitzar les dades). Aquest processador suporta C, Pascal, Fortran i assamblador del 68010.

Associat a cada voxel es poden tenir diferents nombre de bits:

- Un bit: per distingir dos tipus de voxels, els plens i els buits.
- 16 bits: Els 12 més significatius indexen dues taules de *lookup*, *(Color Table, Visibility Table)* per determinar el color i la visibilitat desitjada de cada tipus de voxel. Els quatre menys significatius estan disponibles per les aplicacions.
- 32 bits: Els 24 bits més significatius guarden les coordenades x, y, z de la normal a la superfície que passa pel voxel. Els 8 bits restants indexen les dues taules de *look-up* de color i visibilitat.

L'EPA es fa mitjançant un recorregut BTF de l'arbre, segons la posició de l'observador. Per tant, la generació de la imatge depén de la complexitat de l'escena (tarda de l'ordre d'un segon per a visualitzar un cervell). S'utilitzen dos tipus de *shading*, el constant i un basat en el model empíric de Phong.

Interactivament, és possible fer mesuraments volumètrics, detecció de col.lisions, talls i anàlisi de connectivitat, així com anàlisi mitjançant un escalpel electrònic amb una resolució de 1240x1024 pixels.

Voxel Processor

L'arquitectura de Voxel Processor [GRT89] permet visualitzar superfícies 2D en temps quasi real, suportant *volume matting*, *threshold windows*, transformacions geomètriques i simulacions de cirurgia en volums 3D. És una màquina amb memòria distribuïda que només permet implementar el pipeline de visualització.

Procesa el model de voxels en paral.lel, de forma que la seva unitat de paral.lelisme són cubs de 64x64x64 voxels. Hi ha un processador per cadascun d'aquests cubs.

Mitjançant un Host (VAX) es fa l'adquisició de dades, el seu managament més complexe i la generació de dues taules de control de seqüència (SCT), que s'usaran en els processadors pel càlcul de la visibilitat en el recorregut BTF. Tota aquesta informació passa pel Bus del Host i la Unitat d'Accés, que és qui comunica el Host amb el Voxel Processor (veure figura 11.3).

El Voxel Processor consta de 64 processadors *(processor element)*, PE, de tal forma que cadascun d'ells té un dels 64 mòduls en els què està dividida la memòria. Cada PE, doncs, té associat un cub de 64x64x64 voxels, que es recorre BTF i forma una subimatge 2D (amb el valor del pixel i la profunditat associada a cada pixel).

Un processador intermig (IP) genera una imatge intermitja barrejant vuit subimatges amb l'ajut de les dues taules de control. Són necessaris vuit IP's per a processar les 64 subimatges de sortida dels PE's.

Les vuit imatges intermitjes dels IP's serveixen d'entrada a un processador únic *(Output Processor)*, OP, que genera l'imatge final de 512x512 pixels.

Així, via hardware s'implementen algorismes petits, tot i que donen poca flexibilitat a les aplicacions, resulten ser de gran velocitat, ja que es té dos nivells de paral.lelisme, un a nivell de pipeline i una altre a nivell de multiprocessador.

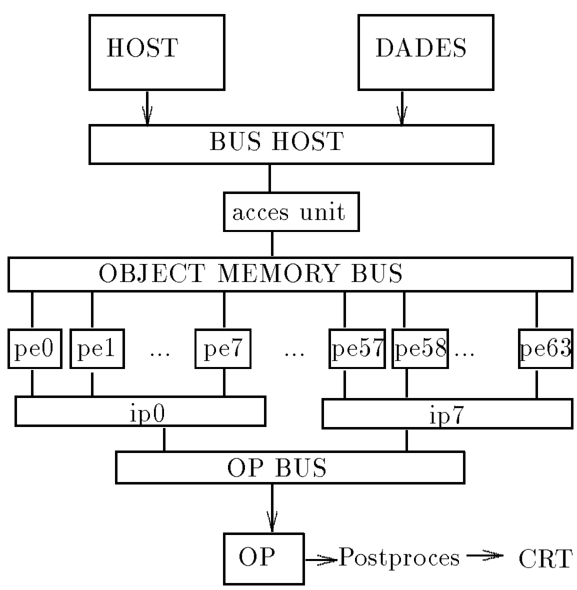

Figura 11.3 *Arquitectura Voxel Processor*

8 Conclusions

S'han analitzat els principals mètodes d'entrada, modelatge i visualització de dades volumètriques. S'han estudiat, també, diferents arquitectures orientades a accelerar aquestes tècniques. A continuació i a mode de conclusió es presenta una relació de les línies de recerca obertes en l'àmbit del modelatge i de la visualització de volum.

Les tècniques actuals en el modelatge de volum encara són lentes, tenen poca precisió i presenten problemes a l'hora de interaccionar en temps real.

Cal millorar la precisió dels dispositius de captació per tal d'obtenir unes dades més acurades. A més, és convenient estudiar la possibilitat de fusionar informació procedent de diferents dispositius d'entrada.

El problema de modelatge de volum esà poc formalitzat; no existeix una formalització matemàtica precisa d'aquests tipus d'objectes, ni s'han analitzat les especificacions dels sistemes de modelatge de volum.

Si bé el model de voxel és el més utilitzat per la seva simplicitat i la seva versatilitat, això no obstant, presenta l'inconvenient de la seva elevada ocupació de memòria. Caldria trobar models alternatius que permetin de compactar la informació. Els treballs realitzats dins d'aquesta línia estan orientats cap a l'analisi de l'utilització d'octrees i cap a l'estudi de la viabilitat de l'aplicació de tècniques de filtratge.

Una altra línia de recerca és el desenvolupament de models híbrids volum-superfície necessaris en certes aplicacions. També s'investiguen els sistemes de modelatge mixtes que suporten simultàniament models de volum i models de sòlid.

Dins de la visualització, s'investiguen mètodes que poguin ser utilitzats en temps real i facilitin la interacció de l'usuari amb el model, com per exemple tècniques de selecció interactiva de les dades a visualitzar.

Alguns autors [Lev90b] [KYC90] han desenvolupat algorismes per a escenes mixtes, formades per volums i superfícies. No tots aquests algorismes, però, permeten volums i superfícies semi-transparents. Aquesta és una línia oberta en aquest àmbit.

Per a accelerar totes les tècniques esmentades, s'estan desenvolupant arquitectures de propòsit específic i s'estan implementant algorismes paral.lels en màquines de propòsit de propòsit general.

9 Bibliografia

A continuació es presenta una enumeració d'articles referents al Modelatge i a la Visualització de Volum, alguns dels quals han estat referenciats en el texte.

[Aga84] Agard, D.A., "Optical Sectioning Microscopy", *Annual Review of Biophysics and Bioengineering*, 191-219, 1984.

[AGL89] Agate, M., Grimsdale, R.L., Lister, P.F., "The HERO Algorithm for Ray-Tracing Octrees", *Workshop Hardware EUROGRAPHICS'89*, 1989.

[AOU87] Anjyo, K., Ochi, T., Usami, Y., Kawashima, Y., "A practical Method of constructing surfaces in three-dimensional digitized space.", *The Visual Computer*, 3, 4-12, Febraury 1987.

[App67] Appel, A., "The Notion of Quantitative Invisibility and the Machine Rendering Solids", *Proceedings of the ACM National Conference*, 387-393, 1967.

[ArZ92] Argiro, V., van Zandt, W., "Voxels: Data in 3D", *Byte*, 177-182, May 1992.

[Art79] Artzy, E., "Display of three-dimensional information in computed tomography", *Computer Graphics and Image Processing*, 9, 196-198, 1979.

