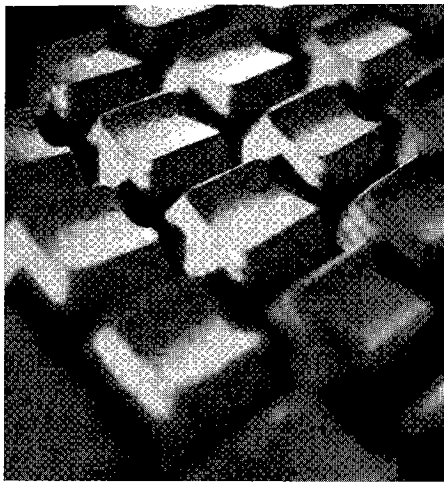


連載解説

マルチメディア ホームコンピューティングの未来

－第2回 エンタテインメントに必要な性能－

釜江尚彦 （財）イメージ情報科学研究所
kamae@tokyo.image-lab.or.jp

前号に書いたようにゲームコンピュータを進化させたマルチメディアホームコンピュータが家庭の情報化の主役になり得る一候補である．このコンピュータプラットフォームの実現可能性は要求される性能に左右される．ここでは処理能力を食う代表として3Dグラフィクスとオーディオの残響付加処理を取り上げ，必要とする性能を推定する．

【3Dグラフィクスの実時間処理】

3Dグラフィクス映像をZバッファ法による隠面消去を基本にした方法で実時間で作成する場合の処理能力を考察する．

●グラフィクスパイプライン

3Dグラフィクス映像を実時間で作成する処理の流れの基本は，図-1に示す通りである[1]．

まず3Dグラフィクスのモデルをポリゴンの集まりでモデル化しておく．このポリゴンデータはポリゴン（通常は三角形や四角形）の頂点座標，法線ベクトルと各ポリゴンをテクスチャに関連付けるアドレスからなる．このポリゴンモデルからなる3D物体（オブジェクト）それぞれを動きに応じてオブジェクト座標系で座標変換し，それを図-2に示す視点座標系に変換する．この変換は基本的にはマトリクス演算である．この1×4のマトリクスは各頂点を表すベクトルで，4×4のマトリクスが変換マトリクスである．この演算には掛け算が16回，足し算が12回必要である．より精度の高いシェーディング計算は後のレンダリング処理に属するが，照明計算は光源とポリゴンの向きとの関係が視点座標系内で求められる．それは基本的には3Dベクトルの内積を求める演算である．ついでクリッピング計算を行うが，これは図-2に示すように視点座標系で画面（スクリーン）内に入るかどうかの計算である．これには視点座標系の原点とスクリーンの四隅を結んでできる四角錐に各頂点が入っているかどうかを計算する．すなわち関心のある座標を（x, y, z）とし，スクリーンのz座標値をWとすると，Sz/WをxやyとW大小比較する．ここにSはスクリーンの大きさを表す座標値である．次の透視投影はスクリーン座標へのマッピングであり，xW/z，yW/zを座標値ごとに計算することになる．したがってジオメトリ処理の演算量は各頂点座標ごとに表-1のようになる．ここで照明計算は拡散反射と鏡面反射の両方を含めた．

●レンダリング処理

a）ラスタライゼーション

主に次の3種の計算からなる．

1）Δ計算：ポリゴンのエッジに沿ってyが1増加したときの輝度値（この他にz値やテクスチャアドレスなどもあるが，簡単のために輝度値とのみ表記する）の増分Δyと，x方向に1増加したときの輝度値の増分Δxを求める．

