グラフィクスとCAD　76－5
（1995．　8．　17）

射影空間の図形の表示技術

新関雅俊
大阪電気通信大学工学部電子機械工学科
〒572　大阪府寝屋川市初町18-8

同次座標による図形・形状処理の研究の一環として，同次座標による図形の表示技術について述べる．射影空間の線分を表示するための方法として，同次座標による線分のクリッピングおよびポリゴンクリッピングをみなおす．また，射影空間の多角形を塗りつぶして表示するためのスキャンコンバージョンのアルゴリズムを紹介する．これらのアルゴリズムは射影空間の図形だけではなく，通常のユークリッド図形を表示するためにも利用できる．射影空間の図形の表示は有理多項式曲線・曲面の表示や円錐曲線・２次曲面の表示に有用である．また，これらのアルゴリズムは一貫した表示プロセスの開発のための重要なツールとなる．

Display of Primitives in Projective Space

Masatoshi Niizeki

Osaka Electro-Communication University
18-8, Hatsu-cho, Neyagawa, Osaka 572, Japan

Display algorithms for line segments and polygons in projective space are introduced. The homogeneous coordinate line segment clipping algorithm is extended to display homogeneous line segments in projective space. A scan conversion algorithm for filling homogeneous polygons is also derived. The algorithms described here are useful in displaying Euclidean primitives as well as projective ones. Applications of these algorithms include the display of rational polynomial curves and surfaces and quadric surfaces. The algorithms also consititute an integral part of a homogeneous coordinate display pipeline.

1．はじめに

これまで，同次座標を用いた幾何演算についていろいろな研究を行ってきた．その研究によって特に，完全4次元処理と4×4行列式法を用いれば，図形の変換や干渉処理を同次座標に基づいて行い，計算過程で通常座標に直さないで済むという大きな利点があることが分かった．その理論体系では射影空間の中の図形を扱う．通常の点，線分，3角形，4面体，多角形，多面体に加えて，無限遠点や同次線分，同次3角形，同次4面体，同次多角形，同次多面体を扱う技術が開発された．この技術によって，次のことが実現した．

(1) 完全4次元処理と可変倍長整数演算による無誤差演算方式

(2) 透視変換など，一般の射影変換を行ったあとの図形に対する統一的な干渉処理の方式

(3) 有理多項式曲線・曲面の制御多角形，多面体に関する処理の方式

(4) 2次曲面に関する同次座標を用いた処理の方式

それらの研究において，射影空間の図形をディスプレイ上に表示するための方法が必要になった．

また図形の干渉処理の研究とともに，図形の表示のためのしくみについての研究を行った．表示を行うためには，変換処理，クリッピング，隠れ線／隠れ面消去，スキャンコンバージョンなど，いくつかの処理を連続して行うが，それぞれの処理のたびに通常座標に直す処理を行うのでは，効率が悪く，精度も落ちる．また，通常座標に直してしまうと正しく行うことができない処理もある．そこで，一貫して同次座標で表示のための処理を行う技術があると便利である．

実際，表示プロセスの中でスキャンコンバージョンやクリッピングがどの時点で行われるかは，場合によって異なる．従って，クリッピングやスキャンコンバージョンのアルゴリズムにおいて，射影変換後の一般化された図形が扱えることが重要になってくる．そのためには，同次座標によって表現された図形を一貫して正しく扱うための技術が必要である．具体的には同次線分，同次3角形，同次多角形，同次多面体を表するための技術が必要になる．

従来から同次座標を用いてクリッピングを行うための方法が提案され，広く利用されている．またクリッピングの処理における直線と点の双対性についての研究も行われている[Blinn 1978], [Arokiasamy 1989]．また多角形に対するクリッピング処理についても同次座標を用いた処理として実現することは容易である．しかし，クリッピング以外の処理について一貫して同次座標を用いた処理を行い，透視変換後の図形を正しく処理するための方法は確実に行われているとはいえない．

そこで，この研究では同次座標によって表現された図形の表示技術について検討する．この研究によって通常の図形を含め，任意の図形を一貫して同次座標によって表示を行うための方法を確立する．またこの方法を利用して，射影空間の図形の表示のための技術を確立することを研究の目的とする．