[AFH85] Artzy, E., Frieder, G., Herman, G.T., "The teory, design, implementation and evaluation of a three-dimensional surface detection ", *Computer Graphics and Image Processing*, 15, 1-24, Febraury 1985.

[BKR89] Bai, Z.D., Krishnaiah, P.R., Rao, C.R., Reddy, P.S., Sun, Y.N., Zhao, L.C., "Reconstrution of the Left Ventricle from Two Orthogonal Projections", *Computer Vision, Graphics and Image Processing*, 47, 165-188, August, 1989.

[BaK89] Bajcsy, R., Kovacic, S., "Multiresolution Elastic Matching", *Computer Vision, Graphics and Image Processing*, 46, 1-21, 1989.

[Bak88] Baker, H.H., "Bulding, Visualizing and Computing on Surfaces of Evolution", *IEEE Computer Graphics and applications*, 8, July 1988.

[BGS85] Barillot, C., Gibaud, B., Scarabin, J., Coatrieux, J., "3D Reconstruction of Cerebral Blood Vessels", *IEEE Computer Graphics and Applications*, 5(12), 13-19, December 1985.

[BCL90] Baronti, S., Casini, A., Lotti, F., "Variable Pyramid Structures for Image Segmentation", *Computer vision, graphics and image processing*, 49, 346-356, March 1990.

[BGP83] Bates, R.H.T., Garden, K.L., Peters, T.M., "Overview of Computerized Tomography with Emphasis on Future Developments", *Proceedings of the IEE*, 1, March 1983.

[BBO91] Becker, S.C., Barret, W.A., Olsen, D.R., "Interactive Mesurament of Three-Dimensional Objects Using a Depth Buffer and Linear Probe", *ACM Transactions on Graphics*, 10, 200-207, April 1991.

[Bid92] Bidasaria, H.B., "Defining and Rendering of Textured Objects trougth the Use of Exponential Functions", *CVGIP: Graphical, Models and Image Processing*, 52, 97-102, March 1992.

[BiS86] Bier, E.A., Sloan, K.R., "Two Part Texture Mappings", *IEEE Computer Graphics and applications*, 6, 40-53, September 1986.

[Bli82] Blinn, J.F., "Ligth Reflection Functions for simulation of clouds and dusty surfaces", *Computer Graphics*, 16, July 1982.

[BoF81] Boissonnat, J.D., Faugeras, O.D., "Triangulation of 3D Objects", *IJCAI-81, Proceedings of International Joint Conference on Artificial Intelligence*, 658-660, August 1981.

[Boi85] Boissonnat, J.D., "Surface reconstruction from planar cross section ", *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 393-397, 1985.

[BrL86] Bright, S., Laflin, S., "Shading of Solid Object Models", *Computer Graohics Forum*, 5(2), 131-138, 1986.

[Bru89] Brunet, P., "3D Structures For The Encoging Of Geometry And Internal Properties", *NATO Meeting, Santa Barbara*, December 1989.

[BrN90] Brunet, P., Navazo, I., "Solid Representation and Operation using Extended Octrees", *ACM Transactions on Graphics*, 9, 170-197, 1990.

[BJN92] Brunet, P., Juan, R., Navazo, I., "Octree Representations in solid Modeling", *Progress in Computer Graphics*, I. Zobrist and C. Sabharwal, 1, 164-215, 1992.

[CaU92] Cameron, G.G., Undrill, P.E., "Rendering Volumetric Medical Image Data on a SIMD Architecture Computer", *Proceedings of Workshop on Rendering*, Bristol 1992.

[Car84] Carpenter, L., "The A-Buffer, An antialiased hidden-surface method", *Computer Graphics*, 18 (3), 103-108, July 1984.

[Cha60] Chandrasekhar, S., "Radiative Transfer", Dover, New York, 1960.

[CCH91] Chang, L.W., Chen, H.W., Ho, J.R., "Reconstruction of 3D Medical Images: A Nonlinear Interpolation Technique for Reconstruction of 3D Medical Images", *CVGIP: Graphics models and image processing*, 53, 382-391, 1991.

[ChH88] Chen, H.H., Huang, T.S., "A Survey of Construction and Manipulation of Octrees", *Computer Vision, Graphics and Image Processing*, 43, 409-431, 1988.

[CHR85] Chen, L.S., Herman, G.T., Reynolds, R.A., Udupa, J.K., "Surface shading in the Cuberille Environment", *IEEE Computer Graphics and Applications*, 5 (12), 33-43, December 1985.

[ChS89] Chen, L.S., Sontag, M.R., "Representation, Display and Manipulation of 3D Digital Scenes and", *Computer Vision, Graphics and Image Processing*, 48, 190-216, November, 1989.

[ChS78] Christiansen, H.N., Sederberg, T.W., "Conversion of Complex Contour Line Definitions into Polygonal Element Mosaics", *ACM Computer Graphics (Proceedings '78)*, 12 (2), 187-192, August 1978.

[ClW88] Cleary, J.G., Wyvill, G., "Analysis of an algorithm for fast ray tracing using uniform space subdivision", *Tha Visual Computer*, 4, 65-83, 1988.

[CLL88] Cline, H.C., Lorensen, W.E., Ludke, S., Crawford, C.R., Teeter, B.C., "Two algorithms for the three-dimensional reconstruction of tomograms", *Medical Physics*, 15, May/Jun 1988.

[CLK90] Cline, H.C., Lorensen, W.E., Kikinis, R., Jolesz, F., "Three-Dimensional Segmentation of MR Images of the Head Using Probability and Connectivity", *Journal of Computer Assisted Tomography*, 14, 1037-1045, November/December 1990.

[CLH89] Cline, H.E., Lorensen, W.E., Herfkens, R.J., Johnson, G.A., Glover, G.H., "Vascular Morphology by Three Dimensional Magnetic Resonance", *Magnetic Resonance Imaging*, 7 (1), January 1989.

[CKB90] Cohen, D., Kaufman, A., Bakalash, R., Bergman, S., "Real-time discrete shading", *The Visual Computer*, 6 (1), 28-36, February 1990.

[CoP91] Cohen, F.S., Patel, M.A.S., "Modeling and Synthesis of Images of 3D Textured Surfaces", *CVGIP:Graphical Models and Image Processing*, 53, 501-510, November 1991.

[CPM92] Cohen, M.F., Painter, J., Mehta, M., Ma, K.L., "Volume Seedlings", *1992 Symposioum on Interactive 3D Graphics*, ACM Press, 1992.

[Col92] Collins, B., "Data Visualization", *Data Visualization*, UKKSSC 241, Gener 1992.

[Con85] Connolly, C.I., "The determination of next best views", *IEEE International Conference Robotics and Automation*, 432-435, March 1985.

[CrM92] Crawfis, R., Max, N., "Direct Volume Visualization of Three-Dimensional Vector Fields", *Proceedings of 1992 Workshop on Volume Visualization*, 55-59, 1992.

[DaH92] Danskin, J., Hanrahan, P., "Fast Algorithms for Volume Ray-tracing", *1992 Workshop on Volume Visualization*, 91-106, 1992.

[DFN91] Dehne, F., Fabri, A., Nassar, M., "Construction of d-Dimensional Hyperoctrees on a Hypercube Multiprocessor", *INRIA*, 1991.

[Dia92] Diamond, R., "Molecular Modelling in Graphics", *International Tables for Crystallography*, Kluwer Academic Publishers, Volum B, 1992.