2）スパン計算：ポリゴンとスキャンラインの交線をスパンと呼ぶ

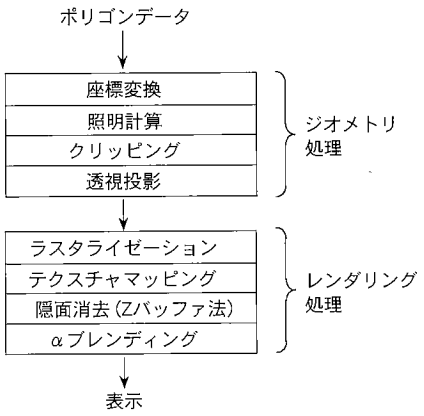


図-1　3Dグラフィクス処理の流れ

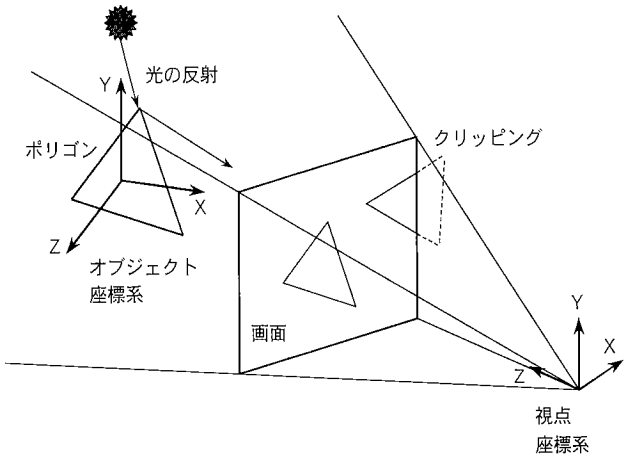


図-2　ジオメトリ処理

	加　算	乗　算	除　算	平方根	べき乗	転送量
座標変換	12	12				
照明計算	8	11	1	1	1	
クリップテスト	6					
透視投影		2	1			
ジオメトリ処理合計	26	25	2	1	1	0

表-1　頂点ごとのジオメトリ演算量

が，そのスパンの左端点の輝度値にΔyを加算して順次次ラインのスパンにおける左端点の輝度値を求める．

3）画素値計算：スパンの左端点での輝度値にΔxを順次加算してスパン内部の各画素位置での輝度値を求める．ただし，スパンの左端からそこに最も近いスパン内の画素位置までの距離をrとすると，スパン内の最左の画素位置での輝度値は，左端の輝度値＋Δx・rで補正する．

b）テクスチャマッピング

1）ポリゴンの各頂点に与えられたテクスチャアドレスを補間してポリゴン内部の画素位置（x, y）でのテクスチャアドレス（u, v）を求める．

2）（u, v）が指すテクスチャ画像の輝度値を読み出して（x, y）に貼り付ける輝度値を求める．

3）2）で求めた輝度値に光源を考慮したシェーディングデータを加味して（x, y）での輝度値を求める．

1）2）の具体的な処理方法によって生成画像の品質に違いが生じる．以下ではそれらの違いを説明する．

1）における最も簡単な補間法は線形補間である．すなわち，頂点に与えられたテクスチャアドレスを線形に補間してポリゴン内部の各画素に対応するテクスチャアドレスを求める．通常は計算量の点からこの方法を用いることが多いが，この場合透視投影によるテクスチャの「ずれ」による画質低下が問題になる．こうしたずれは，ポリゴンの頂点でのZ値の差が大きい場合に顕著に現れ，特に動画像にすると歪みがはっきり分かることがある．この歪みを補正する手法がパースペクティブコレクションであり，ポリゴン頂点のテクスチャアドレスをいったん画面上の座標系に表現してから線形補間を行い，再びポリゴン上の座標系に戻すことで実現できる．具体的には

ア）視点座標系でのポリゴン頂点のテクスチャアドレスu, vと奥行き値wをスクリーン座標系での値に変換した，U＝u/w, V＝v/w, W＝1/wを求める．

イ）スクリーン座標系でのポリゴン頂点のテクスチャアドレスと奥行き値を線形補間して，ポリゴン内部の各画素でのテクスチャアドレスと奥行き値U, V, Wを求める．

ウ）スクリーン座標系でのU, V, Wから視点座標系でのテクスチャアドレスをu＝U/W, v＝V/Wによって求める．

イ）の線形補間は，パースペクテ

	加　算	乗　算	除　算	平方根	べき乗	転送量
ラスタライゼーション	632	104	24			
テスクチャマッピング	250	502	51			800
隠面消去	50					400
αブレンディング	100	100				400
レンダリング処理合計	1032	706	75			1600

