

LINKS-1によるアニメーションの製作について

河合 利幸, 西村 仁志, 出口 弘, 河田 亨,
白川 功, 大村 皓一 (大阪大学・工学部)

1. まえがき

動画制作をコンピュータグラフィックスで行なう場合、内容の芸術性が高くなる程、アーティストの感性を直接データとして表現することが至難であることから、作業は極めて試行錯誤的になる。従ってシステムに対しては、高い対話性が要求される。そのためには次の条件を満足している必要がある。

- 1) 画像を生成するための処理が高速であること。
- 2) データの入力が容易であること。

上記1)に対しては、スーパーコンピュータの使用が考えられるが、高価である。そこで、筆者らは、市販のマイクロプロセッサを多数使用し、並列処理を行なうことで価格と高速性とを両立させたコンピュータグラフィックスシステムを開発してきた1),2)。一方、画像データ定義の対話性を高めるために、高解像度陰影画像の出力は必要最小限にとどめ、他の部分は、線画もしくは低解像度陰影画像で代用する。

2)に対しては、入力装置としてディジタイザ、ロータリエンコーダ、ファンクションスイッチなどを利用することで、できるだけ数値を直接入力する必要のないようにし、感性に基づいた決定を行ないやすくする。

データ定義の作業は、ある意味ではプログラムの作成と類似する点が多くある。プログラムを作る時にいきなり端末に向かって作業せず、まずフローチャートなどから書き始めるように、データ定義の場合でも、骨格は机上で定義した上で正確に入力し、細部の修正を実際にシステムと対話して行なう方が、結局は効率的である。現在利用可能な入力装置からして、前者は従来の図面作成の手順にならない、三面図を電子的に作成、修正、保管できればよい。一方、後者はそれに基づいてできた画像の局所的な修正が連続的に実時間でできることが必要である。

本文では、以上の考察に基づいて、アーティストの使用経験を反映しながら、設計・製作・改良・強化を行なってきたコンピュータグラフィックスシステムLINKS-1における動画制作の手法と支援システム

について述べる。

2. 物体データ構造

物体モデルを表現するデータを以下では単に物体という。特に文脈上、データであることを明記する必要があるときは物体データという場合もある。

本システムでは画像生成アルゴリズムとして視線探索法4),6)を用いている。このアルゴリズムでは、画像1つ1つに対して多くの計算を行なわねばならない反面、物体の表現形式としては、視線との交差判定の可能なものならどんな形式でも良いという自由度がある。さらに反射、屈折、影などの計算も容易である。

データ構造に関しては、計算の効率化、物体の運動・変形などの表現が容易であること、などが要求される。そこで、計算の効率化を図るため、物体をある領域に箱詰めにし、その外側では視線と物体との交差判定を簡略化できるようにしている4)。一方、物体の運動・変形の表現を容易にするために、この箱詰めにした物体データを木構造に接続し、物体の動きを観の物体データに対する移動ベクトルと回転マトリクスで表現する。

物体データを次の3つのリストで構成する。

形状リスト: 物体の形状を記述する式、及び関数から成るリスト3)。

属性リスト: 物体の色、反射係数、材質、貼り付け用データへのポイントなどから成るリスト3)。

構造リスト: 形状リスト、属性リスト、及び他の構造リストを指すポイント、及びこれらのポイントが指すリストの座標変換情報(即ち、移動ベクトルと回転マトリクス)とを持つリスト3)。構造リストを参照する場合の鍵は構造名による方法が主体である。この名前は物体の名前として用いられる。