[DoK92] Doi, A., Koide, A., "A Cell-Traverse Display Algorithm for Regular and Rectilinear 3D Grid Data", *The Journal of Visualization and Computer Animation*, 3, 129-143, 1992.

[DCH88] Drebin, R.A., Carpenter, L., Hanrahan, P., "Volume Rendering", *Computer Graphics*, 22, Agost 1988.

[Dur88] Durs, M.J., "Letters: Additional Reference to Marching Cubes", *Computer Graphics (Proceedings SIGGRAPH'88)*, 22(4), 65-74, August 1988.

[DyW87] Dyer, S., Whitman, S., "A Vectorial Scan-Line Z-Buffer Rendering Algorithm", *IEEE Computer Graphics and Applications*, 7, July 1987.

[DzE92] Dzik, S., Ezrielev, J., "Representing Surfaces with Voxels", *Computer and Graphics*, 16 (3), 295-301, 1992.

[Ear92] Earnshaw , R., "Interdisciplary techniques, toolkits and models for Scientific Visualization ", *Proceedings of the Computer Graphics Workshop on Graphics Modeling and Visualization in Science and technology* , Darmstadt, 13,14 April 1992 .

[EdM92] Edelsbrunner, H., Mucke, E.P., "Three-dimensional Alpha Shapes", *1992 Workshop on Volume Visualization*, 1992.

[EiK88] Eichmann, G., Kasparis, T., "Topologically Invariant Texture Descriptors", *Computer Vision, Graphics and Image Processing*, 41, 267-281, March 1988.

[Elv92] Elvins, T.T., "A Survey of Algorithms for Volume Visualization", *ACM Computer Graphics* , 26 (3), 194-201, August 1992.

[Far83] Farrell, E.J., "Color Display and Interactive Interpretation of Three-Dimensional Data", *IBM Journal of Research and Development*, 27 (4), 356-366, July 1983.

[FYZ85] Farrell, E.J., Yang, W.C., Zappulla, R.A., "Animated 3-D CT Imaging", *IEEE Computer Graphics and Applications*, 5, 26-32, December 1985.

[FaZ89] Farrell, E.J., Zapulla, R.A., "Three-Dimensional Data Visualization and biomedical applications", *CRC Critical Reviews in Biomedical Engineering*, 16, 1989.

[FaC89] Farrell, E.J., Christidis, Z.D., "Three-Dimensional Visualization and Display Technologies", *Proceedings of SPIE-The International Society for Optical Engineering*, 1083, January 1989.

[FAO91] Farrell, E.J., Aukrust, T., Oberhuber, J.M., "Vector and scalar field interpretation", *SPIE, Extracting Meaning from Complex Data:Processing, Display, Interaction*, 1459, 1991.

[FaD91] Faust, J.T., Dyer, D.S., "An effective data format for scientific visualzation", *Proceedings of IBM*, 1991.

[FVL86] Fellinghan, L.L., Vogel, J., Lan, C., Dev, P., "Interactive graphics and 3D modeling for surgical planning and prothesis and implant design", *Proceedings 7th. Annual Conference Computer Graphics Association*, National Computer Graphics Association, 132, 1986.

[FDM87] Fishman, E.K., Drebin, R., Magid, D., Scott, W.W., Ney, D.R., "Volumetric Rendering Techniques: Applications for 3D Imaging of the Hip", *Radiology*, 163 (6), 737-738, 1987 .

[Fly66] Flynn, M.J., "Very high-speed computing systems", *Proceedings of the IEEE*, 54, 1901-1909, 1966.

[FLN90] Foley, A., Lane, D.A., Nielson, G.M., Ramaraj, R., "Visualizing Functions Over a Sphere", *IEEE Computer Graphics and applications*, 10, 32-40, January 1990.

[FHN91] Foley, T.A., Hagen, H., Nielson, G.M., "Recent Methods for Visualizing and Modeling Unstructured Data", 1991.

[FDF90] Foley, J., Van Damm, A., Feiner, S., Hughes, J., "Interactive Computer Grasphics", *Addisson Wesley*, 1990.

[FGR85] Frieder, D., Gordon, D., Reynolds, R.A., "Back-to-front display of voxel based objects", *IEEE Computer Graphics and applications*, 5, 52, January 1985.

[FHM85] Frieder, G., Herman, G.T., Meyer, C., Udupa, J., "Large Software Problems for Small Computers: An Exemple from Medical Imaging", *IEEE Software*, 2 (5), 37-47, Setember 1985.

[Fru91] Fruhauf, M., "Combining Volumen Rendering with Line and Surface Rendering", *Eurographics'91*, 1991.

[FAG83] Fuchs, H., Abram, G., Grant, E., "Near Real-Time Shaded Display of Rigid Objects ", *ACM Computer Graphics* , 17 (3), 65-69, July 1983.

[FKU77] Fuchs, H., Kedem, Z.M., Uselton, S.P., "Optimal Surface Reconstruction from planar contours", *Communications ACM*, 20, 693, 1977.

[FGH85] Fuchs, H., Goldfeather, J., Hultquist, J., Spach, J.P., Austin, J.D., Brooks, F.P., Eyles, J.G., Poulton, J., "Fast spheres, shadows, textures, transparencies and images enhacements in Pixel-Planes", *Computer Graphics*, 19 (3), 111-120, July 1985.

[FLP89] Fuchs, H., Levoy, M., Pizer, S.M., "Interactive Visualization of 3D Medical Data", *IEEE Computer*, 46-51, August 1989.

[GaM69] Galimberti, R., Montanari, U., "An Algorithm for Hidden-Line Enumeration", *Communications of the ACM*, 12 (4), 206, April 1969.

[GaH91] Galyean, T.A., Hughes, J.F., "Sculpting: An Interactive Volumetric Modeling Technique", *Computer Graphics*, 25, 267-274, July 1991.

[GaD82] Ganapathy, S., Dennehy, T.G., "A new general triangulation method for planar contours", *Proceedings SIGGRAPH'82*, 69-75, 1982.

[GWW86] Gargantini, I., Walsh, T.R., Wu, O.L., "Viewing Transformations of Voxel-Based Objects via Linear Octrees", *IEEE Computer Graphics and applications*, 6, 12-21, October 1986.

[Gar90] Garrity, M.P., "Raytracing Irregular Volume Data", *ACM Computer Graphics*, 24 (5), November 1990.

[GeS87] Gelberg, L.M., Stephenson, T.P., "Supercomputing and Graphics in the Earth and Planetary Science", *IEEE Computer Graphics and applications*, 7, 26-33, July 1987.

[GeW92] Van Gelder, A., Wilhems, J., "Interactive Animated Visualization of Flow Fields", *Proceedings of 1992 Workshop on Volume Visualization*, 47-54, 1992.

[Gie92] Giertsen, C., "Creative Parameter Selection for Volume Visualization", *The Journal if Visualization and Computer Animation*, 3, 1-11, 1992.

[GMO89] Goadsell, D.S., Mian, S., Olson, A.J., "Rendering volumetric data in molecular systems", *Journal of Molecular Graphics*, 7, March 1989.

[GRT89] Goldwasser, S.M., Reynolds, R.A., Talton, D.A., Walsh, E.S., "High Performance Graphics Processors for Medical Imaging Applications", *Parallel Processing for Computing Vision and Display*, P.M. Dew, R.A. Earnshaw and T.R Heywood (eds), Addison-Wesley, 461-470, 1989.

[GoO89] Goodsell, D.S., Olson, A.S., "Molecular Applications of Volume Rendering and 3D-Texture Maps", *Proceedings of the Chapell Hill Workshop on Volume Visualization*, University of North Carolina Press, Chapell Hill, Inc., 27-32, May 1989.