2．射影空間

2．1　仮定事項

ユークリッド空間の図形を表示するときには，できるだけ実際の物体の見え方に近い表示を行うことが目標となる．しかし，射影空間はユークリッド空間とは異なる空間の構造をしているので，現実には見ることができないような状況が発生する．そのような表示を行うべき図形が発生した場合には，表示方法についていくつかの約束や仮定を定める必要が出てくる．特に，無限遠点や無限遠平面との関わりによって生じる状況への対処法を定める必要がある．

(1) 考えている図形が普通に我々が考えているユークリッド空間に無限遠平面を追加した空間である

(2) 有限点だけからなる図形は従来の図形と同じ表示をする

(3) 通常の透視変換と同じ行列で変換をして表示する

(4) 無限遠点を含む図形に関しても完全に表示できるようにする（無限遠平面と交わる図形を扱う）

(5) 無限遠の点を含む図形で処理が破綻しないようにする

(6) 無限遠平面より向こうに位置するものは見えないと仮定する

(7) 実際に物体が見える最も遠い位置を無限遠平面と定める．そのため，無限遠平面と交差する図形はその一部しか見えないと考える．

2．2　同次座標

3次元空間の点は4次元の同次座標ベクトルによって表現される．たとえば，$[X\ Y\ Z\ w]$ のようにして点を表す．お互いにスカラー倍の関係にある同次座標ベクトルは同じ点を表現するものと考える．同次座標ベクトルによって表現される点の全体を射

影空間と呼ぶ.

2．3　ユークリッド空間の図形の射影空間への埋め込み

ユークリッド空間にある図形を射影空間に埋め込むための方法として，次のような方法を用いることが多い．第4座標 w を同次座標ベクトルのスケールファクタあるいは重みと呼ぶ．この重みを1とし，他の座標を通常座標の x, y, z 座標とすると，射影空間の1点が得られる．この対応をユークリッド空間から射影空間への埋め込み写像として用いることが多い．このようにするとユークリッド空間の中の点に対応して得られた点はすべて，重みがゼロではない．逆に重みがゼロであるような同次座標ベクトルによって表現される射影空間の点を無限遠点として区別することがある．それに対して重みがゼロでない同次座標ベクトルによって表現される射影空間の点を有限点（有限遠点）と呼ぶことにする．

射影空間の中に存在する図形は射影変換によって，射影空間の中を自由に変形し，移動させることができる．その際，その図形が最初に持っていた性質の多くがそのまま保たれる．

2．3　同次図形の定義

ユークリッド空間の線分，3角形，4面体，多角形，多面体は通常の定義と同じとする．ユークリッド空間の線分，3角形，4面体，多角形，多面体を射影空間に埋め込んで得られる図形をそれぞれ，同次線分，同次3角形，同次4面体，同次多角形，同次多面体と呼ぶ．

（1）同次線分

普通にこれまで利用してきた線分はいわゆる内側線分に加えて，外側線分，半直線（一方の端点が無限遠点となっている線分），無限遠線分などが同次線分である．立体を画面上に表示しようとしたときに，視点が立体に近すぎるときに外側線分が発生することがある．ユークリッド空間の線分を射影空間に埋め込むと同次線分を得る．同次線分は射影変換に関して閉じている．外側線分は表示すると，2つの部分に分かれているように見える．一方は重みが正の部分であり，他方は重みが負の部分である．

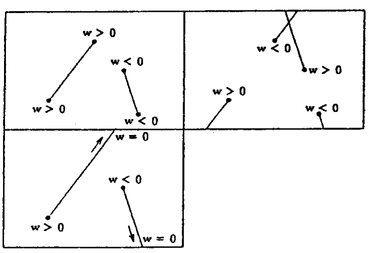

図1　同次線分の例

（2）同次3角形や同次多角形

同次3角形や同次多角形は同次線分を辺とする図形である．同次座標ベクトルの環状の配列が与えられたとする．そのとき，となりあう同次座標ベクトルによって定義される同次線分はすべて同一平面上に存在し，お互いに交差しないとすると，平面は2つの部分に分割される．射影空間の平面上で考えると，この2つの部分のうち一方は円板と同相であり，一方はメビウスの帯と同相である．円板と同相な部分とその境界を併せて同次多角形と呼ぶ．ユークリッド空間の多角形を射影空間に埋め込むと同次多角形が得られる同次多角形は射影変換に関して閉じている．

