Course CVCis: Computer Vision with C#
Chapter C1: The Complete Code of the Cells Project

Main Menu after start of VS 2008: `File -> New Project... ->`
`Visual Studio installed templates: Windows Forms Application`
`Name: cells1 -> Location: C:\temp -> Create directory for solution`: switch off `-> OK`
Delete two superfluous files: `Form1.Designer.cs` and `Program.cs`.
Replace the pre-programmed code of `Form1.cs` by:

```
using System;
using System.Drawing;
using System.Windows.Forms;

public class Form1 : Form
{ static void Main() { Application.Run( new Form1() ); }
  const Int32 xSize = 11;
  const Int32 ySize = 12;
  Byte[,] i0 =  new Byte[ySize  ,xSize  ];
  Byte[,] c2 =  new Byte[ySize  ,xSize  ];
  Byte[,] c0 =  new Byte[ySize+1,xSize+1];
  Byte[,] c1v = new Byte[ySize  ,xSize+1];
  Byte[,] c1h = new Byte[ySize+1,xSize  ];
  Boolean C0, C1V, C1H, C2;
  Brush[] brush = new Brush[10];
  Int32 i, x, y, dx, dy;
  Byte threshold = 1;
  Button[] button = new Button[ySize];

  public Form1()
  { BackColor  = Color.White;
    Text       = "C2_C1V_C1H_C0";
    SetStyle( ControlStyles.ResizeRedraw, true );
    Width  = 800;
    Height = 600;
    for ( i=0; i < 10; i++ )
      brush[i] = new SolidBrush( Color.FromArgb( i*25, i*25, i*25 ) );
    for ( y=0; y < ySize; y++ )
    { button[y] = new Button();
      Controls.Add( button[y] );
      button[y].BackColor = Color.Gray;
      button[y].Text = "nothing";
      button[y].Name = y.ToString();
      button[y].Click += new EventHandler( do_it );
    }
    button[0].Name = button[0].Text = "Homunculus";
    button[1].Name =                  "Threshold";
    button[2].Name = button[2].Text = "Noise";
    button[3].Name = button[3].Text = "Clear";
    button[4].Name = button[4].Text = "C0-Cells";
    button[5].Name = button[5].Text = "C1v-Cells";
    button[6].Name = button[6].Text = "C1h-Cells";
    button[7].Name = button[7].Text = "C2-Cells";
    button[8].Name = button[8].Text = "C1v+C1h-Cells";
    button[9].Name = button[9].Text = "All-Cells";
    button[1].Text = String.Format( "Threshold={0:#}", threshold );
  }
```

```
protected override void OnPaint( PaintEventArgs e )
{ Graphics g = e.Graphics;
  Rectangle r = ClientRectangle;
  dx = r.Width / (xSize+2);
  dy = r.Height / ySize;
  for ( y=0; y < ySize; y++ )
  { button[y].Top = y*dy+1;
    button[y].Left = xSize*dx+1;
    button[y].Width = r.Width - button[y].Left - 2;
    button[y].Height = dy-2;
  }
  Brush redbrush = new SolidBrush( Color.Red );
  Pen   redpen   = new Pen( Color.Red, 5 );
  for ( y=0; y < ySize+1; y++ )
    for ( x=0; x < xSize+1; x++ )
    { try { g.FillRectangle( brush[i0[y,x]], x*dx, y*dy, dx, dy ); }
      catch {};
      try { if ( C2 && c2[y,x] > 0 )
         g.FillRectangle( redbrush, x*dx+dx/4, y*dy+dy/4, dx/2, dy/2 ); }
      catch {};
      try { if ( C1V && c1v[y,x] > 0 )
         g.DrawLine( redpen,  x*dx, y*dy+1, x*dx, (y+1)*dy-1 ); }
      catch {};
      try { if ( C1H && c1h[y,x] > 0 )
         g.DrawLine( redpen,  x*dx+1, y*dy, (x+1)*dx-1, y*dy ); }
      catch {};
      try { if ( C0 && c0[y,x] > 0 )
         g.FillRectangle( redbrush,  x*dx-5, y*dy-5, 11, 11 ); }
