

# フラクタル幾何学の造形美 (VI)

## —高解像度画像の作成—

写真学科

米 川 靖 彦

### 1. はじめに

筆者はパーソナルコンピュータを使用してフラクタル画像の研究をしている<sup>(1)(2)(3)</sup>。しかしながら作成した画像は640×400ドットの低解像度の画像しか出来て来ない。これを作品として展示するには無理が生じてくる。今回情報処理センターに導入された UNIX システムである富士通 S ファミリーに 3 次元 CG ソフトと 2 次元 CG ソフトが組み込まれている。

2 次元 CG ソフトを利用すれば高解像度の画像とすることが出来る。また、表示された色は簡単に変えることが出来る。これらの機能を利用して 1920×1600ドットの高解像度画像をハードコピーとして4×5フィルムへの出力を試みた。マンデルプロ集合やジュリア集合などの考え方などは前の報告<sup>1)(2)(3)</sup>と重複するので今回は省いている。

### 2. 主な使用機器及びソフト

- (1) 画像作成用パーソナルコンピュータ  
日本電気製 PC9801RA2 に数値演算プロセッサを付加
- (2) フレームバッファ及び双方向インターフェイス  
リンクス製の PFB-1 と双方向インターフェイスを上記のパーソナルコンピュータに取付
- (3) UNIX マシン  
富士通製の S-4/260C
- (4) デジタルフィルムレコーダ  
MATRIX 製の QCR-Z フィルムレコーダーを PC9801VM に取付
- (5) 光磁気ディスク

ソニー製光磁気ディスク NWP-539 を上記の PC9801VM に取付

- (6) 2次元画像作成用プログラム  
リンクス製の 2次元画像作成プログラム PICTORIS
- (7) その他のプログラム  
リンクス製の CG プログラムユーティリティ
- (8) フィルムレコーダーコントロールプログラム  
リンクス製の QCR-Z コントロールプログラム

### 3. プログラム (画像生成用プログラム)

画像生成には Turbo PASCAL (Ver 5.0) を言語に使用した。list-1, list-2, list-3 のプログラムを使用した。list-1 は画像作成プログラムであり、list-3 は表示プログラムである。list-2 は画像作成と表示のどちらにも共通のプログラムである。このプログラムは京都大学に勤務されている宇敷重広氏のプログラム (Turbo PASCAL Ver 3.0 用) に手を加えて12枚の画像を fig-1 のように連続して作成するようにしてある。このプログラムの欠点は作成した画像の実数軸、虚数軸の入力した範囲が分からないことである。したがってデータは別に記録しておかなければならない。

Turbo PASCAL は Ver 4.0からは IBM と NEC の機種に依存するようになった。その分、ユニットというモジュールを追加してプログラムし易いようになった。Ver 5.0は富士通 FMR シリーズにも対応した。