[GTG84] Goral, C., Torrance, K., Greenberg, D., Battaile, B., "Modelling the interaction of light between diffuse surfaces", *ACM Computer Graphics*, 18 (3), 212-222, July 1984.

[GoR85] Gordon, D., Reynolds, R.A., "Image Space Shading of Three-Dimensional Objects", *Computer Vision, Graphics and Image Processing*, 29 , 361-376, 1985.

[GoU85] Gordon, D., Udupa, J.K., "Fast surface tracking in three-dimesional binary images", *Computer, Vision and Image Processing*, 29, 361-376, 1985.

[Gou71] Gouraud, H., "Continous shading of curved surfaces", *IEEE Transactions on computers*, 20 (6), 623-629, 1971.

[GRW87] Granholm, J.W., Robertson, D.D., Walker, P.S., Nelson, P.C., "Computer Design of Custom Femoral Stem Protheses", *IEEE Computer Graphics and applications*, 7, 26-35, Febraury 1987.

[Gre89] Greene, N., "Voxel Space Automata: Modeling with Stochastic growth Processes in Voxel Space", *Computer Graphics*, 23, July 1989.

[Hal89] Hall, R., "Illumination and color in Computer Generated Imagery", *Monographs in Visual Communication*, D. Rogers, Springer Verlag, 1989.

[HRY78] Harris, L.D., Robb, R.A., Yuen, T.S., Ritman, E.L., "Non-invasive numerical dissection and display of anatomic structures using computerized X-ray tomography", *Recent and Futture Developments in Medical Imaging*, Proc. Soc. Photo Opt. Instrum. Eng., 152, 1978.

[Hec86] Heckbert, P.S., "Survey of Texture Mapping", *IEEE Computer Graphics and applications*, 6, 56-67, November 1986.

[HeR85] Heffernan, P.B., Robb, R.A., "A new method for shaded surface display of biological and medical images", *IEEE Transactions in Medical Imaging*, 4, 26-38, 1985.

[HeH89] Helman, J., Hesselink, L., "Representation and Display of Vector Field Topology in Fluid Flow Data Sets", *Computer*, 27-36, August 1989.

[HeL79] Herman, G.T., Liu, H.K., "Three-Dimensional Display of Human Organs from Computed Tomograms", *Computer Graphics and Image Processing*, 1-21, 1979.

[HeU81] Herman, G.T., Udupa, J.K., "Display of Three-Dimensional Discrete Surfaces", *Proceedings of SPIE*, 283 , 90-97, 1981.

[HeU83] Herman, G.T., Udupa, J.K., "Display of Three-Dimensional Digital Images", *IEEE Computer Graphics and Applications*, 3 (5) , 39-46, 1983.

[HeW83] Herman, G.T., Webster, D., "A topological proof of a surface tracking algorithm ", *Computer, Vision and Image Processing*, 23, 162-177, 1983.

[Her83] Herman, G.T., "The Special Issue on Computerized Tomography", *Proceedings of the IEEE*, 1, March 1983.

[HTU88] Herman, G.T., Trivedi, S.S., Udupa, J.K., "Manipulation of 3D Imagery", *Volumen Visualization, A.Kaufman*, 1988.

[HZB92] Herman, G.T., Zheng, J., Bulcholtz, C.A., "Shape-based Interpolation", *IEEE Computer Graphics and Applications*, 69-79, May 1992.

[HiS89a] Hibbard, W., Santek, D., "Visualizing Large Data Sets in Earth Sciences", *IEEE Computer*, 22, 53-57, 1989.

[HiS89b] Hibbard, W., Santek, D., "Interactivity is the key", *Proceedings of Chapel Hill Volume Visualization Workshop*, 39-43, May 1989.

[HBP90a] Hohne, K.H., Bomans, M., Pommert, A., Riemer, M., Schierds, C., Tiede, U., Wiebecke, G., "3D Visualization of Tomographic Volume Data Using the Generalized Voxel-Model", *The Visual Computer*, 6(1), 28-36, February 1990.

[HBP90b] Hohne, K.H., Bomans, M., Pommert, A., Riemer, M., Tiede, U., Wiebecke, G., "Rendering Tomographic Volume Data Using: adequacy of methods for diferents modelitis and Organs", *3D Imaging in Medicine*, K.H. Hohne, H. Fuchs and S. Pizer, Springer Verlag, 197-216, 1990.

[HoB90] Hottier, F., Billion, A.C., "3D Echography: Status and Perspective", *3D Imaging in Medicine: Algorithms, Systems, Applications*, K.H. Hoene, H. Fuchs and S.M. Pizer (eds.), Springer-Verlag, Berlin, 21-41, 1990.

[HsS89] Hsiao, J.Y., Sawchuck, A.A., "Unsupervised Textured Image Segmentation Using Feature Smoothing", *Computer Vision, Graphics and Image Processing*, 48, 1-21, October, 1989.

[Ina91] Inakage, M., "Volume Tracing Soft Objects", *Proceedings IBM*, 1991.

[Jai81] Jain, A.K., "Advances in Mathematical Models for Image Processing", *Proceedings of the IEEE*, May 1981.

[Jun92] Jung, V., "Visualization of Flow Simulation Data in Environmental Modelling", *Proceedings of the Computer Graphics Workshop on Graphics Modeling and Visualization in Science and technology* , Darmstadt, 13,14 April 1992 .

[KaH84] Kajiya, J., Von Herzen, B., "Ray Tracing Densities", *ACM Computer Graphics*, 18 (3), 166-174, July 1984.

[KaK89] Kajiya, J., Kay, T., "Rendering fur with three dimensional textures", *ACM Computer Graphics*, 23(3), 271-277, July 1989.

[KaS89] Kartikeyan, B., Sarkar, A., "A Unified Approach for Image Segmentation Using Exact Statistics", *Computer Vision, Graphics and Image Processing*, 48, 217-229, November 1989.

[KaS91] Kartikeyan, B., Sakar, A., "An Identification Approach for 2D Autoregressive Models in Describing", *CVGIP:Graphical Models and Image Processing*, 53, 121-131, March 1991.

[KaS86] Kaufman, A., Shimony, E., "3D Scan-Conversion Algorithms for Voxel-Based Graphics", *Proceedings of ACM Workshop on Interactive 3D GRaphics*, New York, 45-75, October 1986.

[Kau87] Kaufman, A., "Efficient Algorithms for 3D Scan-Conversion Algorithms of Parametric Curves, Surface and Volumes", *Computer Graphics*, 21, 171-179, July 1987.

[Kau88a] Kaufman, A., "Efficient Algorithms of 3D Scan-Conversion Algorithms of Polygons", *Computers and Graphics*, 12, 213-219, 1988.

[Kau88b] Kaufman, A., "The CUBE Workstation - a 3D voxel-based graphics environment", *The Visual Computer*, 4, 210-221, October 1988.

[KaB88] Kaufman, A., Bakalash, R., "Memory and Processing for 3D Voxel-Based Imagery", *IEEE Computer Graphics and applications*, 8, 10-23, November 1988.

[KCY90] Kaufman, A., Cohen, D., Yagel, R., "Volumetric Shading Techniques", *Volume Visualization*, A. Kaufman, IEEE Computer Society Press Tutorial, 169-173, 1990.

[KYC90] Kaufman, A., Yagel, R., Cohen, D., "Intermixing Surface and Volume Rendering", *3D Imaging in Medicine: Algorithms, Systems and Applications*, K.H. Hohne, H. Fuchs, S.M. Pizer, Springer-Verlag, 217-228, 1990.