これらのリストは、図1に示されるように樹木状に構成される。

物体形状の表現形式として、LINKS-1では大きく分けて、三角板表現、楕円体表現、濃度球表現、直方体表現の4種類3),4)が使用されている。1つの

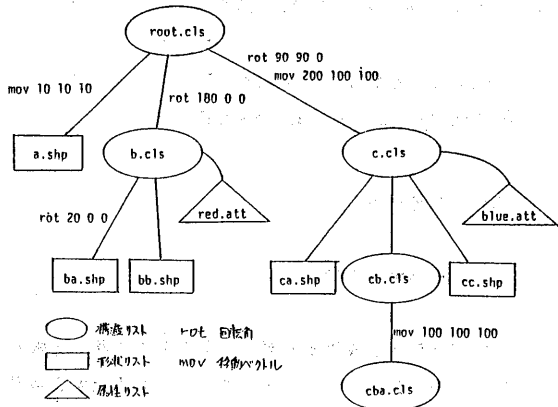


図1 物体データ構造

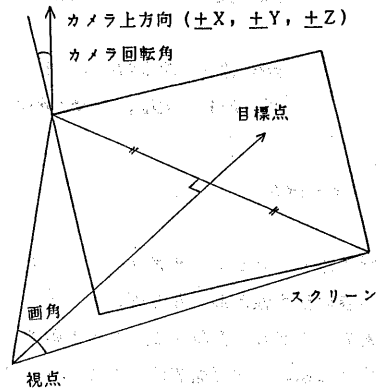


図2 視線定義

物体データはこれらの組み合わせで表現される。可動部分を含まない物体データならば、1つの形状リストとし、互いに可動な物体は構造リストを介して接続する。

3. データ定義

1枚の画像を生成する際には、様々なデータの定義が必要になる。以下では、どのようなデータが必要であるかについて述べる。

1) 物体形状定義と接続

物体の形状を定義するには、まず物体を前章で述べた表現法で表わせるまで分解する。この分解した部品のデータが形状リストである。次にこれらのデータを構造リストを用いて木構造に接続し、1つの物体データを構成する。さらに1枚の画像に必要な物体データをそれぞれ同様にして作成し、これら複数個の物体データを同じく構造リストを用いて接続し、1枚分の画像データとする。接続の仕方は構造リスト中の回転マトリクスと移動ベクトルによって決定される。この回転マトリクスと移動ベクトルは、親の構造リストにおける座標系に対する、子のリストの座標系の座標変換情報である。即ち、 x_i を構造リスト C_i の座標系での位置ベクトル、 R_i を C_i にたいする回転マトリクス、 m_i を移動ベクトルとすれば、親の座標系への変換は次式で与えられる。

$$x_i - 1 = R_i x_i + m_i$$

2) 視線定義

1枚の画像を生成するためには、カメラに関する諸

データを決定する必要がある。即ち、視点の位置、目標点の位置、カメラ上方向、カメラ回転角、画角の5種類である(図2)。

3) 光源定義

LINKS-1では、平行光線、点光源、スポットライトの3種類の光源(4)が用意されている。平行光線では色(強度)・方向、点光源では色(強度)・位置・方向、スポットライトでは色(強度)・位置・方向・スポット半径をそれぞれ指定する。

4) 表面定義

物体の色、材質、反射係数、屈折率、透過率などの表面状態を定義する(3),4)。即ち属性リストを作ることが表面定義である。

以上の4つの定義を行なうことで、1枚の静止画を生成することができる。動画を作成するには、さらにカメラや物体を動かす必要がある。即ち、以上の定義における諸データの時間的変化を定義しなければならない。

部品操作: 物体位置の時間的変化を定義することであり、構造リストの回転マトリクスと移動ベクトルの時間的変化を指定することで定義する。

形状変形: 物体形状の時間的変化を定義することであり、形状リスト内のデータの時間的変化を指定することで定義する。

視線補間: 視点位置、目標点位置などの時間的変化を定義することである。

光源操作: 光源位置や、色などの時間的変化を定義することである。

色操作: 物体の色などの表面状態の時間的変化を定義することであり、属性リスト内のデータの時間的変化

を指定することで定義する。

以上のような操作の表現は、従来の動画を作る時の手法にならない、あるシーンの鍵(キー)となるようなフレーム(キーフレーム)をいくつか定義することで行なう。実際に動画にする場合、1秒当たり24ないし30枚の画像が必要である。あるシーンの各フレームのデータは、定義されているいくつかのキーフレームの各データを標本値として、指定したコマ数でスプライン補間10)することにより生成する。

4. フレームメモリスistem

LINKS-1のハードウェアは、現在図3のようにBIOSプロセッサ5)、画像生成システム2)、フレームメモリスistem、アニメーションシステムの各システムを結合した構成となっている。

