

## 9.3 ランダムドット

### (1) 立体視とは

3次元的な物体を2次元平面上に描いて、その2次元平面の図形から元の3次元物体を再構成して視ることを立体視といいます。通常は、左目で見える図形と右目で見える図形を別々に見ることで立体視します。ステレオカメラで写した写真を、ステレオスコープで見ることも立体視のひとつです。

漫画等の世界では、図9-7のような色眼鏡を使って、立体視することもあります。これは、青いフィルムを通して見ると青色が消え、赤色のところだけが見え、赤いフィルムを通してみると赤色が消え、青色のところだけが見えることを利用したものです。

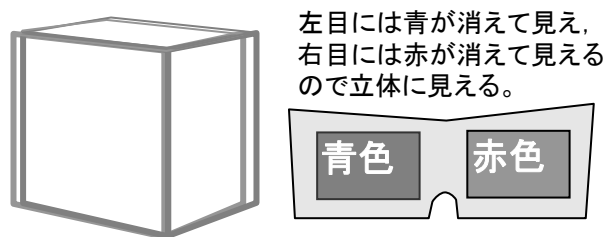


図 9-7 色眼鏡による立体視

### (2) ランダムドット・ステレオグラムとは

ランダムドット・ステレオグラムとは1枚の図で立体視が可能な絵のことです。3次元図形上のランダムな点に対して、左目で見える箇所と右目で見える箇所に点が描かれている絵ですが、焦点をずらしてこの絵を見ると立体に見えます。

ここでは、この絵を描くプログラムを示します。プログラム中では、見えている物体面が紙面に対して平行であると仮定し、ある基準となる面から物体面の距離を  $d$  として、紙面上の間隔  $T$  を以下の式で近似しています。

$$T = C - R \cdot d$$

ここで、 $C$  : 物体が基準となる位置にある場合に対応する紙面上の点の間隔、 $R$  : ずらすための定数(値が大きいとき、飛び出し方や引っ込み方が大きくなる)です。

## (2)ランダムドット・ステレオグラムの見方

ランダムドット・ステレオグラムの上方に、以下の図 9-8(a)の状態の2つの点があります。

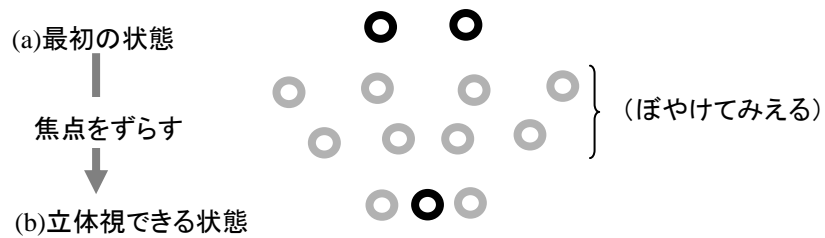
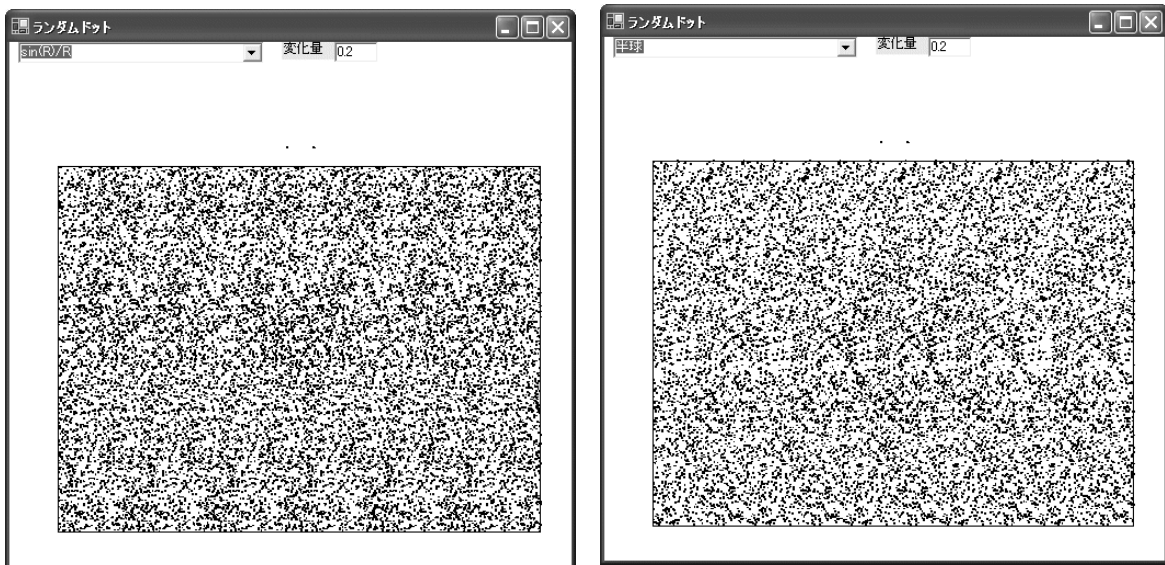


図 9-8 ランダムドット・ステレオグラムの上の点

目をリラックスさせた状態で紙面に視線を向け、平行法の場合は、紙面より先のほうへ、交差法の場合は、紙面より手前のほうに焦点を合わせるように見ると、これがぼやけて見えます。