Ver 3.0ではグラフィックを行なうには inline 命令で行なう必要があったが、Ver 4.0以降ではグラフィックユニットの使用を宣言すればよい。ま

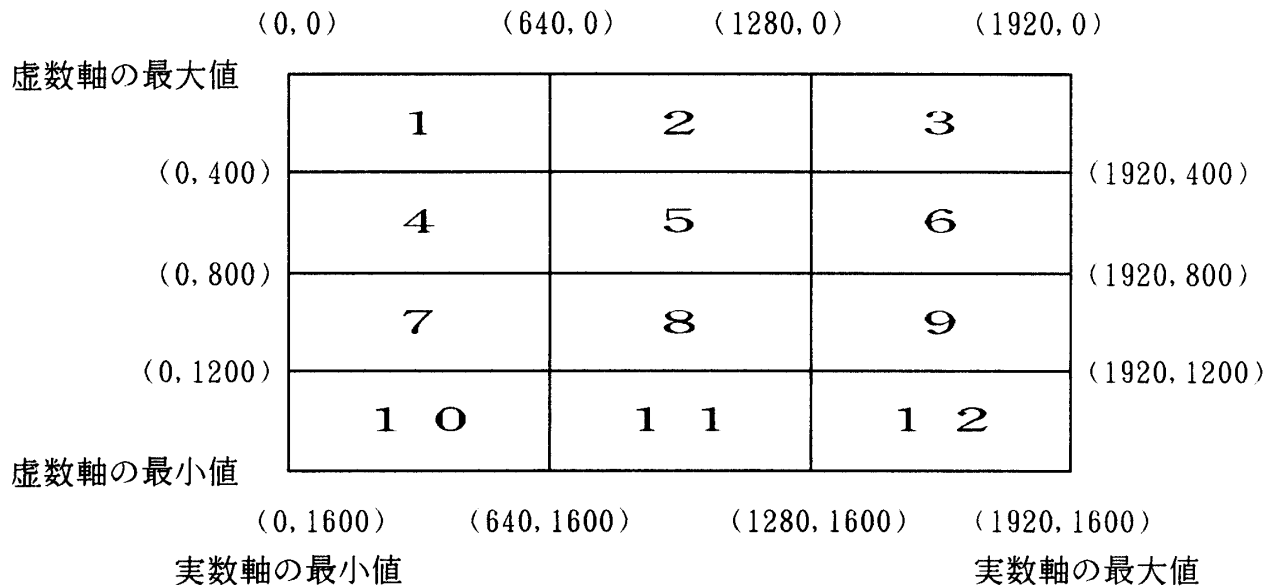


Fig. 1

た、グラフィック専用のコマンドも作成されており、プログラムをするには簡単となった。

#### 4. 作品制作

作品制作におけるフローチャートにして fig-2 に示す。

それぞれの手順を順を追って説明するが、コマンドなどの細かいパラメーターの与え方などについてはレファレンスマニュアル等を参照して下さい。

##### (1) 画像生成

作品制作はパソコン上で640×400ドットの画像を12枚作成する。作成した画像データはパソコン上の VRAM のデータを直接記録したものである。画像ファイルの大きさは640×400ドットの画像1枚で約130 Kbyteである。このデータをリンクス製のフレームバッファに転送をし、リンクス形式の PIC ファイルとしてセーブする(リンクスより提供された v12.exe を使用)。ファイルの画像データ(runlength タイプ)は圧縮してあるので不定である。

##### (2) 画像転送

イーサネット上の TCP/IP プロトコルの ftp コマンドを使用してバイナリーデータを S-4 に転送する(ftp hostname [ ] binary [ ] put

filename [ ] )。

##### (3) 画像の繋ぎ合せ

S-4 にログインした後、piccat コマンドで画像を繋ぎ合わせるが(piccat filename (.pic) [ ]) 繋合わされた画像データ(pixel タイプ)は非常に大きく、9.2 MByte 超となっている。

##### (4) 画像色の変更

紫から赤への虹色を360度の円盤上に置き、120度ずつ回転して色の変更を行なう。3回ほど回転させれば元の色に戻る。また、S-4 に搭載しているフレームバッファ FMX-32 は1024×1024ドットの大きさのため、1920×1600ドットの画像色は1度に変更できない。そこで1枚ずつ画像の色を変更しておいた後で繋ぎ合せを行なう。

④ バックグラウンドで qserv ジョブを起動しておく(qserv & [ ])。

④ 色円盤を120度回す(qhue [ ])。回す角度は任意でもよい(qhue x x x [ ])。

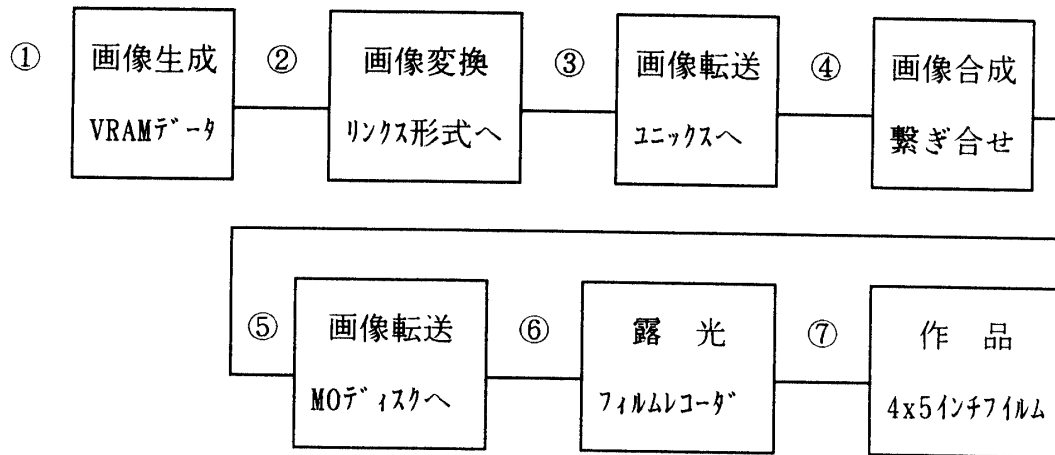
④ 12枚全部の画像の色が変換できたならば、繋ぎ合せを行なう(piccat filename [ ])。