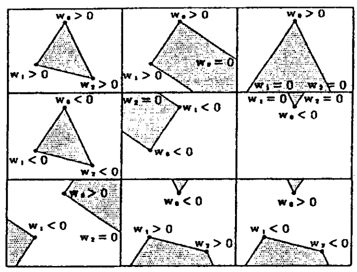

図2　同次3角形の例

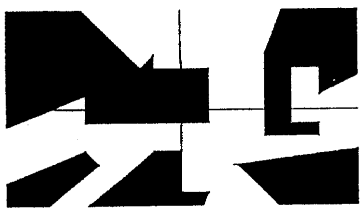

図3　同次多角形の例

3．同次線分の表示と同次座標クリッピング

同次座標クリッピングの方法はBlinnらによって示されている．同次座標によるクリッピングは，通常の線分の表示のために使用されることが多いが，実際のアルゴリズムの内容としては同次座標ベクトルによって表現された同次線分の表示を行うためのアルゴリズムの一部である．射影空間の中の線分である同次線分を表示するためにはこの同次座標クリッ

ピングの技術を用いる．

3．1　2次元同次座標クリッピングの概略

クリッピングの第一段階として行うのは，与えられた線分を含む直線が与えられた直線の集合に属しているかどうかの判定である．その直線の集合に属していない直線は，クリッピングウインドウと交差している可能性がない．クリッピングの第一段階では，このように直線の集合に対する直線の内外判定である．第二段階では線分の両端点がどのような位置に存在しているかによって，線分がクリッピングウインドウとどのような交差をしているかを判定する処理である．線分の端点に関する第二段階の処理では，点の同次座標によって構成される行列式の符号判定式によって処理を行うことができる．

3．2　同次線分の表示

一般に同次線分を表示するためには，同次座標クリッピングを2回行えばよい．一回のクリッピング処理を行うことによって，線分上の重みが正の部分が表示される．続けて同次線分を定義する同次座標ベクトルの符号を逆転させ，もう一回同じクリッピング処理を行うことにより，もともとは同次線分の重みが負の部分が表示される．同次線分はディスプレイ上では2つの独立した部分によって構成されることがあるので，同次線分を表示するためにはこのように2つの部分にわけた表示を行うことになる．

同次線分を表示する処理の中で行われる数値計算は，この2つのクリッピング処理で共通の部分がほとんどであり，重複して計算する必要はない．また，同次線分の表示そのものは，完全に独立した2つの処理にわけることができるので，並列化によって通常の線分の表示処理と比べて速度の低下を引き起こさないで実現することができる．

3．3　3次元空間の同次線分の表示

3次元空間に存在している同次線分を表示するときには3次元の同次座標クリッピングを行う必要がある．表示のための仮定により無限遠平面と交わる図形は，無限遠平面より手前に位置する部分しか見えない．見える部分だけを求めて表示するためには，クリッピングを行わなければならない．つまり無限遠平面と交わる同次線分の表示のときには，無限遠平面を越えた位置にある部分をクリッピングをして排除しなければならない．3次元の同次座標クリッピングの技術を用いて，視点ピラミッドの後方の面であるヨーン面 (yon plane) として無限遠平面を設定すれば簡単に実現することができる．

表示の際には2次元の場合と同様に，同次線分の重みが正の部分と負の部分を表示するために同次座標クリッピングを2つの処理にわけて行う．

3．4　ポリゴンクリッピング

ポリゴンクリッピングを行うためのアルゴリズムはすでにいくつか提案されている．それらのアルゴリズムの計算部分はいずれも容易に同次座標を用いた処理に置き換えることができる．この同次座標によるポリゴンクリッピングの処理を多角形の重みが正である部分と，重みが負である部分についてわけて2つの処理として行うことにより，射影空間の中での同次多角形のクリッピングが実現できる．