表-2　レンダリング処理の演算量

ィブコレクションを行わない場合のテクスチャアドレスを求める方法と同じであり，これはラスタライゼーションで実行する．このため，ポリゴンの頂点ごとに行うア)の処理と，ポリゴン内部の各画素ごとに行うウ）の処理とがパースペクティブコレクションを行うことで余分に発生することになる．

また2）では，テクスチャアドレス（u, v）が指すテクスチャ画像を読み出すと述べたが，u, vは一般には整数値とはならない．そのため簡単な方法としては，(u, v) に最も近い整数格子点をテクスチャアドレスとする．これをポイントサンプリングという．しかし，ポイントサンプリングでは画質が悪いため（u, v）の近傍4点で輝度値を読み出し，それらを加重平均するバイリニア補間が用いられる．

さらに小さいポリゴンに大きなテクスチャ画像を貼り付けるような画像の縮小を伴う場合，折り返し歪みというモアレが生ずる．この折り返し歪みを少ない計算量で弱める手法がミップマップ法という手法である．

これは

1）あらかじめ低域通過フィルタをかけて縦横両方向を1/2, 1/4, 1/8, ... に縮小した複数レベルの画像（ミップマップ）を生成しておく．
2）テクスチャ画像をポリゴンに貼り付ける際の縮小度を求めて，縮小度に応じたレベルのミップマップを選択する．

縮小度からレベルを求めると一般には整数値とはならないので，両側のレベルを用いて加重平均する．

c）隠面消去（Zバッファ法）

フレームバッファと同じサイズの奥行き値（Z値）のバッファ（Zバッファ）を用い，1画素ごとに隠面消去を行う手法である．基本的には以下の手順で行う．

1）ポリゴン頂点で与えられたZ値を線形補間してポリゴン内部の各画素でのZ値を計算する．
2）1）で計算した新Z値とZバッファ中のZ値とを比較して，新Z値の方が小さい場合にのみフレームバッファ，Zバッファへの書き込みを行う．

1）の計算はラスタライゼーションでの処理となるため，隠面消去としては，2）の比較処理だけである．

d）αブレンディング

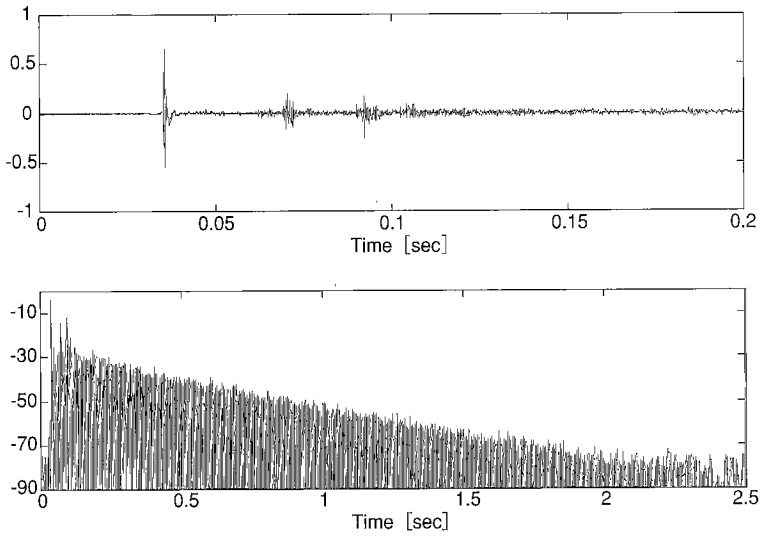
レンダリングの最終段階で計算結果の輝度値とフレームバッファ中の輝度値とを混合係数αで混ぜ合わせる．この混合係数αは，通常は入力時にポリゴンの頂点に与えられ，シェーディングの際に輝度値とともに線形補間で求められる．輝度値をR_i, G_i, B_i とすると計算式は以下のようになる．