④ 表示色の補色にする(qnega [ ])。

④ 色を120度回す(qhue [ ])。

④ マンデルブロ集合の黒の部分が白く反転するので黒に変換する(qcconv x x x x x x x x

画像の色変換を行わない場合



画像の色変換を行なう場合

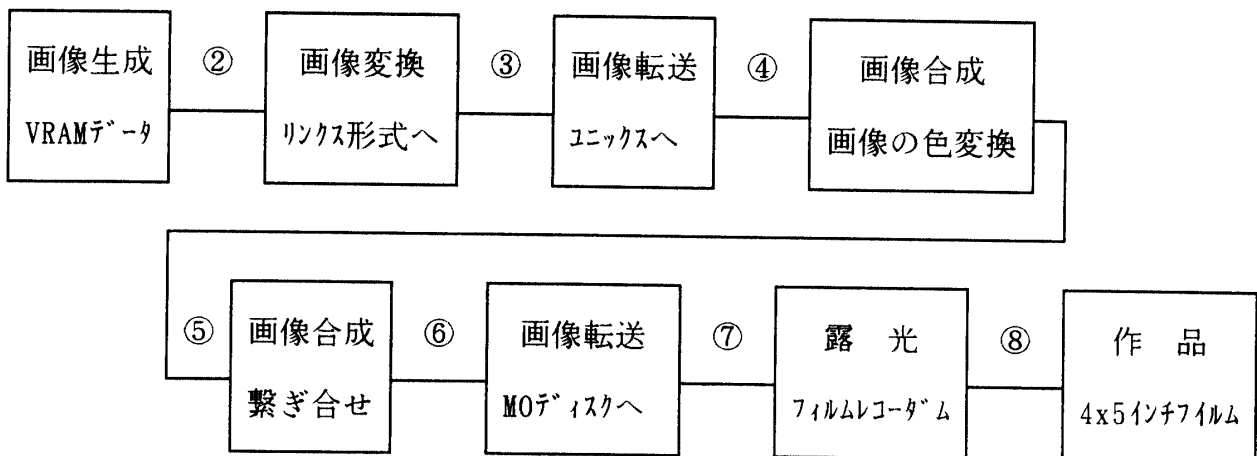


Fig. 2

□)。

◇12枚全部の画像の色が変換できたならば、繋ぎ合せを行なう (piccat filename□)。  
 このようにして(3)で作成した1枚の画像(1920×1600ドットから色円盤を120度及び240度回転させた画像と反転させた画像、色円盤を120度及び240度回転させた画像とで6枚の画像を作成し、同じ図柄の色違い画像を1セットとした。

(5) 光磁気ディスクへの画像転送

デジタルカメラはPC9801VMに接続されており、S-4から直にコントロールできない。pic形式の画像が640×480ドットであり、1,670万色の画像の大きさは約1MByteとなるためこれ以上の大

きさのデータはフロッピーディスクでは取り扱えない。このPC9801には光磁気ディスクが接続されており、大量のデータが取り扱えるようになっている。前記(2)の画像転送とは逆に、繋ぎ合わせた画像はS-4から光磁気ディスクへ戻さなければならない(get filename□)。大容量ハードディスクでもよいのだが光磁気ディスクの方が容量が大きい。

(6) フィルムへの露光

PC9801VMに接続されているQCR-Zカメラでフィルムの露光を行なう(qcrrec□ rec filename□)。

QCR-Zカメラはデフォルトで2048×2048

l i s t — 1

```

Program Mandelbrot;

{$IFDEF CPU87}
{$N+}
{$ELSE}
{$N-}
{$ENDIF}

uses Crt,
      Dos,
      Graph;

{$I GRLIB1.PAS}

Type Complex = Record re, im : double end;

var Wd, a, x0, x1, x2, x3, z1, z2, XNC : Complex;
    rr, Wmin : extended;
    GraphDriver, Graphmode, Detect, lcc, i : integer;
    Ch1, Ch2 : Char;
    FileNmRad1, FileNmRad, TodayDate, si : String14;
    Wminre : array [1..12] of real;
    Wmaxre : array [1..12] of real;
    Wminim : array [1..12] of real;
    Wmaxim : array [1..12] of real;

Procedure Ingr;
begin
  GraphDriver := InstallUserDriver('PC98', @Detect);
  GraphDriver := detect;
  InitGraph(Graphdriver, Graphmode, '');
end;

Procedure WindowSet;
begin
  clrscr;
  writeln('Region');
  {writeln(' AX : ', Wmin.re:10:7, ' : ', Wmax.re:10:7);
  writeln(' AY : ', Wmin.im:10:7, ' : ', Wmax.im:10:7);}
  write(' AXmin = '); Readln(Wminre[1]);
  write(' AXmax = '); Readln(Wmaxre[12]);
  write(' AYmin = '); Readln(Wminim[1]);
  write(' AYmax = '); Readln(Wmaxim[12]);
  Wd.re := (Wmaxre[12] - Wminre[1]) / 3;
  Wd.im := (Wmaxim[12] - Wminim[1]) / 4;
  Wminre[2]:=Wminre[1] + Wd.re;
  Wminre[5]:=Wminre[1] + Wd.re;
  Wminre[8]:=Wminre[1] + Wd.re;
  Wminre[11]:=Wminre[1] + Wd.re;

```

```

Wminre[3]:=Wminre[1] + Wd. re * 2;
Wminre[6]:=Wminre[1] + Wd. re * 2;
Wminre[9]:=Wminre[1] + Wd. re * 2;
Wminre[12]:=Wminre[1] + Wd. re * 2;
Wminre[4]:=Wminre[1];
Wminre[7]:=Wminre[1];
Wminre[10]:=Wminre[1];

Wmaxre[1]:=Wminre[1] + Wd. re;
Wmaxre[4]:=Wminre[1] + Wd. re;
Wmaxre[7]:=Wminre[1] + Wd. re;
Wmaxre[10]:=Wminre[1] + Wd. re;
Wmaxre[2]:=Wminre[1] + Wd. re * 2;
Wmaxre[5]:=Wminre[1] + Wd. re * 2;
Wmaxre[8]:=Wminre[1] + Wd. re * 2;
Wmaxre[11]:=Wminre[1] + Wd. re * 2;
Wmaxre[3]:=Wmaxre[12];
Wmaxre[6]:=Wmaxre[12];
Wmaxre[9]:=Wmaxre[12];

Wmaxim[1]:=Wmaxim[12];
Wmaxim[2]:=Wmaxim[12];
Wmaxim[3]:=Wmaxim[12];
Wmaxim[4]:=Wmaxim[12] - Wd. im;
Wmaxim[5]:=Wmaxim[12] - Wd. im;
Wmaxim[6]:=Wmaxim[12] - Wd. im;
Wmaxim[7]:=Wmaxim[12] - wd. im * 2;
Wmaxim[8]:=Wmaxim[12] - wd. im * 2;
Wmaxim[9]:=Wmaxim[12] - wd. im * 2;
Wmaxim[10]:=Wmaxim[12] - wd. im * 3;
Wmaxim[11]:=Wmaxim[12] - wd. im * 3;
Wmaxim[12]:=Wmaxim[12] - wd. im * 3;

Wmin:=Wminim[1];
Wminim[1]:=Wmin + Wd. im * 3;
Wminim[2]:=Wmin + Wd. im * 3;
Wminim[3]:=Wmin + Wd. im * 3;
Wminim[4]:=Wmin + Wd. im * 2;
Wminim[5]:=Wmin + Wd. im * 2;
Wminim[6]:=Wmin + Wd. im * 2;
Wminim[7]:=Wmin + Wd. im ;
Wminim[8]:=Wmin + Wd. im ;
Wminim[9]:=Wmin + Wd. im ;
Wminim[10]:=Wmin ;
Wminim[11]:=Wmin ;
Wminim[12]:=Wmin ;
end;

Procedure FileAssign;
begin
  write(' FileNameRadical = '); Readln(FileNmRad1);