3．5　同次線分の表示方法のまとめ

射影空間の中の平面（射影平面）上に存在する線分（同次線分）を表示するためのは，2次元の同次座標クリッピングの処理を2度行えばよい．2つの処理部分ではそれぞれ同次線分の重みが正の部分と，重みが負の部分を表示する．この表示方法は同次座標クリッピングの技術でなければ正しく行えない．

3次元空間の中に存在する任意の線分（同次線分）を表示するためには2つの処理によって同次線分の重みが正の部分と，負の部分にわけて表示を行う．また，無限遠平面によるクリッピングを行うことによって無限遠平面より向こうに存在する線分の部分を排除することができる．

同次座標クリッピングを用いることによって射影空間の中の線図形を表示することができる．同次線分を表示することができれば，多面体のワイヤフレーム図や多角形の境界を表示することができる．

4．同次多角形のスキャンコンバージョンのアルゴリズム

同次線分の表示ができれば，射影空間の中の立体などのワイヤフレーム表示をすることができるが，それだけでは不十分である．実際の表示には面による表示が必要になる．そのために使用するのは，多角形のスキャンコンバージョンのアルゴリズムである．射影空間の中の多角形である同次多角形に関しても，通常の多角形に対するものと同様の考え方で

塗りつぶしをすることができる．しかし同次多角形の塗りつぶしのためには，射影空間はユークリッド空間と異なる構造を持つことと，同次多角形の多様性を考慮した拡張されたアルゴリズムに替える必要がある．

ここではスクリーン上に同次多角形を塗りつぶして表示するための方法について述べる．同次多角形の塗りつぶしは，前述の同次多角形に関するポリゴンクリッピングを行うことによって通常の多角形の塗りつぶしとして行うことができる．ポリゴンクリッピングによって得られる同次多角形の2つの部分はいずれもスクリーン上では単なる多角形であり，普通の多角形のスキャンコンバージョンアルゴリズムで塗りつぶすことができる．

しかし，ここではあえてポリゴンクリッピングの処理を行わず，与えられた同次多角形を直接扱うことで塗りつぶす方法について述べる．変換処理，線分のクリッピング，多角形のクリッピング，スキャンコンバージョン，そして隠れ線／隠れ面消去をそれぞれ独立した処理として実現する．それぞれの処理を独立させることによって，それぞれの処理を表示プロセスにおいてある程度以上，自由な順序で行うことができるようになる．それぞれの処理を自由に配置できるようになることで，目的にとって理想的な表示プロセスを実現することができるようになる．

また，本研究で利用しているように射影空間の図形を簡便な方法で表示するためには，通常の多角形のスキャンコンバージョンのプログラムを同次多角形に関するものに置き換えるだけで済ますことが最も簡単である．そのため，ここでは同次多角形を表示するために直接スキャンコンバージョンを行うための方法を示す．

4．1　前提条件

同次多角形のスキャンコンバージョンを行うために考慮に入れなければならないいくつかの前提事項がある．

(1) スクリーンは射影平面である

表示をするスクリーンは射影平面の一部である．そのため，スキャンラインは射影直線であり，スキャンライン上に1個無限遠点が存在する．

(2) 同次多角形の向きの規約

同次多角形の頂点の順序に向きが定められている．同次多角形を含む平面の表裏を定めることができる．平面の表側からみたときに同次多角形の正の重みを持つ頂点については反時計回り，負の重みを持つ頂点は時計回りになっているものを今回の表示の対象とする．この規約は，ひとつの立体を与えられたときに前方面だけを表示することに相当する．後方面も表示したい場合にはすべての頂点の重みの符号を逆にして考えるとよい．この規約は，通常の多角形の表示を行っているときの考え方と合致している．この規約によって，開始稜線／終了稜線などの区別をすることが可能になる．

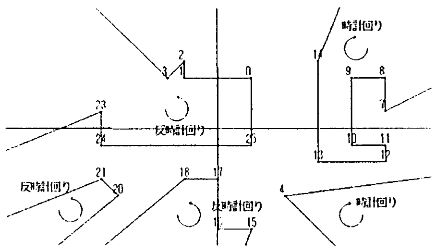

図4　同次多角形の頂点の順序

(3) スクリーン座標系

ここで使用したスクリーン座標系は左上端に原点を置き，右向きをx軸の正の方向，下向きをy軸の正の方向となるようにとる．この設定の仕方は現在のスクリーン座標系として最もよく利用されているものである．