BIOSプロセッサはファイル管理及びプログラム開発用として、Zilog社のSystem8000を使用している。

画像生成システムは、マスタ1台、スレーブ64台、計65台のユニットコンピュータ(以下UC)から成るマルチコンピュータシステムである。

フレームメモリスistemは、BIOSプロセッサとつながるUC(画像生成システム中のものと同種のもの)と高解像度のフレームメモリから成る。UCには、ディジタイザとスイッチボックス、もしくはロータリエンコーダボックスが、フレームメモリにはビデオカメラ、CRTディスプレイ、VTRなどのビデオ機器が、入出力装置として接続されている2)。

フレームメモリは画像を画素単位の輝度情報として記憶するもので、1024×1024画素の分解能をもち、メモリ容量にして3メガバイトである。各画素はRGB各8ビット、計24ビットから成り、約1600万色の表現が可能である。

ここでは、フレームメモリスistemを用いたデータ定義用のソフトウェアについて述べる。

4.1 三面図入力システム

前章で述べた物体形状定義を行なうためには、表現形式により多少異なるが、3次元座標値の決定という作業が必要である。従来は直接数値を入力していたので、値の読み取りや、読み取りミスなどによる時間的損失が少なからず存在した。これをなくすには、非数値入力装置が必要であるが、現時点では3次元ディ

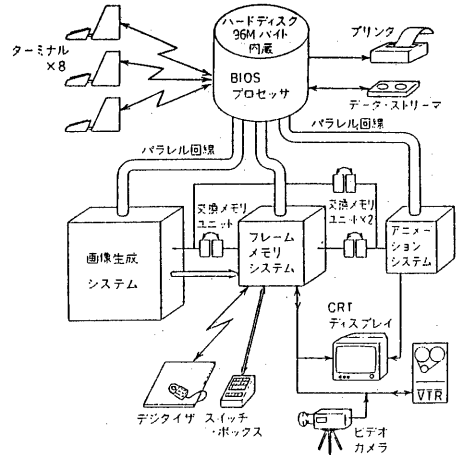


図3 LINKS-1のシステム構成

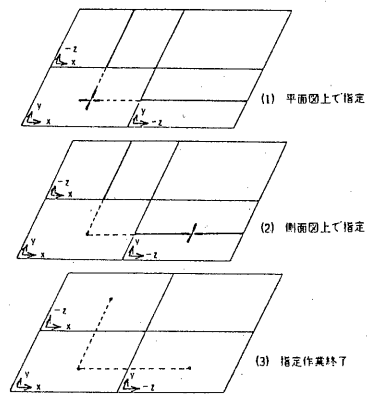


図4 新しく入力する点の指定

タイザは高価であり精度も不十分なため、2次元のディジタイザを使用することにした。

3次元の値を入力するには、予め物体の三面図を用意しておく。三面図のうちの2面で1回ずつディジタイザを用いて指定することにより3次元空間における1つの点の座標を入力する(図4)。既に入力されている点を指定する時には1回ですむ場合もある。これらの作業は前節で述べたフレームメモリスistemで行なう。入力結果は次々とフレームメモリを経てCRT上に三面図として表示される。

現在三面図入力システムは、濃度球表現専用のMPLASと、汎用の多角形入力エディタPLAS、曲面パッチ入力エディタPPLASがある。

MPLASでは、ディジタイザから濃度球の中心を入力する。有効半径は予め設定しておく。入力するとその濃度球の有効半径と、しきい値4)で定められた濃

点名	X座標	Y座標	Z座標
P 1			
P 2			
⋮			
P _n			

線名	点名	点名
L 1	P 1	P 2
L 2	P 2	P 3
⋮	⋮	⋮
L _n	P _n	P 1

図5 PLASにおけるデータ記述

度球が単独で存在する場合の物体表面が三面図上で異なる色の円として表示される。MPLASの最大の特徴は、視線を三面図上で指定して、入力結果を直ちに陰影画像として見るができることである。この機能により、迅速な対話的修正が行なえる。

PLASでは、図5のように多角形の集合を点の座標と点の関係として記述する。また、最初に三面図を紙の上で作っておく必要のないよう、CRT上方眼が表示でき、垂線の足や、ある平面上の点などが直ちに求められるようになっている。MPLASでは陰影表示を行なうのに対し、PLASでは三面図と同時に線画による透視図表示を行ない、入力形状の確認に役立っている。入力結果は三角板表現などに変換される。また、いずれのエディタにおいても、部分拡大や既に入力した部分の表示抑制が可能である。