```

```

    {write(' TodayDate = 1990/' );readln(TodayDate);}
  ClrScr;
end;

Procedure MandelbrotPicture;
var lx, ly, lcol, IYT, IXT, K, IMAX : integer;
    Axd, Ayd, EPSI, zz : extended;
Label PA;

Procedure CheckInterruption;
begin
  if KeyPressed then
    begin
      cleardevice;closegraph;Halt;
    end;
end;

begin
  IMAX := 3000; EPSI :=4.0;
  for i:=1 to 12 do
    begin
      str(i+9, si);si:=si;
      FileNmRad := concat(FileNmRad1, si);

  Axd :=(Wmaxre[i] - Wminre[i]) / 640; Ayd := (Wmaxim[i] - Wminim[i]) / 400;
  for IYT := 1 to 400 do
    begin
      a.im := Wminim[i] + Ayd * IYT ; ly := 400 - IYT;
      for IXT := 0 to 640 do
        begin
          IX := IXT;
          lcol := 7 ; PutPixel(lx, ly, lcol);
          a.re := Wminre[i] + Axd * IXT ; x0.re := a.re; x0.im := a.im;
          for K := 1 to IMAX do
            begin
              x2.re := sqr(x0.re) - sqr(x0.im) + a.re;
              x0.im := 2.0 * x0.re * x0.im + a.im; x0.re := x2.re;
              zz := sqr(x0.re) + sqr(x0.im);
              if zz > EPSI then
                begin
                  if k <= 20 then
                    begin
                      lcol := (k mod 2)*5 + 1; goto PA;
                    end;
                  if k <= 50 then
                    begin
                      lcol := (k mod 2)*5+9; goto PA;
                    end;
                  if k <= 140 then
                    begin
                      lcol := (k mod 2)*3+2; goto PA;
                    end;
                end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```
    end;
  if k <= 270 then
    begin
      Icol := (k mod 2)*12+1; goto PA;
    end;
  if k <= 540 then
    begin
      Icol := ((k div 3) mod 2)*3+ 3; goto PA;
    end;
  if k <= 750 then
    begin
      Icol := 13; goto PA;
    end;
  if k <= 1000 then
    begin
      Icol := ((k div 4) mod 2)*11 + 4; goto PA;
    end;
  if k <= 1300 then
    begin
      Icol := ((k div 5) mod 2)*2 + 1; goto PA;
    end;
  if k <= 2000 then
    begin
      Icol := 13; goto PA;
    end;
  if k <= LMAX then
    begin
      Icol := 14;goto PA;
    end;
  end;
  Icol := 0;
PA:  PutPixel(lx, ly, Icol);
    end;
    Checkinterruption;
  end;
  grsave(FileNmRad);
  cleardevice;
end;
end;

begin
  Ingr;
  cleardevice;
  Windowset;
  Fileassign;
  MandelbrotPicture;
  closegraph;
end.
```

l i s t — 2

```
{ This file is for PC-9801. }
```

```
Const SegBase1 = $A800; SegBase2 = $B000; SegBase3 = $B800; SegBase4 = $E000;
```

```
Procedure wait;  
begin Repeat until KeyPressed; end;
```

```
Type String14 = String[14];
```

```
Procedure GrLoad(var FNameRad : String14);  
var FName : String14; GrFile : File;  
begin  
  clrscr;  
  FName := concat(FNameRad, '.IMG');  
  
  Assign(GrFile, FName); Reset(GrFile);  
  BlockRead(GrFile, Mem[SegBase1:0], 255);  
  BlockRead(GrFile, Mem[SegBase2:0], 255);  
  BlockRead(GrFile, Mem[SegBase3:0], 255);  
  BlockRead(GrFile, Mem[SegBase4:0], 255);  
  close(GrFile);  
end;
```

```
Procedure GrSave(var FNameRad : String14); {modified for PC-9801 }  
var FName : String[14]; GrFile : File;  
begin  
  FName := concat(FNameRad, '.IMG');  
  Assign(GrFile, FName); Rewrite(GrFile);  
  BlockWrite(GrFile, Mem[SegBase1:0], 255);  
  BlockWrite(GrFile, Mem[SegBase2:0], 255);  
  BlockWrite(GrFile, Mem[SegBase3:0], 255);  
  BlockWrite(GrFile, Mem[SegBase4:0], 255);  
  close(GrFile);  
end;
```



l i s t — 3

```

Program Grout;

uses dos, crt, graph;

{$I GRLIB.PAS}

var fileNmRad : string14;
    GraphDriver, Graphmode : integer;

begin
  Graphdriver := installuserdriver('PC98', @detect98);
  Graphdriver := detect;
  InitGraph(GraphDriver, Graphmode, '');
  write ('Graphic file name = ');
  readln(FileNmRad);
  if FileNmRad <> '' then Grload(FileNmRad);
  Wait; cleardevice; closeGraph;
end.