[Kau90a] Kaufman, A., "Volume Visualization", IEEE Computer Society Press Tutorial, August 1990.

[Kau90b] Kaufman, A., "Introduction to Volume Visualization", *Volume Visualization*, IEEE Computer Society Press Tutorial , 1-18, August 1990.

[KYB90] Kaufman, A., Yagel, R., Bakalash, R., Spector, I., "Volume Visualization in Cell Biology", *Proceedings Visualization*, IEEE Computer Society Press. Los Alamitos, California, 161-1676, October 1990.

[KSF88] Kehtarnavaz, N., Simar, L.R., Figuereido, R.J.P., "A Syntactic/Semantic Technique for Surface Reconstruction from Cross-Sectional Contours", *Computer Vision, Graphics and Image Processing*, 42, 399-409, June 1988.

[Kep75] Keppel, E., "Approximating Complex Surfaces by Triangulation of Contours", *IBM Journal of Research and Development*, 19 (1), 2-11, January 1975.

[Ker90] Kerlick , G.D., "ISOLEV: a level surface cutting-plane program for fluid flow data", *Extracting Meaning from Complex Data: Processing, Display, Interaction*, Proceeding of SPIE Conference, 1259, 2-13, 14-16 February 1990.

[KKF89] Kluksdahl, N.C., Kriman, A.M., Ferrry, D.K., "Transport in Semiconductors", *Computer*, 60-66, August 1989.

[KDK86] Koide, A., Doi, A., Kajioka, K., "Polyhedral approximation approach to molecular orbital graphics", *Journal of Molecular Graphics*, 4, 149-155, September 1986.

[Kon89] Kong, T.Y., "Survey: Digital Topology: Introduction and Survey", *Computer vision, graphics and image processing*, 48, 357-393, December 1989.

[Kov89] Kovalesky, V.A., "Finite Topology as Applied to Image Analysis", *Computer Vision, Graphics and Image Processing*, 46, 141-161, May 1989.

[KKG90] Kramer, D.M., Kaufman, L., Guzman, R.J., Hawryzsko, C., "A General Algorithm for Oblique Image Reconstruction", *IEEE Computer Graphics and Applications*, 10, 62-67, March 1990.

[Kre90] Kreuger, W., "Volume Rendering and Data Feature Enhacement", *Proceedings of San Diego Workshop on Volume Visualization*, 20-26, November 1990.

[Kro85] Kronlof, K., "A viewing pipeline for discrete solid modeling", *The Visual Computer*, 1, 24-36, 1985.

[Lal89] Lalvani, H., "Coding and generating complex periodic patterns", *The Visual Computer*, 5, 180-202, 1989.

[LCW80] Lane, J., Carpenter, L., Whitted, J., Blinn, J., "Scan-Line methods for displaying parametrically defined surfaces", *ACM Computer Graphics*, 23, January 1980.

[LaH91] Laur, D., Hanrahan, P., "Hierarchical Splatting: A Progressive Refinemet Algorithm for Volume Rendering", *Computer Graphics*, 25(4), 285-318, July 1991.

[LSK89] Lee, M.W., Satoh, T., Kunii, T.L., "Comparative Operations in Solid Modeling", *Workstations for Experiments*, July 1989.

[LeR82] Lee, Y.T., Requicha, A.A.G., "Algorithms for computing the volume and other integral properties of solids II. A family of algorithms based on representation conversion and cellular approximation", *Communications of ACM*, 25, 642-650, 1982.

[LMP89] Leith, A., Marko, M., Parsons, D., "Computer Graphics for Cellular Reconstruction", *IEEE Computer Graphics and applications*, 9, 16-24, September 1989.

[LeP84] Levinthal, A., Porter, T., "Chap - A SIMD Graphic Processor", 18 (3), 77-82, July 1984.

[Lev88] Levoy, M., "Display of Surfaces from Volume Data", *IEEE Computer Graphics and applications*, 8, 29-37, May 1988.

[Lev90a] Levoy, M., "Efficient Ray Tracing of Volume Data", *ACM Transactions on Graphics*, 9, 245-261, July 1990.

[Lev90b] Levoy, M., "A Hybrid Ray Tracer for Rendering Polygon and Volume Data", *IEEE Computer Graphics and applications*, 10, 33-40, March 1990.

[Lev91] Levoy, M., "Viewing Algorithms", *Volume Visualization*, A. Kaufman, 1991.

[Lev92] Levoy, M., "Volume Rendering using the Fourier Projection-Slice Theorem", *Graphics Interface'92*, 61-69, 1992.

[Lew83] Lewitt, R.M., "Reconstruction Algorithms: Transform Methods", *Proceedings of the IEEE*, 1, March 1983.

[LiT90] Liang, P., Todhunter, J.S., "Representation and Recognition of Surface Shapes in Ranges Images: A Differential Geometry Approach", *Computer Vision, Graphics and Image Processing*, 52, 78-109, October 1990.

[LiC89] Lin , W.C., Chen , S.Y., "A New Surface Interpolation Technique for Reconstructing 3D Objects from Serial Cross-Sections", *Computer Vision, Graphics and Image Processing*, 48, 124-143, 1989.

[LCC89] Lin, W.C., Chen, S.Y., Chen, C.T., "A New Surface Interpolation Technique for Reconstructing 3D Objects from Serial Cross Sections", *Computer Vision, Graphics and Image Processing*, 48, 1-21, October 1989.

[Liu77] Liu, H.K., "Two- and three-dimensional boundary detection", *Computer Graphics and image processing*, 6, 126, 1977.

[LLL89] Long, M.B., Lyons, K., Lam, J.K., "Adquisition and Representation of 2D and 3D Data from Turbulent Flows and Flames", *Computer*, 22, 39-45, August 1989.

[LoC87] Lorensen, W.E., Cline, H.E., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *Computer Graphics*, 21, July 1987.

[LoC91] Lorensen, W.E., Cline, H.E., "Volume Modeling", *Course Notes of SIGGRAPH '91*, 7, 31-43, 1991.

[MCP91] Ma, K.L., Cohen, M.F., Painter, J.S., "Volume Seeds: A volume exploration technique", *The Journal of Visualization and Computer Animation*, 2, 135-190, May 1991.

[MaG87] MacShan, M.D.L., Glickman, A.S., "Color Displays for medical imaging", *Color and the Computer*, J. Durret, Academic Press, 205-220 , 1987.

[MaT87] Magnenat, N., Thalmann, D., "An Indexed Bibliography on Image Synthesis", *IEEE Computer Graphics and applications*, 7, 27-38, August 1987.

[MNR91] Maguire, G.Q., Noz, M.E., Rusinek, H., Jaeger, J., Kramer, E.L., Sanger, J.J., Smith, G., "Graphics Applied to Medical Image Registration", *IEEE Computer Graphics and Applications*, 20-27, March 1991.

[Mah72] Mahl, R., "Visible surface algorithms for quadric patches", *IEEE Transactions on Computers*, C-21, January 1972.

[MFK90] Mao, X., Fujishiro, I., Kunii, T.L., "A Translucent Display Algorithm for G-octree Represented Grey-Scale", *The journal of visualization and Computer animation*, 1, 22-25, 1990.

[MKD90] Marshall , R., Kempf , J., Dyer , S., "Visualization Methods and Simulation Steering for a 3D Turbulence Model of Lake Erie ", *ACM Computer and Graphics* , 89-97 , 1990.