PPLASでは、次のような手順で三角板近似による曲面データを作成する。

- 1) 格子状の節点をPLASと同様に入力する。曲面はこれらの節点を通る滑らかな面として定義される。
- 2) スプライン関数を用いて節点間を格子状につなぎ、各節点での傾きを求める。
- 3) 格子の4隅の節点の座標と傾きによりFパッチ9)を生成し、小三角板に分割する。これを各格子で行なう。即ち、1枚のパッチはFパッチの集まりとして定義される。
- 4) 陰影表示を行ない、満足な曲面が得られるまで、節点位置を修正して、以上の操作を繰り返す。

```

root.cls
( a.shp mov 10 10 10
  b.cls red.att rot 180 0 0
    ( ba.shp rot 20 0 0
      bb.shp
    )
  c.cls blue.att rot 90 90 0 mov 200 100 100
    ( ca.shp
      cb.cls
        ( cba.cls mov 100 100 100
        )
      cc.shp
    )
)

```

図6 接続定義言語

4.2 キーフレームエディタ

動画を制作するには、前章で述べたとおり、視線や物体の接続などを決定し、キーフレームを作成する必要がある。それを対話的に行なうためのシステムがキーフレームエディタである。ハードウェアとしては三面図入力システムと同様、フレームメモリシステムを用いる。入力装置としてはロータリエンコーダ及びスイッチボックスを使用する。

本システムで作業に入る前に、図6に示す接続記述言語を用いて、シーンを構成する物体データの初期接続を定義しておく。この記述は、図1に示した図的表現と1対1対応しており、BIOSプロセッサのエディタを用いて容易に作成できる。この時、親の構造リストに対する子の構造リストの座標系の各軸まわりの回転角と移動ベクトルを概略の数値で指定しておく。

本システムを用いた視線定義、接続の修正、光源定義は次のように行なう。

1) 三面図上でMPLASと同様に視点と目標点を入力し、視線の概略の定義を行なう。同様に、光源の方向も定義する。

2) 定義した視線での線画による透視図が画面の一部に表示される。

3) 三面図による定義が終わると、ロータリエンコーダを用いて微調整を行なう。この時、画面全体に線画による透視図が表示される。画角やカメラ回転角も指定する。

4) 接続の修正をロータリエンコーダを用いて回転角と移動ベクトルを指定することで行なう。修正を行なっている物体は、輝度を上げて表示される。

ロータリエンコーダによる修正作業では、現在の値からの変位を指定する。構図を決定するには、現在見ている画面をもとにどう変えるかを考えるのが自然で、使いやすいのでこの方式を採用した。

物体数が増えると、微調整の際に線画の描画時間が無視できなくなる。そこで本システムでは、画像生成システムのマルチプロセッサを使用し、各プロセッサに物体データを割り当て、並列処理で描画を行なう。

4.3 アトリビュートエディタ

これは表面定義を支援するためのシステムである。ハードウェアは同じくフレームメモリシステムを用いる。入力装置はディジタイザで、カーソルを動かす、必要なコマンドを入力する(図7)。

表面定義の中でも反射係数の決定は、光源、物体形状の影響もあって困難であった。本システムでは、選択した色と係数で任意の光源と形状による陰影表示が直ちに行なえ、各係数の効果が容易に確認できる。色は色相、彩度、明度を連続的に変化させて選択する。

4.4 テクスチャ定義システム

これは物体表面に模様(texture)を入れるための貼り付け(mapping)に必要な2次元データをディジタイザとスイッチボックスを用いて入力するシステムである(図8)。入力する方法は、キャンバスに絵を画くと同様にディジタイザ上でスタイラスペンを用いて画く。画いた結果はCRT上に表示される。このシステムの特徴は以下に示す通りである。

- 1) ペン先の形は丸形と四角形の2種、太さは15段階のうちから選択できる。
- 2) 色の選択は、RGB成分のそれぞれを直接に増減させて選ぶ方法、100枚のカラーパレットから選ぶ方法、2色の間の任意の中間色を選べるグラデーションバーを用いる方法のいずれでも可能である。
- 3) 操作ミスに備えて、バックアップ機能がある。
- 4) 画面上の任意の部分を最大256倍まで拡大可能である。これにより細部の描画が容易に行なえる。
- 5) 任意の2点間の直線が引ける。必要ならジャギの目立たない直線を引くこともできる。
- 6) 連続した単一色の領域、または単一色で囲まれた連続領域を単一色で塗り潰すことができる。
- 7) 画像データを圧縮し、ファイルとして記憶しておき、読み込んで再び修正することができる。
- 8) 注入型と拡散型の2種類のブラッシングが可能である。注入型は水彩画の効果を持つもので、半透明な絵の具による作画に相当する。ブラッシングペンのペン先の範囲内の画素を0からn-1の添字で表わし、i番目の画素の元の色をR_i, G_i, B_i, ペンの色をR_p,