```

ドットのモードで露光するように設定されている。設定によっては4096×4096ドットもできる。また、640×480ドットの大きさの画像を2048×2048ドットで露光すると中心部に小さく露光されることになる。rec サブコマンドにはパラメーターがついている。step [x, y]=1 1?と聞いてくるので3 $\square$ とすれば縦3倍横3倍の画像がフィルムへ露光される。オフセットはそのまま $\square$ を入力すると、出力される画像はフィルムの中心で露光される。

以上のようにして作品を制作した。出力された画像はほぼCRTの縦横3倍づつの大きさに引き伸ばしをしても鑑賞に耐え得るものとなっている。

## 5. まとめ

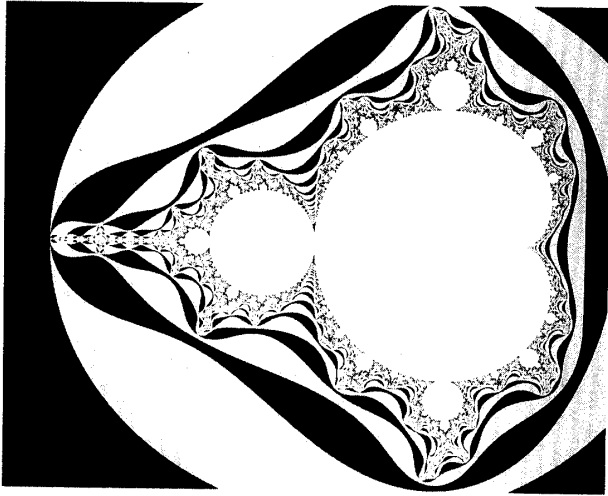
マンデルブロ集合の高解像度写真としてフィルムに露光することを目的として作品制作を行なったが、非常に品質のよいフィルムが出来た。picture-1~picture-16までの作品を提示している。前の報告<sup>1)2)3)</sup>において、作品はCRT画面をそのまま撮影していたので掲載された作品上にモアレ模様が出ていた。しかし、今回はライティングスポットデジタルカメラで4×5インチのフィルムに露光

したものを複写したネガを使用したためモアレが消えている。また、同じ図形の作品において、色の付け方によってかなり受ける印象が変化する。

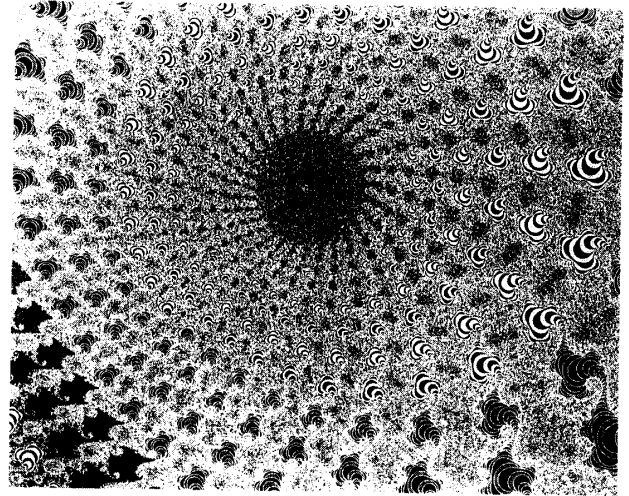
出来た画像の品質はこの誌面の上では分からないが、非常によいものとなっている。しかし、作成する画像は1920×1600ドットの大きさのため、12枚の画像を作成するにも平均2~3日を必要とした。中には5日くらい必要としたものもある。また繋ぎ合わせた画像は9.2 Mbyte超であるためMOディスクへの転送にも平均12分くらい必要であり、露光には約30分が必要であった。このように1枚の4×5インチのフィルム画像を作成するのに非常に多くの時間が必要であった。

ただ、12枚の画像を繋ぎ合わせることにより、細部のディテールがフィルム上に非常に肌理細かく記録されているから、この画像を拡大しても十分鑑賞に耐えるものとなっている。例えば、このフィルムを全紙の大きさに引き伸ばすためのカラープリント処理を行なっても十分に鑑賞に耐え得るものとなっている。従って、今回の初期の目的である高解像度画像の作成については非常によい成績を得ることが出来た。