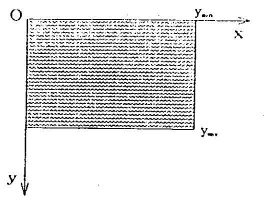

図5　スクリーン座標系

(4) スキャンライン

スキャンラインは前述のように射影直線であり，円と同じ構造を持っていると考えることができる．このスキャンライン上に1点だけ無限遠点が存在する．同次多角形の境界（辺）とこのスキャンライン

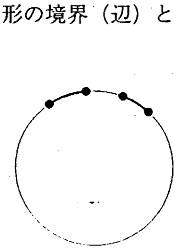

図6　射影直線のスキャンライン

は通常の多角形の場合と同様に偶数回交差する（ただし交点が重なる場合もある）．スキャンラインと同次多角形の内部が交差する部分は，何カ所かの区間になっている．それぞれの区間は同次線分である．従って，内側線分，外側線分，半直線となっているものがありうる．これらの区間をスキャンラインごとに順にスクリーン上に表示することによって同次多角形が表示される．

また従来のスキャンラインと大きく違うのは，すべてのスキャンラインは無限遠点で交わるということである．すべてのスキャンラインは画面上では交わるところはないが，すべて同じ無限遠点を通る．つまり，スキャンラインの集合は1個の無限遠点を共有する射影直線の集合である．この無限遠点ははx軸方向の無限遠点であり，同次座標で表現すると[1 0 0]となる．

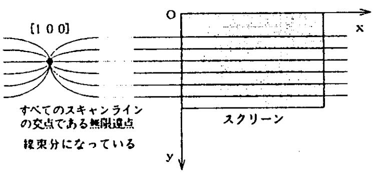

図7　スキャンラインどうしの交点

4．2　アルゴリズムの詳細

スキャンコンバージョンのアルゴリズムの流れは通常の多角形の場合とほぼ同じである．

(1) 稜線の進入と退出

スキャンラインそれぞれに関してxソートリストとyソートリストを管理する．xソートリストによって稜線のスキャンラインへの進入と退出を監視する．スキャンラインは円と同じ構造であるため，yソートリストは円順序が扱えるようにする．

通常の多角形の場合は内側線分だけを処理すればよいが，同次多角形の稜線は，内側線分，外側線分，半直線，無限遠線分によって構成される．それぞれの種類はスキャンラインとの交差の仕方が異なる．同次多角形の稜線でスキャンラインと交点を持ちはじめる時点で進入稜線となり，交点を持たなくなった時点で退出稜線となる．あらかじめ水平な稜線（x座標が一定の稜線）はすべて排除しておく．また，スキャンラインとまったく交差しない稜線もまたあらかじめ排除しておく．

内側線分は通常の多角形の場合と同様の進入／退出の処理を行う．スキャンラインと交点を持ちはじめる時点でで稜線ブロックをxソートリストからyソートリストに移す．また，交点を持たなくなった時点でyソートリストから削除する．

外側線分の扱いは同次多角形に特有である．外側線分は最初は必ずスキャンラインと交差する．スキャンラインが移動して，端点がスキャンラインに達すると一度その稜線はスキャンラインと交わらなくなる．もうしばらくスキャンラインが進むと，再度進入する．そのため，外側線分ははじめからxソートリストに入れておく．交点を持たなくなればいったんyソートリストから削除するが，再度進入する可能性がある稜線についてはもう一度xソートリストに戻すか，再進入稜線リストのような別のリストに入れて管理すればよい．そして，再度進入した時点でyソートリストに移す．

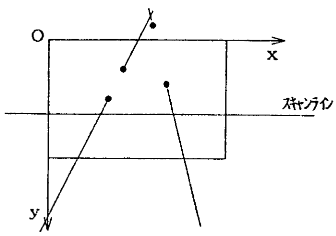

図8　外側線分の進入／退出

半直線の扱いは内側線分とまったく同じである．ただし，半直線の一方の端点である無限遠点がスキャンラインの無限遠点と一致している場合だけは区別する必要がある．この場合には，この稜線は常にスキャンラインと交点を持つ．それ以外の場合には，内側線分と区別して扱う必要はない．