      catch {};
    }
}
```

```
  protected void do_it( object sender, System.EventArgs e )
  { switch( ((Button)sender).Name)
    { case "Homunculus"://**********************************
        i0 =  new Byte[,]{ {1,0,0,0,0,0,0,0,0,0,2},
                           {0,0,0,0,9,9,9,0,0,0,0},
                           {0,0,0,6,9,5,9,6,0,0,0},
                           {0,0,0,0,9,9,9,0,0,0,0},
                           {0,0,0,0,0,9,0,0,0,0,0},
                           {5,6,9,9,9,9,9,9,9,6,5},
                           {0,0,0,0,9,9,9,0,0,0,0},
                           {0,0,0,0,9,9,9,0,0,0,0},
                           {0,0,0,0,9,0,9,0,0,0,0},
                           {0,0,0,0,9,0,9,0,0,0,0},
                           {0,0,0,0,9,0,9,0,0,0,0},
                           {3,0,0,0,3,0,3,0,0,0,4} };
        C0 = C1V = C1H = C2 = false;
        break;
      case "Threshold"://*************************************
        if ( ++threshold > 9 ) threshold = 1;
        button[1].Text = "Threshold=" + threshold.ToString();
        break;
      case "Noise"://*****************************************
        Random random = new Random();
        for ( y=0; y < ySize; y++ )
          for ( x=0; x < xSize; x++ )
          { Int32 noise =  random.Next() % 3 - 1;//gives -1 or 0 or +1
            noise += i0[y,x];//add former gray value
            if (noise < 0)      i0[y,x] = 0;
            else if (noise > 9) i0[y,x] = 9;
            else                i0[y,x] = (Byte)noise;
          }
         break;
      case "Clear":///****************************************
        for ( y=0; y < ySize; y++ )
          for ( x=0; x < xSize; x++ ) i0[y,x] = 0;
        threshold = 1; button[1].Text = "Threshold=1";
        break;
      case "C0-Cells"     : C0  = true; C1V = C1H = C2  = false; break;
      case "C1v-Cells"    : C1V = true; C0  = C1H = C2  = false; break;
      case "C1h-Cells"    : C1H = true; C0  = C1V = C2  = false; break;
      case "C2-Cells"     : C2  = true; C0  = C1V = C1H = false; break;
      case "C1v+C1h-Cells": C1H = C1V = true; C0 = C2   = false; break;
      case "All-Cells"    : C0  = C1V = C1H = C2 = true;         break;
    }
    for ( y=0; y <= ySize; y++ ) //clear cells
      for ( x=0; x <= xSize; x++ )
      { try { c2 [y,x] = 0; } catch {}
        try { c1h[y,x] = 0; } catch {}
        try { c1v[y,x] = 0; } catch {}
              c0 [y,x] = 0;
      }
    Byte above, below, left, right;
    for ( y=0; y < ySize; y++ )
      for ( x=0; x < xSize; x++ )
      { if ( i0[y,x] >= threshold ) c2[y,x] = 1; else continue;
        try { above = i0[y-1,x  ]; } catch { above = 0; }
        try { below = i0[y+1,x  ]; } catch { below = 0; }
        try { left  = i0[y  ,x-1]; } catch { left  = 0; }
        try { right = i0[y  ,x+1]; } catch { right = 0; }
        if (above < threshold) {c1h[y  ,x  ]=c0[y  ,x  ]=c0[y  ,x+1]=1;}
        if (below < threshold) {c1h[y+1,x  ]=c0[y+1,x  ]=c0[y+1,x+1]=1;}
        if (left  < threshold) {c1v[y  ,x  ]=c0[y  ,x  ]=c0[y+1,x  ]=1;}
        if (right < threshold) {c1v[y  ,x+1]=c0[y  ,x+1]=c0[y+1,x+1]=1;}
      }
    Invalidate();
  }
}
```