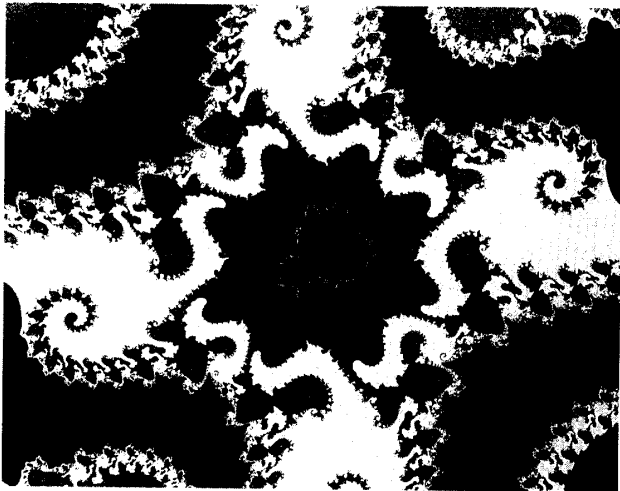
No	実数軸の範囲 Wmin. re ~ Wmax. re	虚数軸の範囲 Wmin. im ~ Wmax. im
1	-2.3000000 ~ .7000000	-1.2500000 ~ 1.2500000
2	-.7458000 ~ -.7462400	-.1067817 ~ -.1078983
3	-.7454356 ~ -.7454210	.1130029 ~ .1130147
4	-.7454356 ~ -.7454210	.1130029 ~ .1130147
5	-.7454680 ~ -.7453850	.1030340 ~ .1129840
6	-.7455380 ~ -.7450540	.1128568 ~ .1132602
7	-.7130000 ~ -.4082000	-.4762250 ~ -.7302250
8	-.7465410 ~ -.746378	.1075436 ~ .1076780
9	-.7510400 ~ -.7408000	.1059683 ~ .1145017
10	-.7510400 ~ -.7408000	.1059683 ~ .1145017
11	-.1992000 ~ -.1295400	-1.0119100 ~ -1.0699600
12	-1.7810000 ~ -1.7640000	0.0000000 ~ -.01300000
13	-1.2540240 ~ -1.2528610	.0462039 ~ .0471731
14	-1.7114000 ~ -1.7113500	.0000876 ~ .0001084
15	-.7459100 ~ -.7444800	.1120792 ~ .1132708