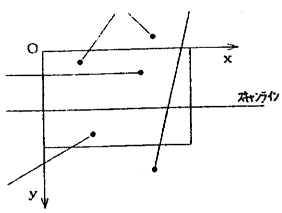

図9　半直線の進入／退出

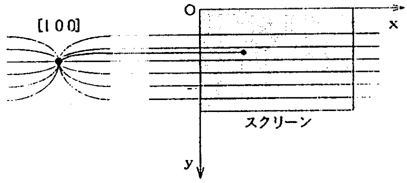

図10　スキャンラインの交点を通る半直線

無限遠線分は両端点の重みがゼロである線分である．無限遠線分がスキャンラインと交点を持つとすれば，それはスキャンライン上の唯一の無限遠点である点 [1 0 0] においてである．無限遠線分がこの点と交点を持つかどうかの判定は，その同次線分の重み異符号であるかどうかの判定となる．この点と交差しない無限遠線分はあらかじめアルゴリズムから排除する．この点と交差する無限遠線分は最初からスキャンラインと交差するので，yソートリストに加えておく．無限遠線分は決して退出しない稜線である．

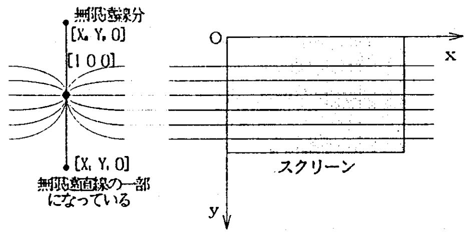

図１１　スキャンラインの交点を通る無限遠線分

(2) 開始稜線／水平稜線／終了稜線の区別

開始稜線とはスキャンライン上を塗りはじめる位置を表し，終了稜線は塗り終わる位置を表す．それぞれの稜線が開始稜線，水平稜線，終了稜線であるかどうかの区別は容易である．同次多角形の場合には，稜線の重みの符号を考慮に入れた判定を行わなければならないところが異なる．また，外側線分は２度目の進入のときには開始／終了の判定が逆になる．最初に開始稜線であった外側線分は再進入のときには終了稜線となる．

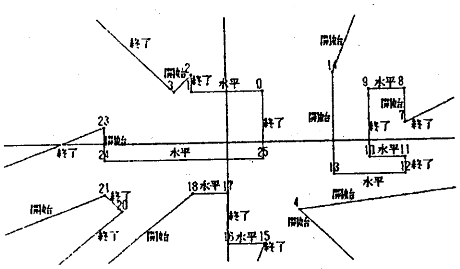

図１２　開始稜線，水平稜線，終了稜線

(3) 頂点の上下の区別

稜線の中に外側線分や無限遠線分が含まれると，どちらの頂点で稜線がスキャンラインに進入し，どちらで退出するかを区別するために注意が必要になる．しかし注意深く調べることにより，容易に判定をすることができる．

(4) 描画

yソートリストの内容を見て，画面上に表示を行う．そのときに表示する区間はそれぞれ同次線分であるため，同次線分の表示ルーチンを用いる必要がある．これは，同次座標クリッピングの方法を簡略化したものを用いることができる．

(5) アルゴリズムの全体とデータ構造

アルゴリズムの流れは，通常の多角形のスキャンコンバージョンと全く同じである．稜線ブロックを作成し，その内容でxソートリストを作成する．スキャンラインの進行とともにyソートリストへの進入／退出を行い，必要な区間をyソートリストを見て塗りつぶす．外側線分の再進入の管理が行いやすいような情報や，稜線の同次座標の重みを記憶するための部分が稜線ブロックに追加されている．

(1)	次のブロックへのポインタ next
(2)	稜線の型フラグ flag
(3)	スキャンライン上のX座標 X
(4)	スキャンライン上のw座標 w
(5)	スキャンライン毎のXの増分 dX
(6)	進入時点のy座標 yt
(7)	退出時点のy座標 yb
(8)	再進入時点のX座標 Xr
(9)	再進入時点のw座標 wr