[Mat89] Matsuyama, T., "Expert Systems for Image Processing: Knowledge-Based Composition of", *Computer Vision, Graphics and Image Processing*, 48, 22-49, October, 1989.

[MHC90] Max, N., Hanrahan, P., Crawfis, R., "Area and Volume Coherence for efficient Visualization of 3D Scalar Functions", *Computer Graphics, Proceedings of San Diego Workshop on Volume Visualization*, 25, 27-33, November 1990.

[MaH76] Mazziotta, J.C., Huang, H.K., "THREAD (3D Reconstruction and Display) with Biomedical Applications in Neuron Ultraestructure and Computerized Tomography", *American Federation of Information Processing Societies*, 45, 241-250, 1976.

[MDB87] McCormick, B.H., Defanti, T.A., Brown, M.D., "Visualization in Scientific Computing", *Computer Graphics*, 21 (5), 1987.

[Mea82] Meagher, D., "Geometric Modeling using Octrees Encoding", *Computer Graphics and Image Processing*, 19, 129-147, 1982.

[Mea84b] Meagher, D.J., "A New Mathematics for Solids Processing", *Computer Graphics World*, October 1984.

[Mea85] Meagher, D.J., "Applying Solids Processing Methods to Medical Planning", *Proceedings of NCGA '85*, 101-109, April 1985.

[Mea82] Meagher, D.J., "Applying Solids Processing Methods to Medical Planning", *Proceedings of NCGA '85*, 101-109, April 1985.

[MeS92] Meuer, H-W., Stromaier, E., "Present parallel computers-concepts and architectures", *Informationstechnik*, 34, 1992.

[MiK92] Miyazawa, T., Koyamada, K., "A High-speed Integrated Renderer for Interpreting Multiple 3D Volume Data", *The Journal of Visualization and Computer Animation*, 3 , 65-83, 1992.

[MBL91] Miller, J.V., Breen, D.E., Lorensen, W.E., O'Bara, R., Wozny, M.J., "Geometrically Deformed Models: A Method for extracting Closed Geometric Models from Volume Data.", *Computer Graphics*, 25, July 1991.

[Mok88] Mokrzcycki, W., "Algorithms of Discretitatization of Algebraic Spatial Curves on Homogeneous Cubical Grids", *Computers and Graphics*, 12, 477-487, 1988.

[MPS92] Montani, C., Perego, R., Scopigno, R., "Parallel Volume Visualization on a Hypercube Architecture", *Proceedings of 1992 Workshop on Volume Visualization*, 9-16, 1992.

[Mor85] Morffew, A.J., "Protein Modelling using Computer Graphics", *Advances in Biotechnological Processes*, 5, 31-58, 1985.

[Mur91] Muraki, S., "Volumetric Shape Description of Range Data using "Blobby Model"", *Computer Graphics*, 25, 227-235, July 1991.

[Mur92] Muraki, S., "Approximation and Rendering of Volume Data Using Wavelet Transforms", *Submitted to Visualization'92*, 1992.

[NOB88] Naidich, T.P, Osborn, R.E., Baner, B., Naidich, M.J., "Median Cleft Face Syndrome: MR and CT Data from 11 Children", *Journal of Computer Assisted Tomography*, 12 (1), 57-64, 1988.

[NHY85] Nakamae, E., Harada, K., Yasuda, M., Sato, A., "Stereographic display of an object consisting of multi-surfaces", *Trans Information Processing Society of Japan*, 26, 181-188, 1985.

[Nav86] Navazo, I., "Contribucio a les tecniques de modelat geometric d'objectes poliedrics usant la codificacio amb arbres octals", 1986.

[NAB86] Navazo, I., Ayala, D., Brunet, P., "A Geometric Modeller Based on the Exact Octree Representation of Polyhedra", *Computer Graphics Forum*, 5, 91-104, 1986.

[NFM90] Ney, D.R., Fishman, E.K., Magid, D., "Volumetric Rendering of Computed Tomography Data: Principles and Techniques", *IEEE Computer Graphics and applications*, 10, 24-32, March 1990.

[NSR90] Nielson, G.M., Shriver, B., Rosenblum, L.J., *Visualization in Scientific Computing*, IEEE Computer Society Press, 1990.

[NiV90] Nielson, G.M., Voegele, K., "An annotated bibliography of Scientific Visualization. Part 1", *The Journal of Visualization and Computer Animation*, 1, 59-65, Febraury 1990.

[NiV91] Nielson, G.M., Voegele, K., "An annotated bibliography of Scientific Visualization. Part 2", *The Journal of Visualization and Computer Animation*, 2, 2-8, 1991.

[NFH91] Nielson, G.M., Foley, T.A., Hamman, B., Lane, D., "Visualizing and Modeling Scattered Multivariate Data", *IEEE Computer Graphics and Applications*, 47-55, May 1991.

[Nie91] Nielson, G.M., "A Perspective on Visualization in Scientific and Engineering Computation", *IEEE Computer, Special 4th Anniv. Issue of The Computer Society*, September 1991.

[NiH92] Ning, P., Hesselink, L., "Vector Quantization for Volume Rendering", *Proceedings of 1992 Workshop on Volume Visualization*, 69-74, 1992.

[NiL91] Nitzberg, B., Lo, V., "Distributed shared memory: a survey of issues and algorithms", *IEEE Computers*, 24 (8), 52-59, 1991.

[Nov92] Novak, M.M., "Multivariate Modelling of the Dynnamics of Charged Particles", *Proceedings of the Computer Graphics Workshop on Graphics Modeling and Visualization in Science and technology* , Darmstadt, 13,14 April 1992 .

[NoA92] Novins, K., Arvo, J., "Controlled Precision Volume Integration", *1992 Workshop on Volume Visualization*, 1992.

[ORo81] O'Rourke, J., "Polyhedra of Minimal Area as 3D Object Models", *IJCAI-81, Proceedings of International Joint Conference on Artificial Intelligence*, 664-666, August 1981.

[OaC90] Oakley, J.P., Cunningham, M.J., "A Function Space Model for Digital Image Sampling and its Application", *Computer Vision, graphics and image processing*, 49, 171-197, February 1990.

[Pal92] Palmer, T.C., "A Language for Molecular Visualization", *IEEE Computer Graphics and Applications*, 23-32, May 1992.

[PAS93] Pasko, A.A., Adzhiev, V.D., Sourin, A.I., "Multidimensional Geometric Modelling System Based on a Function Representation of Objects: Concepts and Specification", *Submitted Eurographics'93*, 1993.

[PaT90] Payne, B.A., Toga, A.W., "Surface Mapping Brain Function on 3D Models", *IEEE Computer Graphics and Applications*, Setember 1990.

[PeH89] Perlin, K., Hoffert, E.M., "Hypertexture", *Computer Graphics*, 23, July 1989.

[PeZ91] Perlin, K., Zhu, B., "Surflets", *Proceedings IBM*, 1991.

[Phi92] Phillips , D., "Picture this X", *Computer Graphics World*, 43-52, September 1992.

[Pho75] Phong, B., "Illumination of Computed Generated Images", *Communications of the ACM*, 18 (6), 311-317, 1975.

[PFM86] Pizer, S.M., Fuchs, H., Mosher, C., Lifshitz, L., Abraham, G.D., Ramanathan, S., Whitney, B.T., Roseman, J.G., Staab, E.V., Chaney, E.L., Sherouse, G., "3D shaded graphics in radiotherapy and diagnostic imaging", *Proceedings of Annual Conference National Computer Graphics Association*, 107, 1986.