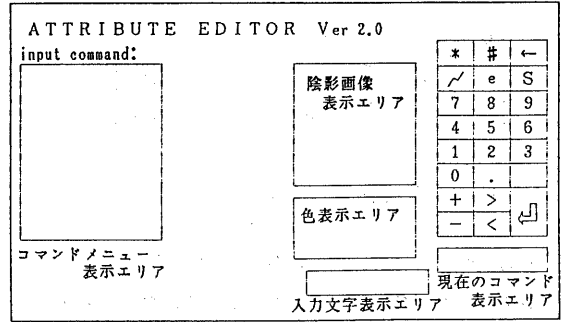
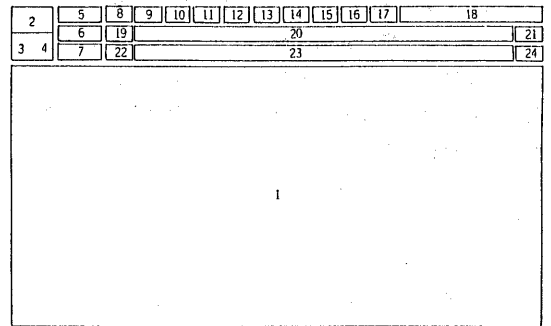


図7 アトリビュートエディタの画面構成



1. キャンバス
2. ペンの色 (1677万色以上)
3. ペンの太さ (15段階)
4. 色の透明度 (64段階)
- 5-7. ペンの色のRGB成分の数値表示 (各256段階)
- 8-17. カラーパレット (10枚×10種、計100枚)
18. モードおよびタイプの表示
- 19-21. グラデーションバー1
- 22-24. グラデーションバー2

図8 テクスチャ定義システムの画面構成

G_p, B_p とするとき、ブラッシング後の色R_i' , G_i' , B_i' は次式で表わされる。

$$\begin{bmatrix} R_i' \\ G_i' \\ B_i' \end{bmatrix} = e \begin{bmatrix} R_p \\ G_p \\ B_p \end{bmatrix} + (1-e) \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} .$$

但し、0 ≤ i ≤ n-1, 0 ≤ e ≤ 1 .

上式で e はブラッシング効果係数と呼ばれるもので、この値が大きいくほど透明度は小さくなる。この値はスイッチボックスのスイッチを押し、連続的に増減できるようになっている。

一方、拡散型は色を滑らかに変化させたいときに用い、ちょうど絵の具をキャンバスの上でのばすような動きをする。この場合は、注入型のペンの色R_p, G_p, B_pの代わりに、その範囲内の色の平均値を用いたもの

と考えることができ、次式のようになる。

$$\begin{bmatrix} R_i' \\ G_i' \\ B_i' \end{bmatrix} = \frac{e}{n} \sum_{j=0}^{n-1} \begin{bmatrix} R_j \\ G_j \\ B_j \end{bmatrix} + (1-e) \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix}$$

但し、 $0 \leq i \leq n-1$, $0 \leq e \leq 1$.

この場合の e もブラシング効果係数と呼んでおり、この値を小さくしてブラシングすると、きめの細かいグラデーションを得ることができる。このブラシングによって、立体感のある絵が容易に画ける。

以上のような機能により、試行錯誤的な作画が容易にできる。その他、表面の微小な向きの変化を貼り付けることにより、表面がざらざらした物体や、表面に凹凸のある物体を表わす法線マッピング⁸⁾用のデータ作成にも使われる。

5. アニメーションシステム

4章で述べた手法によりキーフレームを定義し、そのデータを補間することにより、全フレームのデータを作成するわけであるが、作成したデータの検証を対話的に行なうのは高品位画像では画像生成に時間がかかり、不可能である。そこで、低解像度もしくは線画による実時間表示で動きなどの確認を行なった後、高品位画像による収録を行なうことにした。その確認のためのシステムがアニメーションシステムである。