図１３　稜線ブロックの内容

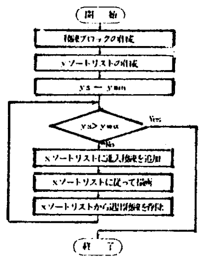

図１４　アルゴリズムの流れ

5．応用例

5．1　同次線分の表示例

同次線分のひとつの大きな応用は有理多項式曲線・曲面の表示，そしてその制御多角形・制御多面体の表示である．図に示したBezier表現の曲面では，制御ベクトルの重みが正と負のものが混在しているので，同次線分の表示ルーチンを使って制御多面体を表示した方が曲面パッチの形状に沿った形と

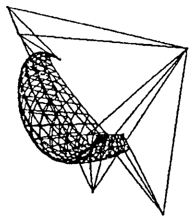

図１５　通常の線分での曲面表示

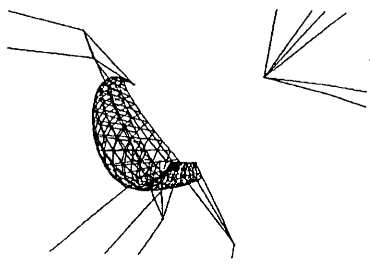

図１６　同次線分での曲面表示

なる．

また，円錐曲線や２次曲面のように無限に広がる図形の表示のときには，２次曲面を射影空間の図形と考えて同次線分の表示のルーチンを用いると安心して表示できる．

５．２　同次多角形の表示例

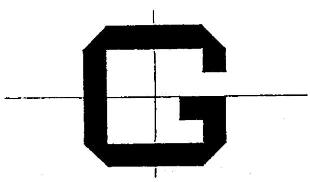

図１７　通常多角形の表示例

同次多角形のスキャンコンバージョンのアルゴリズムを行った例を示す．通常の多角形を表示する場合にも，さほどスピードの低下はないようにプログラムが工夫されている．大きく無限の方に広がるような同次多角形も正しく表示される．

６．結論

射影空間の同次線分の表示方法として，同次座標クリッピングの方法について述べた．

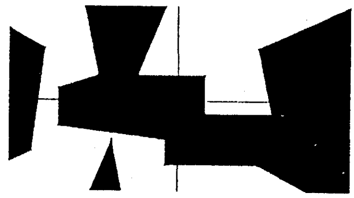

図１８　同次多角形の表示例

射影空間の同次多角形の表示方法として，同次多角形のスキャンコンバージョンのアルゴリズムを示した．

射影空間の図形に関する干渉処理に関する研究においてこれらの処理方法を応用した．

それらの応用例として，有理多項式曲線・曲面の表示や２次曲面の表示を示した．

一般に，射影空間などのようにユークリッド空間とは異なる空間に存在する図形を表示するためには，特殊な技術が必要になる．

７．今後の研究

この方式をさらに高速で有効な表示プロセスの中に組み込む研究が期待される．

参考文献

[1] 山口富士夫，コンピュータディスプレイによる図形処理工学，日刊工業新聞社，1981.

[2] A. Arokiasamy, Homogeneous Coordinates and the Principle of Duality in Dimensional Clipping, Computer Graphics, Vol. 13, No. 1, pp. 99-100, 1989.

[3] A. Kaufman, Efficient Algorithms for Scn-Converting, Computers & Graphics, Vol. 12, No. 2, pp. 213-219, 1988.

[4] J.F. Blinn, M.E. Newell, Clipping Using Homogeneous Coordinates, ACM Computer Graphics, Vol. 12, No. 3, August 1978, pp. 197-203.

[5] Y.D. Liang, B.A. Barsky, An Analysis and Algorithm for Polygon Clipping, Comm. of ACM, Vol. 26, No. 11, Nov. 1983, pp. 868-877.

[6] Y.D. Liang, B.A. Barsky, A New Concept and Method for Line Clipping, ACM Trans on Graphics, Vol. 3, No. 1, Jan. 1984, pp. 1-22.

[7] I.E. Sutherland, G.W. Hodgman, Reentrant Polygon Clipping, Comm. of ACM, Vol. 17, No. 1, Jan. 1974, 186-196.