$\alpha R_0 + (1-\alpha) R_1$

なお，他の輝度値G, Bも同様である．

●処理の計算量とデータ転送

計算量とデータ転送量を概算するために，ここでは1ポリゴン当たりの計算量とデータ転送量を整理する（表-2）．ただし，1ポリゴンは10ライン，50画素からなる標準的なメッシュ三角形と仮定する．メッシュ三角形とは，頂点を共有して繋がっている三角形群で，頂点数＝ポリゴン数＋2となる．このため，繋がっている三角形数が多くなれば頂点数とポリゴン数がほぼ同じと考えられるので，ここでは頂点数＝ポリゴン数とした．また，照明計算は，拡散反射光，鏡面反射光各1個，テクスチャマッピングはパースペクティブコレクションあり，バイリニア補間の場合で考える．細かい計算方式により計算量が若干変わるため，下記に示すデータは大まかな目安である．先に述べた処理ごとに，加算（減算，比較を含む），乗算，除算（逆数），平方根，べき乗の数と，メモリアクセスのデータ転送量をまとめる．また，データ形式はZ4バイト，RGBα4バイトとして算出した．表の転送量の単位はバイトである．なお，ジオメトリ処理はポリゴンサイズに無関係で浮動小数点演算，レンダリング



上：インパルス応答の先頭部分（縦軸Linear表示）
下：インパルス応答の全景（縦軸dB表示）

図-3　音楽専用ホールでのインパルス応答の例

処理はポリゴンサイズに依存し通常は固定少数点演算のため，計算量の合計はそれぞれ分けて算出した．

ジオメトリ処理の平方根やべき乗計算は通常の乗加算器を用いて計算した場合数十ステップ消費する．このため，それらの演算数が少なくても，特殊な演算器がなければジオメトリ処理で100～150回の乗加算が必要になる．特殊な演算器がない場合，たとえば，10Mポリゴン/秒の描画を行わせようとすると，ジオメトリの演算だけで1～1.5GFLOPSの演算性能が必要になる．

また，レンダリング処理では，1ポリゴン当たり，約1800回の演算と1600バイトのデータ転送が必要になる．たとえば，5Mポリゴン/秒の描画を行わせる場合（ジオメトリ処理で裏面除去の処理を追加すればレンダリングでの処理すべきポリゴン数は半分になる），約9GFLOPSの演算性能と，約8Gバイト/秒のデータ転送性能が必要になる．

【音響信号の実時間処理】

音響信号処理に必要な処理で比較的高い性能を要求するのは音場処理とMIDI音源である．まず音場の概説から入ることにする．

●音場

音が音源で発生後受音点に到達するまで音波は空間を伝搬していく．室内の場合は壁，天井，床などの周壁で音波を反射する．このように音波が伝搬する空間を音場（sound field）と呼ぶ．野外では障害物がない限り音源からの音波は遠くに広がっていき，次第に減衰するので，音場は単純である．室内では周壁の影響を受けるので，室内音場は次の2点の特徴を有する[2]．

a）音源からある距離の受音点における音の強さは自由空間に比べて大きく距離が離れていてもそれほど減衰しない．

b）音源が停止した後にも遅れて到達する反射音により残響を生じる．

コンサートホールでは“響きがいい”というような言葉でその善し悪しが評価されるように音場の設計が大きな問題となる．コンサートホールでは音場に関する種々の現象の中で残響が重要である．室内で音源から音を出すと，ある時間ののち定常状態に達し，音源を止めても音は瞬間になくならず，次第に減衰していく．このように音源を止めたあとも音が残る現象を残響という．

一般に受音点では，まず音源から直接到来する直接音ののちに，周壁などで反射した音が直接くるが，これを初期反射音という．続いて何度も反射音を繰り返した音が重なり合って到来するが，これを後部残響音と呼ぶ．このような連続した音を出す音源では直接音が，過去の音の初期反射音や後部残響音と重なり合って受音点に到来することになる．このような残響の特性を測定する方法が開発されている．インパルス応答の例を図-3に示す．

これはコンサートホールのステージ上に音源を置き，客席で受音した実測のインパルス応答である．まず最も強い信号である直接音があり，次いで0.1秒あたりに何本かの初期

反射音がある．これはホール内の周壁で1～3回程度反射した音である．その後密度の濃い後部残響音が続くが，これは時間とともに減衰するが，高い周波数ほど早く減衰するので，周波数特性の測定に便利なのが近接四点法と呼ばれる方法である[3]．