図9にその構成を示す。予めメモリ内に多数枚の画像を画いておき、表示の時にはバススイッチをDMAC側にし、メモリの領域を次々と切り換えて表示する。即ち、ラスタスキャンディスプレイで実時間動画表示が可能である。表示画素は 256×256 で、画素当たり白黒16階調表示の場合160枚、白黒2値表示なら640枚分の画像を5メガバイトの表示メモリに書えることができる。毎秒30枚表示すると、前者は5.3秒、後者は21.3秒間の動画表示ができることになる。また、表示速度は任意に変えることができる。

メモリに書き込む画像は、キーフレームエディタから得たデータを補間し画像生成システムにより並列処理で生成し、メモリ交換ユニット²⁾を介して得られる。並列処理を行なうことで画像(主として線画)の生成時間を大幅に短縮することができる。表示画像の生成はソフトウェアで行なうため、自由度が高い。即ち、

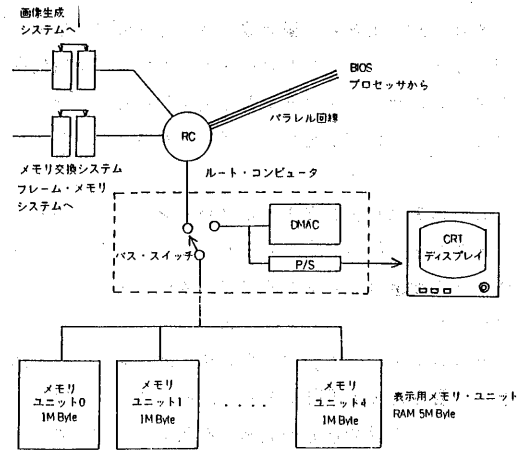


図9 アニメーションシステムのハードウェア構成

物体の運動やカメラの移動の検証だけでなく、物体の変形の検証も同時に行なうことが可能である。

また、市販のマイコンのVRAM方式の拡張であるから、画像表示用のメモリは他の用途にも使用できるので、他のシステムと比較してはるかにコストパフォーマンスが高く、ベクタスキャンタイプのシステムでは不可能な白黒階調表示もできる。

6. むすび

以上、LINKS-1における動画制作の手法と支援システムについて述べた。いずれのシステムにおいても最も考慮されることは使いやすさという点である。そのため実際にアーティストたちに使用してもらい、その意見を絶えず反映させるように努めている。しかし、アーティストによっては意見の食い違う場合もあり、また実際に使用してみないと使用感というものはわからない場合が多い。これらをうまく解決し、さらに使いやすくてローコストなシステム作りが今後の課題である。

また、意外に多いのが形状変形を伴う作品である。形状変形は、キーフレーム方式で動画を作成すると、フレーム数だけ物体データが必要になってしまう場合がある。その形状変形のためのデータ作成とその検証の高速化と容易化も今後の課題の1つである。

参考文献

- 1) 西村 他, "LINKS-1:コンピュータグラフィックスシステム"
- 2) 中山 他, "画像生成用マルチマイクロコンピュータシステム"
- 3) 河合 他, "画像データ操作システムLINKS-DMS"
- 4) 吉村 他, "LINKS-1 における画像生成手法"
以上 情報処理学会マイコン研究会資料
24-1,2,4,5 1982, 11, 2
- 5) 大野 他, "汎用BIOSマシン", 電気学会情報処理研究会資料 1982, 4
- 6) T. Whitted, "An Improved Illumination Model for Shaded Display", Comm. ACM, vol. 23, No. 6, pp.343-349, June 1980
- 7) J. F. Blinn, et al., "Texture and Reflection in Computer Generated Images", Comm. ACM, vol. 19, No. 10, pp.542-547, Oct. 1976
- 8) J. F. Blinn, "Simulation of Wrinkled Surfaces", Proc. SIGGRAPH '78, Vol. 12, No. 3, pp.286-292, 1978
- 9) J. C. Ferguson, "Multivariable Curve Interpolation", J. ACM, Vol. no.2, pp.235-242, April 1964
- 10) D. F. Rogers, J. A. Adams, "Mathematical Elements for Computer Graphics", McGraw-Hill, 1976