[PiF87] Pizer, S.M., Fuchs, H., "Three dimensional image presentation techniques in medical imaging", *Proceedings Int. Symp. Comput. Assist. Radiol.*, H.U. Lembke, M.L. Rhodes Eds. Springer-Verlag, 589, 1987.

[PoD88] Porter, T., Duff, T., "Compositing Digital Images", *ACM Computer Graphics*, 22 (4), 65-74, August 1988.

[Pot85] Potmesil, M., "Generating octree models of 3D objects frrom their silhouettes in a sequence of images", *ATT Bell Laboratories*, October 1985.

[PoH89] Potmesil, M., Hoffert, E.M., "The Pixel Machine: A Parallel Image Computer", *Computer Graphics*, 23 (3), 69-78, July 1989.

[PuB92] Puig, A., Brunet, P., "Modelado de solidos no homogeneos mediante octrees. Aplicacion al modelado de yacimientos.", *CEIG* , Junio 1992.

[PuC86] Purvis, D.G., Culberson, C., "On the Graphical Display of Molecular Electrostatic Force-Fields and Gradients of the Electron Density", *Journal of Molecular Graphics*, 4, 89-92, June 1986.

[RaS91] Rao, A.R., Schunck, B.G., "Computing Oriented Texture Fields", *CVGIP: Graphical Models and Image Processing*, 53, 157-185, March 1991.

[Ree83] Reeves, W.T., "Particle Systems - A Technique for Modeling a Class of Fuzzy Objects", *Computer Graphics*, 17, 356-376, July 1983.

[ReB85] Reeves, W.T., Blau, R., "Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems", *Proceedings of SIGGRAPH*, 19 (3), 313-322, 1985.

[Rei89] Reilly, P., "Data Visualization in Archaeology", *IBM System Journal*, 28 (4), 1989.

[ReV82] Requicha, A., Voelker, H., "Solid Modelling: A Historical Summary and Comtemporary Assesment", *IEEE Computer Graphics and applications*, 9-22, 1982.

[Req80] Requicha, A., "Representations for Rigid Solids: Theory, Methods and Systems", *Computing Surveys*, 12 (4), December 1980.

[RKR87] Rhodes, M., Kuo, Y.M., Rothman, S.L.G., Woznick, C., "An application of Computer Graphics and Networks to Anatomic Model", *IEEE Computer Graphics and applications*, 7, 12-25, February 1987.

[Rho90] Rhodes, M., "Computer graphics in medicine", *IEEE Computer Graphics and applications*, 10, 20-23, March 1990.

[Rho91] Rhodes , M., "Computer Graphics in Medicine: The Past Decade ", *IEEE Computers Graphics and applications* , 11 , 52-54 , January 1991 .

[RVB90] Richter, R., Vos, J.B., Battaro, A., Gavrilakis, S., "Visualization of Flow Simulation", *Scientific Visualization and Graphics Simulation*, D. Thalmann, editor, 161-171, 1990.

[Ris92] Rist, U., "Visualization on Volumetric Flow-Field Data", *Proceedings of the Computer Graphics Workshop on Graphics Modeling and Visualization in Science and technology* , Darmstadt, 13,14 April 1992 .

[Rob85] Robb, R.A., "Three-Dimensional Biomedical Imaging", *CRC Press, Inc, Boca Raton, Fla*, 1, 1985.

[RoH90] Robb, R.A., Hanson, D.P., "ANALYZE: A software system for biomedical image analysis.", *Proceedings of 1st Conference on Visualization in Biomedical Computing*, 507-518, May 1990.

[Ros89a] Rosemblum, L.J., "Visualization of Experimental Data at the Naval Research Lavoratory", *Computer*, 95-101, August 1989.

[Ros89b] Rosenblum, L.J., "Visualizing Oceanographic Data", *IEEE Computer Graphics and applications*, 9, 14-20, May 1989.

[RoN91] Rosemblum, L.J., Nielson, G.M., "Guest Editors'Introduction", *IEEE Computer Graphics and Applications*, 15-17, May 1991.

[Ros81] Rosenfeld, A., "Image Pattern Recognition", *Proceedings of the IEEE*, May 1981.

[Ros90] Rosenfeld, A., "Image Analysis and Computer Vision: 1989", *Computer Vision, graphics and image processing*, 50, 188-240, August 1990.

[Ros89c] Rossignac, J.R., "Considerations on Interactive Rendering of Four-Dimensional Volumes", *Proceedings of Chapel Hill Workshop on Volume Visualization*, C. Upson, editor, 67-76, May 1989.

[Sab88] Sabella, P., "A Rendering Algorithm for Visualizing 3D Scalar Fields", *Computer Graphics*, 22, August 1988.

[SaC89] Sabella, P., Carlbom, I., "An Object-Oriented Approach to the Solid Modeling of Empirical Data", *IEEE Computer Graphics and applications*, 9, 24-35, September 1989.

[Sam90] Samet, H., "Applications of Spatial Data Structures", *Computer Graphics, Image Processing and GIS*, Addisson-Wesley, 1990.

[Sam89] Samet, H., "Implementing Ray Tracing with Octrees and Neighbor Finding", *ACM Computer and Graphics*, 13, 445-460, 1989.

[SaP89] Sandness, G., Prusinkiewicz, P., "Koch Curves as Attractors and Repel", *IEEE Computer*, 22, 53-57, 1989.

[ScM86] Schwartz, E.L., Merker, B., "Computer-Aided Neuroanatomy: Differential Geometry of Cortical Surfaces and an Optimal Flattening Algorithm", *IEEE Computer Graphics and applications*, 6, 36-44, March 1986.

[SMW88] Schwartz, E.L., Merker, B., Wolfson, E., Shaw, A., "Computational Neuroscience: Applications of Computers Graphics and Image Processing to 2D and 3D modeling of the funcional architecture of Visual Cortex.", *IEEE Computer Graphics and applications*, 8, July 1988.

[She87] Sheriff, R.E., "Geophysical Methods", Prentice Hall, 1987.

[ShN89] Shirley, P., Neeman, H., "Volume Visualization at the Center for Supercomputer Research and Development", *Proceedings of the Chapell Hill Workshop on Volume Visualization*, University of North Carolina Press, Chapell Hill, Inc., 17-20, May 1989.

[SpK90] Speray, D., Kennon, S., "Volume Probes: Interactive Data Exploration on Arbitrary Grids", *ACM Computer Graphics*, 69-76, 1990.

[SYM92] Stredney, D., Yagel, R., May, S.F., Torello, M., "Supercomputer Assisted Brain Visualization with an Extended Ray-Tracer", *Workshop on Volume Visualization, Boston*, October 1992.

[SFF89] Stytz, M.R., Frieder, G., Frieder, O., "Three Dimensional Medical Imaging: Algorithms and Computer Systems", *ACM Computer Surveys*, 23, December 1989.

[SuG78] Sunguroff, A., Greenberg, D., "Computer generated images for medical applications", *Computer graphics*, 12, 196-202, 1978.

[SSS74] Sutherland, I., Sproull, R., Shumacker, R., "A Characterization of Ten Hidden-Surface Algorithms", *ACM Computing Surveys*, 6, 1-55, 1974.

[TaD88] Tam, Y.W., Davis, W.A., "Display of 3D Medical Images", *Proceeding of Graphics Interfaces*, 78-86, June 1988.

[THR87] Tiede, U., Hohne, K., Riemer, M., "Comparison of surface rendering techniques for 3D tomographic objects", *Computer Assisted Radiology*, U. Lemke, Springer-Verlag, 599-610, 1987.