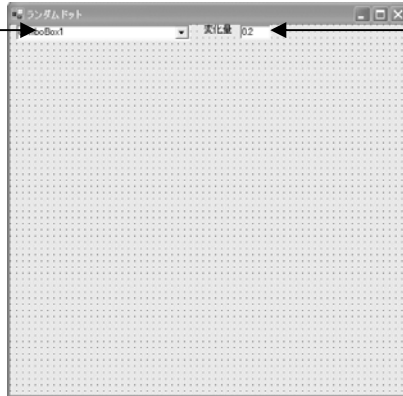
ぼやけた円が 2 個、中央にはっきりした円が 1 個に見えるように目の焦点をずらすと((b)の状態)、ステレオグラムが立体に見えてきます。経験のない方は、以下の例でやってみましょう。この例は、後で示すプログラムで描いたものです。



### (3) プログラムの説明

#### ■フォームの定義

Name : comboBox1



Name : textBox1

#### ■プログラム

2次元の絵を描くことになりますので、以下の `using` を追加しておきます。

```
using System.Drawing.Drawing2D;
```

#### 【データ宣言】

```
public Matrix matrix = new Matrix();
public string 処理;
```

#### 【表示用関数】

これらの関数は、XY座標を渡したらZ座標値を返します。もし、追加したい関数があれば自由に追加しましょう。

```
private double SinR_dev_R(double X, double Y)
{ double R = Math.Sqrt(X * X + Y * Y); if(R <= 0.0000001) return 1.0;
  return Math.Sin(R)/R;
}
private double 半球(double X, double Y)
{ double R = Math.Sqrt(X*X+Y*Y);
  if(R>5) return -1.0; else{R = R / 5; return Math.Sqrt(1 - R * R) - 1;}
}
private double 円筒(double X, double Y)
{ double R = Math.Abs(X);
  if(R>5) return -1.0; else{R = R / 5; return Math.Sqrt(1 - R * R) - 1;}
}
```

## 【関数の呼出し】

追加関数があれば、case と return 文を追加します。

```
private double 関数呼出(double X, double Y)
{ int ID=comboBox1.SelectedIndex;
  switch(ID)
  { case 0:return SinR_dev_R(X,Y);
    case 1:return 半球(X,Y);
    case 2:return 円筒(X,Y);
  }
  return 0;
}
```

## 【ドット表示の本体】

```
public void ドット表示(PaintEventArgs e)
{ double X1 = -12; double X2 = 12;
  double Y1 = 10; double Y2 = -10;
  double UNIT = 6; double XY = 1; int Loop = 3000;
  double R0 = double.Parse(textBox1.Text);
  double Period = (X2-X1)/6; double Z0 = R0*Period;
  double R, X, Y, Z;
  Pen pen = new Pen(Color.Black, 0.001F);
  Brush brush = new SolidBrush(Color.Black);
  e.Graphics.Clear(Color.White);
  float XX=(float)(Period/6.0); float YY=(float)(Y1*1.1);
  e.Graphics.DrawEllipse(pen, -XX, YY, 0.1F, 0.1F);
  e.Graphics.DrawEllipse(pen, XX, YY, 0.1F, 0.1F);
  Random RD = new Random();
  for(int i=0; i<Loop; i++)
  { R = RD.NextDouble(); Y = R * (Y2 - Y1) + Y1;
    Z = 関数呼出(X1, Y);
    R = RD.NextDouble();
    X = (Period + Math.Abs(Z0 * Z)) * R + X1;
    while(X<=X2)
    { double XP = (X - (X1 + X2) * 0.5) * UNIT;
      double YP = (Y - (Y1 + Y2) * 0.5) * UNIT;
      e.Graphics.FillRectangle(brush, (float)(X - 0.025),
        (float)(Y - 0.025), 0.1F, 0.1F);
      Z = 関数呼出(X + Period * 0.5, Y);
      X = X + Period - XY * Z0 * Z;
    }
  }
}
```

## 【OnPaint のオーバーライド】

```
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e); e.Graphics.Clear(Color.White);
    e.Graphics.Transform = matrix;
    if (処理!="") ドット表示(e);
    Pen pen = new Pen(Color.Black, 0.001F);
    e.Graphics.Transform = matrix;
    e.Graphics.DrawLine(pen, -12F, 10F, 12F, 10F);
    e.Graphics.DrawLine(pen, -12F, -10F, 12F, -10F);
    e.Graphics.DrawLine(pen, -12F, 10F, -12F, -10F);
    e.Graphics.DrawLine(pen, 12F, 10F, 12F, -10F);
}

```

## 【コンボボックス選択(再描画)】

```
private void comboBox1_SelectedIndexChanged
(object sender, System.EventArgs e)
{
    処理="Exe"; this.Invalidate();
}

```

## 【座標変換マトリックス設定】

```
private void window(float X1, float Y1, float X2, float Y2)
{
    float W = this.Width ; float H = this.Height;
    float SX = W / (X2 - X1) ; float SY = H / (Y2 - Y1);
    matrix.Scale(SX, SY); matrix.Translate(-X1, -Y1);
}

```

## 【初期設定】

追加関数があれば,

```
comboBox1.Items.Add("<関数を示す文字列>");
```

の形式で追加します。

```
private void Form1_Load(object sender, System.EventArgs e)
{
    処理="";
    window(-12F * 1.2F, 12F * 1.4F, 10F * 1.4F, -10F * 1.4F);
    comboBox1.Items.Clear();
    comboBox1.Items.Add("sin(R)/R");
    comboBox1.Items.Add("半球");
    comboBox1.Items.Add("円筒");
}

```