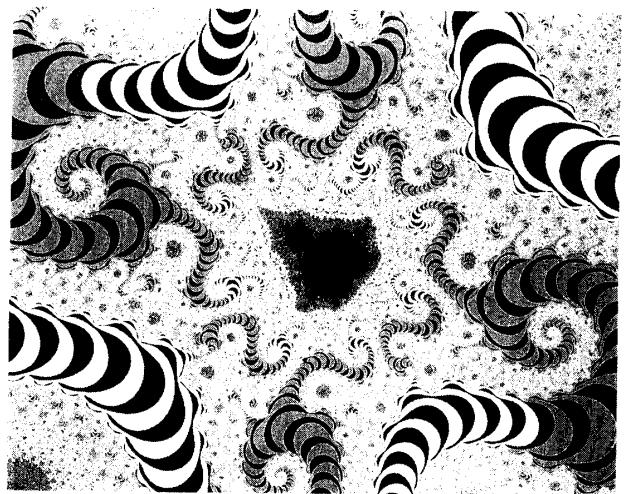
picture-1



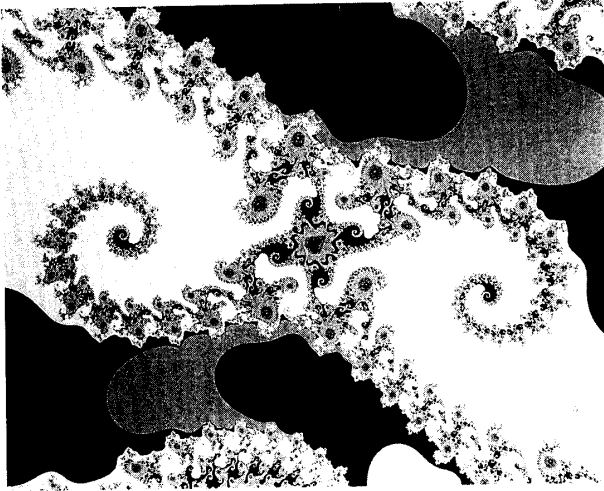
picture-2



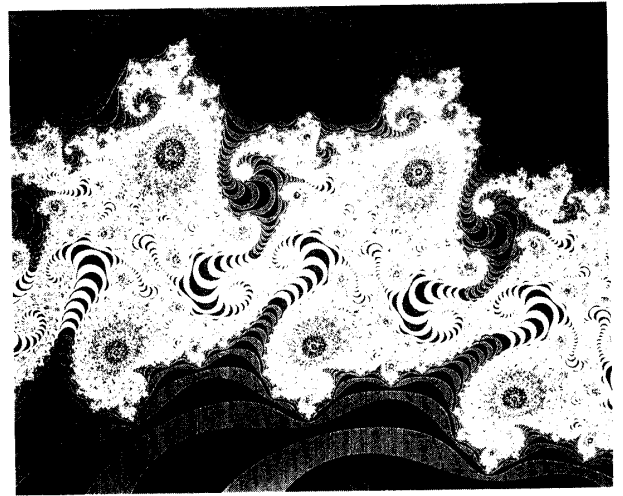
picture-3



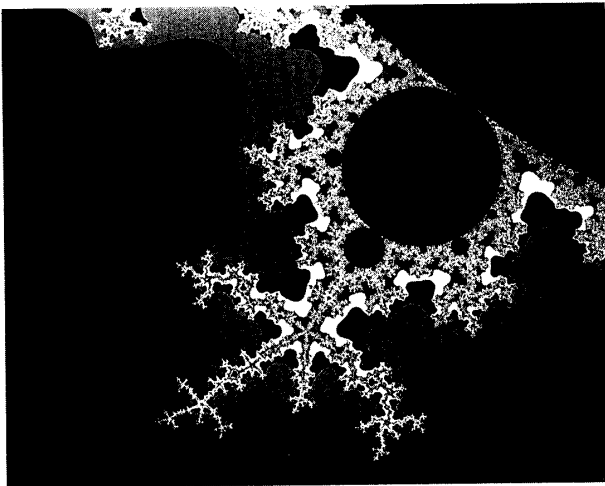
picture-4



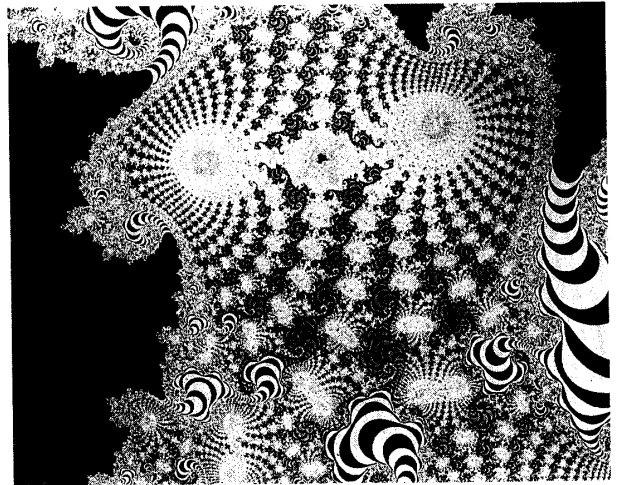
picture-5



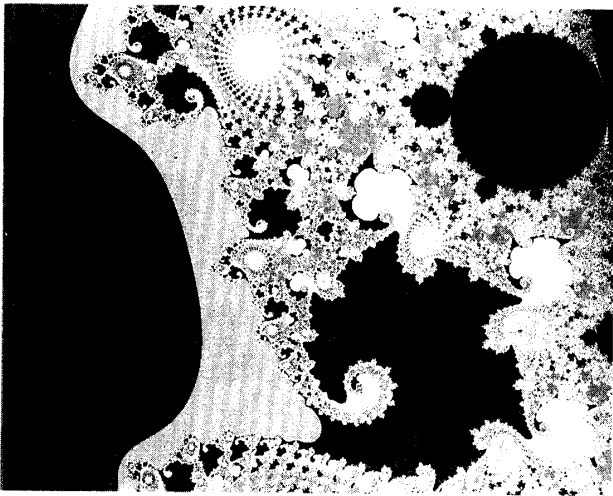
picture-6



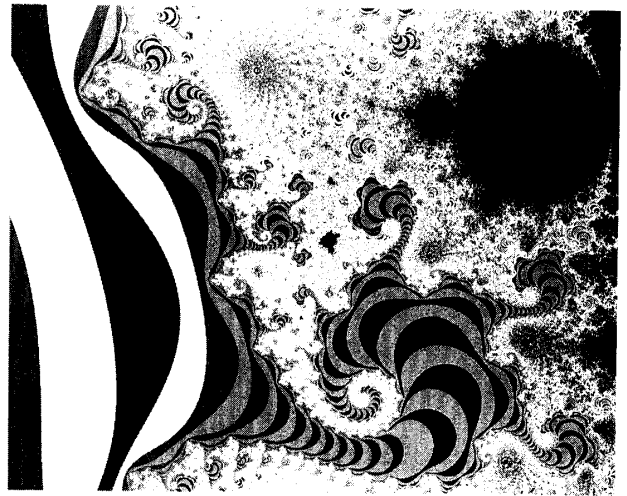
picture-7



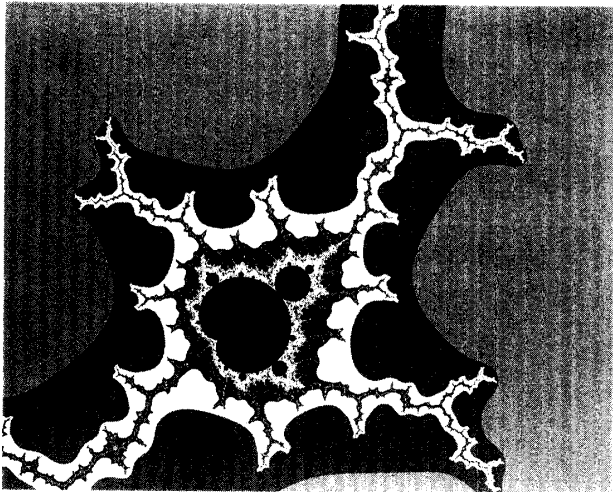
picture-8



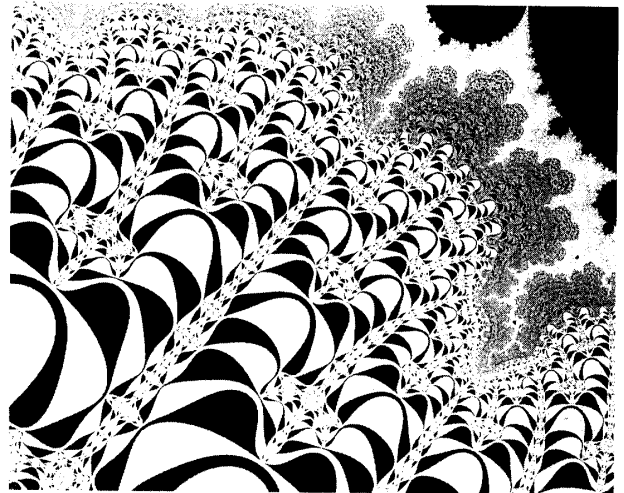
picture-9