[THB90] Tiede, U., Heinz-Hoehne, K., Bomans, M., Pommert, A., Riemer, M., Wiebecke, G., "Investigation of Medical 3D-Rendering Algorithms", *IEEE Computer Graphics and Applications*, 10, 41-53, March 1990.

[ToT90] Toennies, K.D., Tronnier, U., "3D Modeling Using An Extended Cell Enumeration Representation", *Computer Graphics*, 24 (5), 13-20, November 1990.

[Tre92] Treinish, L.A., "Correlative Data Analysis in the Earth Sciences", *IEEE Computer Graphics and Applications*, 10-12, May 1992.

[TuT84] Tuy, H.K., Tuy, L., "Direct 2D Display of 3D objects", *IEEE Computer Graphics and applications*, 4, 29, October 1984.

[URB90] Udupa, J.K., Raya, S.P., Barret, W.A., "A PC-based 3D imaging system for biomedical data", *Proceedings of 1st Conference on Visualization in Biomedical Computing*, 295-303, May 1990.

[Ups86] Upson, C., "The Visual Simulation of Amorphous Phenomena", *Visual Computer*, 2, 321-326, September 1986.

[UpK88] Upson, C., Keeler, M., "V-Buffer: Visible Volume Rendering", *Computer Graphics*, 22, 59-64, August 1988.

[UFL89] Upson, C., Faulhaber, T., Laidlaw, D., Schlegel, D., Vroom, J., Gurwitz, R., van Dam, A., "The Application Visualization System: A Computational Environment for Scientific Visualization", *IEEE Computer Graphics and applications*, 9, 30-41, July 1989.

[VPR92] Vandermeulen, D., Plets, P., Ramakers, S., Suetens, P., Maedral, G., "Integrated Visualization of Brain Anatomy and Cerebral Blood Vessels", *Proceedings of 1992 Workshop on Volume Visualization*, 55-60, 1992.

[VMG83] Vannier, M.W., Marsh, J.L., Gado, M.H., "Three-dimesional display of intracanial soft-tissue structures", *Am. J. Neuroradiol.*, 4, 520, 1983.

[VMW83] Vannier, M.W., Marsh, J.L., Warren, J.O., "Three dimensional computer graphics for craniofacial surgical planning and evaluation ", *ACM Computer Graphics*, 17 (3), 263-273, July 1983.

[VeP77] Veen, A., Peachey, L.D., "TROTS: A Computer Graphics System Three-Dimesional Reconstruction from Serial Sections", *Computer and Graphics*, 2, 135-150, 1977.

[VeA88] Veenstra, J., Ahuja, N., "Line Drawings of Octree-Represented Objects", *ACM Transactions on Graphics*, 7 (1), 61-75, January 1988.

[WLC88] Walker, M., Lo, R.S., Cheng, S.F., "Hidden Line Detection in Polytree Representations", *ACM Computer and Graphics*, 12, 65-69, 1988.

[Wat87] Watkins, H.K., "Graphics in Reservoir Simulation", *Computer Graphics Forum*, 6, 111-118, 1987.

[Web90] Webber, R.E., "Ray-tracing voxel data via biquadratic local surface interpolation", *The Visual Computer*, 6 (1), 8-15, February 1990.

[WeC71] Wein, M., Castelman, K.R., "Reconstructing 3D Specimens from 2D-Section Images", *Proceedings of SPIE Conference*, 26, 131-138, May 1971.

[Wes89] Westover, L., "Interactive Volume Rendering", *Proceedings of Chappel Hill Volume Visualization Workshop*, 1989 .

[Wes90] Westover, L., "Footprint Evaluation for Volume Rendering", *Computer Graphics*, 24, August 1990 .

[Whi81] Whitted, T., "An improved illumination model for shaded display", *Proceeding of the 18th. Design Automotion Conference of IEEE*, 463-470, 1981.

[WCA90] Wilhems, J., , Challinger, , Aper, , Ramamoorthy, "Direct Volume Rendering of Curvilinear Volumes", *Computer Graphics, Proceedings of San Diego Workshop on Volume Visualization*, 25, 1990.

[WiG90] Wilhems, J., Van Gelder, A., "Topological Considerations in Isosurface Generation-Extended Abstract", *Computer Graphics, Proceedings of San Diego on Volume Visualization*, 24, 79-86, November 1990.

[Wil91] Wilhems, J., "Decisions in Volume Rendering", *Course Notes of the SIGGRAPH Tutorial: State-of-the-Art in Volume Visualization* , 8, July 1991.

[WiG91] Wilhems, J., Van Gelder, A., "A Coherent Projection Approach for Direct Volume Rendering", *ACM Computer Graphics*, 25 (4), July 1991.

[WiG92] Wilhems, J., Van Gelder, A., "Octrees for Isosurface Generation", *ACM Transactions on Graphics*, 11, 202-227, July 1992.

[WNB91] Williams, G.N., Nelson, E.L., Barnett, D.M., Parmley, K., "Visualization of molecular dynamics via ray-tracing and animation in a vectorized environment", *IBM, Journal of Research Development*, 35 (1/2), January-March 1991.

[WiM92] Williams, P.T., Max, N., "A Volume Density Optical Model", *1992 Workshop on Volume Visualization*, 1992.

[WoL88] Wolfe, R.H., Liu, C.N., "Interactive Visualization of 3D Seismic Data: A Volumetric Method", *IEEE Computer Graphics and applications*, 8, July 1988.

[WrH79] Wright, T., Humbrecht, J., "ISOSRF - An Algorithm for Plotting Iso-Valued Surfaces of a Function of Three Variables", *Computer Graphics (Proceedings of SIGGRAPH '79)*, 13, 182-189, August 1979.

[Wri72] Wright, W.V., "The Two-Dimensional Interface System for Molecular Studies", *ACM SIGPLAN Notices*, 7, 76-85, October 1972.

[WKS86] Wyvill, G., Kunii, T.L., Shirai, Y., "Space Division for Ray-Tracing in CSG", *IEEE Computer Graphics and Applications*, 1986.

[WWM87] Wyvill, G., Wyvill, B., McPheeters, C., "Solid Texturing of Soft Objects", *IEEE Computer Graphics and Applications*, 7, December 1987.

[XuL88] Xu, S.B., Lu, W.X., "Surface Reconstruction of 3D Objects in Computerized Tomography", *Computer Vision, Graphics and Image Processing*, 44, 270-279, December 1988.

[YaK92] Yagel , R., Kaufman, A., "Template-Based Volume Viewing", *Proceedings of Eurographics 1992*, 11 (3), 153-167, 1992.

[YYO92] Yasuda, T., Yokoi, S., Ohshita, H., Toriwaki, J., "3D Visualization of an ancient Egyptian Mummy", *IEEE Computer Graphics and Applications*, 13-17, May 1992.

[YaS83] Yau, M.M., Srihari, S.N., "A hierarchical data structure for multidimensional digital images", *Communications of ACM*, 26, 504-515, 1983.

[YKK91] Yla-Jaaski, J., Klein, F., Kubler, O., "Fast Direct Display of Volume Data for Medical Diagnosis", *CVGIP:Graphical Models and Image Processing*, 53, 7-18, January 1991.

[YoY89] Yokoya, N., Yamamoto, K., "Fractal-Based Analysis and Interpolation of 3D Natural Surface Shapes and their Application to Terrain Modeling", *Computer Vision, Graphics and Processing*, 46, 284-302, June 1989.

[ZaJ91] Zahlten, C., Jurgens, H., "Continuation Methods for Approximating Isovalued Complex Surfaces", *Eurographics '91*, 1991.

ÍNDEX