このような残響を音に付加することによって声や演奏に「はり」や「つや」，豊かな音量感を持たせるだけでなく，ホールや教会，洞窟といった音場の雰囲気を再現させることができる．

●残響付加のための処理

一般に残響のない音源をドライソース（dry source）と呼ぶ．たとえばMIDIで書かれた音楽を音源のみで演奏する場合，これはドライソースである．このドライソースの音に残響を付加するためにはドライソースの音に残響のインパルス応答を畳み込む必要がある．インパルス応答はr(t)，ドライソースからの音をs(t)とすると実際に聴く音a(t)はa(t)＝Σs(t－nu)r(nu)という演算になる．ここにuは音のサンプリング周期であり，足し算はnについて0（現時点）からuの周期で時間軸を無限大まで行うが，しかし実際には残響は2秒ないし3秒なので3秒間で十分である．たとえばサンプリング周期を50KHzとするとnは0から150K（K＝1000）となる．この演算をサンプリング周期ごとに行うので，毎秒の演算回数は150K×50K＝7.5G回となる．これだけの回数の掛け算を行い，その結果を足し込むことを浮動小数点で行うので要求される演算性能は15GFLOPSということになる．これがスピーカごとに必要になる．一見すると大きな数字であるが，現在のDSPが数GFLOPSの能力を持っていることを考えると実現性は十分あると考えられる数字である．

この検討は時間軸の畳み込み演算で行ったが，これらを周波数軸へ変換すると畳み込み演算は単なる掛け算になることはよく知られている．周波数軸では演算量がほぼ2桁少ない処理量ですむ．変換に伴う処理を考慮しても確実に1桁以上少ない演算量で畳み込みが計算できる．周波数軸での演算の欠点は遅延である．直交変換のためには音源からのデータをある時間分ためてから行わなければならないので，その分処理に伴う遅延が避けられない．レコードのように蓄積されたドライソースの音を再生するには少々の遅延を伴っても差し支えないが，電子楽器での演奏を畳み込んで残響付加するには遅延は10ms以下程度に抑制しなければならない．

●音源のための演算

一般にソフトシンセサイザと呼ばれ，MIDIデータのストリームを入力し，オーディオデータのストリームを実時間で出力するデコーダ（グラフィクスのようにレンダラーともいう）である．ソフトシンセサイザはまず音源1チャンネル分の波形を求めるため与えられたピッチ情報に従って波形データを補間処理する．左右と残響，リバーブ効果用データの3つに分離したもののそれぞれでゲインを制御し，他の音源からのデータを足し合わせる．その後リバーブ効果を処理し，左右別々にそれぞれのデータに加算する[4]．一般的に波形データは整数で表され，乗算処理が浮動少数点の方が速いのでこのような浮動少数点と整数の乗り換えが必要になる．この演算は通常は10ms分程度にまとめて行うのでそれだけの遅延が生じる．この演算は比較的単純だが，たとえばサンプリング周波数を50KHzとし，64チャネルの音があれば毎秒50K×64＝3.2M回の演算が必要になる．必要な演算は波形テーブルから読み出したデータの補間とゲインの乗算を左右で行うので15演算で，これに整数から浮動小数点への変換処理が加わる．これを音源ごとに行うことになるが，64音とすれば64×15×3.2M＝3GFLOPSとなる

【あとがき】

マルチメディアホームコンピュータに必要な性能を推定した．処理能力は高いに越したことはないが，コストと使用可能な技術の兼ね合いである．さらにメモリ量の推定も必要であるが，この程度の処理性能はコスト高を招かない範囲で実現可能と思われる．第3回はこの推定をもとに標準的なマルチメディアホームコンピュータを議論する．

参考文献

1) 相川恭寛: Open GLプログラミングガイド，技術評論社（1995）．
2) 大賀寿郎，山崎芳男，金田　豊: 音響システムとディジタル処理，電子情報通信学会（1995）．
3) Yamasaki, Y. and Itow, T.: Measurement of Spacial Information in Sound Fields by Closely Located Four Point Microphone Method, J. Acoust. Soc. Jpn. (E), Vol.10, pp.101-110 (1989).
4) 新井　純: シンセサイザーセオリー，新人社（1988）．

（平成12年5月29日受付）