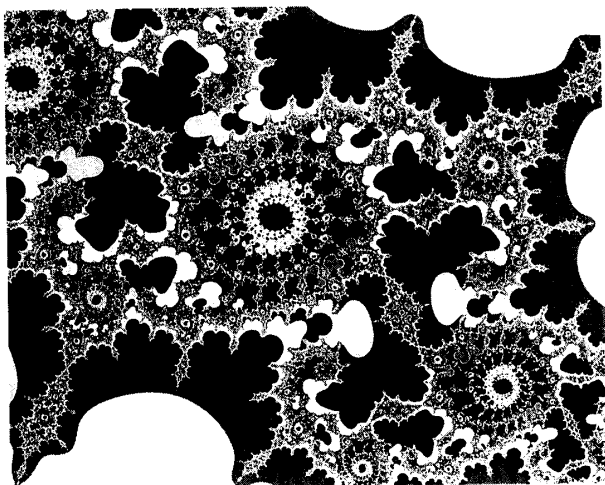
picture-10



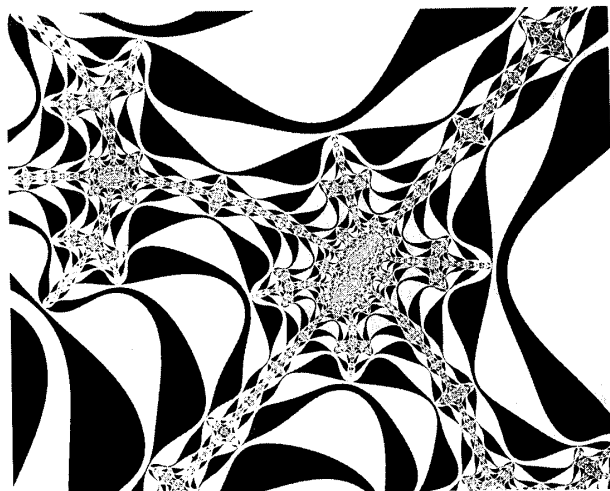
picture-11



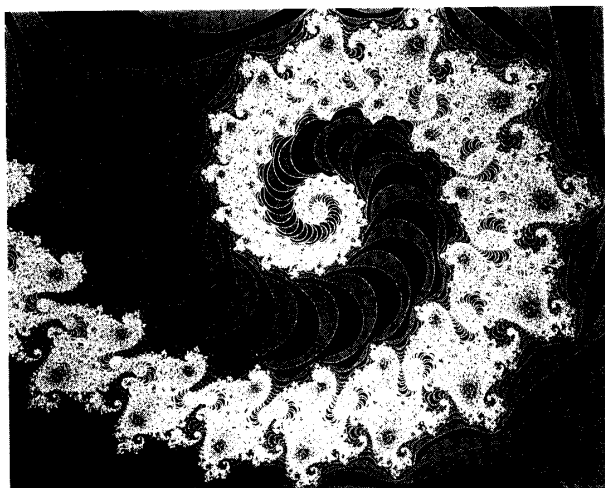
picture-12



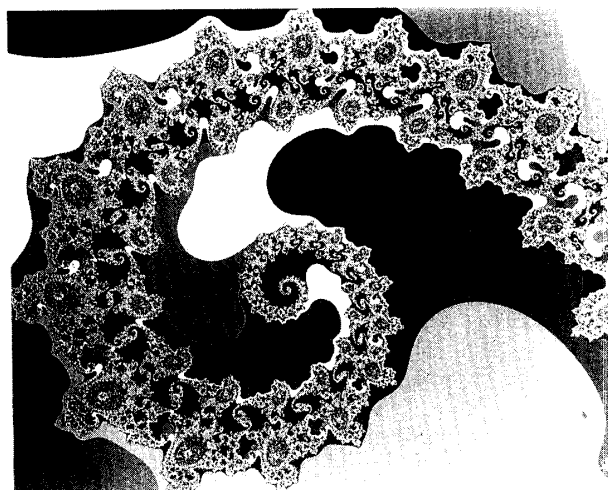
picture-13



picture-14



picture-15



picture-16

### 参考文献

- 1) 米川靖彦：フラクタル幾何学の美(Ⅰ)，九州産業大学芸術学部研究報告，Vol. 20, p. 155-162 (1989, 2)
- 2) 米川靖彦：フラクタル幾何学の美(Ⅱ)，九州産業大学芸術学部研究報告，Vol. 20, p. 162-170 (1989, 2)
- 3) 米川靖彦：フラクタル幾何学の美(Ⅲ)，九州産業大学芸術学部研究報告，Vol. 21, p. 77-88 (1990, 2)