

Course IPCis: Image Processing with C#

Chapter C3: The Filter Project

Copyright © by V. Miszalok, last update: 10-01-2008

- ↓ [Projekt filter1 mit leerem Fenster](#)
- ↓ [Bild lesen und anzeigen, Vorbereitungen für 6 Bilder](#)
- ↓ [Code für Rauschen](#)
- ↓ [Code für Tiefpass](#)
- ↓ [Code für die Hochpässe](#)
- ↓ [Experimente](#)
- ↓ [Beispielbilder](#)
- ↓ [Weitere Aufgaben](#)

Projekt filter1 mit leerem Fenster

Anleitung für **Visual Studio 2008**:

1) Main Menu nach dem Start von VS 2008: File -> New Project... ->
 Visual Studio installed templates: Windows Forms Application
 Name: filter1 -> Location: C:\temp -> Create directory for solution: ausschalten ->
 OK

Es meldet sich Form1.cs[Design].

2) Sie müssen zwei überflüssige Files löschen: Form1.Designer.cs und Program.cs.
 Sie erreichen diese Files über das Solution Explorer - filter1-Window:
 Klicken Sie das Pluszeichen vor filter1 und dann das Pluszeichen vor Form1.cs.
 Klicken Sie mit der **rechten** Maustaste auf den Ast Program.cs. Es öffnet sich ein Kontextmenu.
 Sie Klicken auf Delete. Eine Message Box erscheint: 'Program.cs' will be deleted permanently.
 Sie quittieren mit OK.
 Klicken Sie mit der **rechten** Maustaste auf den Ast Form1.Designer.cs und löschen auch dieses File.

3) Klicken Sie mit der **rechten** Maustaste auf das graue Fenster Form1.
 Es öffnet sich ein kleines Kontextmenü. Klicken Sie auf View Code.
 Sie sehen jetzt den vorprogrammierten Code von Form1.cs. Löschen Sie den gesamten Code vollständig.

4) Schreiben Sie in das vollständig leere File Form1.cs folgende 3 Zeilen:

```
public class Form1 : System.Windows.Forms.Form
{ static void Main() { System.Windows.Forms.Application.Run( new Form1() ); }
}
```

5) Klicken Sie Debug -> Start Without Debugging Ctrl F5.
 Beenden Sie filter1.exe.

Bild lesen und anzeigen, Vorbereitungen für 6 Bilder

Sie löschen alles und schreiben in das leere Codefenster Form1.cs folgenden Code:

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

public class Form1 : Form
{ static void Main() { Application.Run( new Form1() ); }
  Brush bbrush = SystemBrushes.ControlText;
  Bitmap Original, Noise, Lowpass, HighVertical, HighHorizontal, HighGradient;
  Byte[,] grayarray;
  Int32 x, y, xSize, ySize, gray;
```

```

public Form1()
{
    MenuItem miRead = new MenuItem( "&Read", new EventHandler( MenuFileRead ) );
    MenuItem miExit = new MenuItem( "&Exit", new EventHandler( MenuFileExit ) );
    MenuItem miFile = new MenuItem( "&File", new MenuItem[] { miRead, miExit } );
    Menu = new System.Windows.Forms.MainMenu( new MenuItem[] { miFile } );
    Text = "Filter1";
    SetStyle( ControlStyles.ResizeRedraw, true );
    Width = 1024;
    Height = 800;
}

void MenuFileRead( object obj, EventArgs ea )
{
    OpenFileDialog dlg = new OpenFileDialog();
    dlg.Filter = "bmp files (*.bmp)|*.bmp|All files (*.*)|*.*" ;
    if ( dlg.ShowDialog() != DialogResult.OK ) return;
    Original = (Bitmap)Image.FromFile( dlg.FileName );
    if ( Original == null ) return;
    Cursor.Current = Cursors.WaitCursor;
    xSize = Original.Width;
    ySize = Original.Height;
    Noise = new Bitmap( xSize, ySize, PixelFormat.Format16bppRgb555 );
    Lowpass = new Bitmap( xSize, ySize, PixelFormat.Format24bppRgb );
    HighVertical = new Bitmap( xSize, ySize, PixelFormat.Format16bppRgb555 );
    HighHorizontal = new Bitmap( xSize, ySize, PixelFormat.Format16bppRgb555 );
    HighGradient = new Bitmap( xSize, ySize, PixelFormat.Format16bppRgb555 );
    grayarray = new Byte [xSize, ySize];
    for ( y=0; y < ySize; y++ )
        for ( x=0; x < xSize; x++ )
            {
                Color color = Original.GetPixel( x, y );
                gray = ( color.R + color.G + color.B ) / 3;
                grayarray[x, y] = (Byte)gray;
            }
    Cursor.Current = Cursors.Arrow;
    Invalidate();
}

void MenuFileExit( object obj, EventArgs ea )
{
    Application.Exit();
}

protected override void OnPaint( PaintEventArgs e )
{
    Graphics g = e.Graphics;
    if ( Original == null ) { g.DrawString( "Open an Image File !", Font, bbrush, 0, 0 );
return; }
    Rectangle cr = ClientRectangle;
    g.DrawImage( Original , 0 , 0 , cr.Width/3, cr.Height/2 );
    g.DrawImage( Noise , cr.Width/3 , 0 , cr.Width/3, cr.Height/2 );
    g.DrawImage( Lowpass , (2*cr.Width)/3, 0 , cr.Width/3, cr.Height/2 );
    g.DrawImage( HighHorizontal, 0 , cr.Height/2, cr.Width/3, cr.Height/2 );
    g.DrawImage( HighVertical , cr.Width/3 , cr.Height/2, cr.Width/3, cr.Height/2 );
    g.DrawImage( HighGradient , (2*cr.Width)/3, cr.Height/2, cr.Width/3, cr.Height/2 );
}
}

```

Klicken Sie Debug -> Start Without Debugging Ctrl F5. Erproben Sie das Programm. Lesen Sie Bilder der Formate BMP, ICO, GIF, JPG, PNG, TIFF.

Code für Rauschen

Version2: Beenden Sie Ihr Programm filter1.

Schreiben Sie in der Funktion void MenuFileRead(object obj, EventArgs ea) unterhalb der Klammer der inneren for-Schleife, aber noch vor Cursor.Current = Cursors.Arrow::

```

// Version 2 Noise
Int32 amplitude = 256;
Int32 half_amplitude = amplitude / 2;
Random random = new Random();
for ( y=0; y < ySize; y++ )
    for ( x=0; x < xSize; x++ )
        {
            gray = grayarray[x, y];
            gray += random.Next(amplitude) - half_amplitude;
            if (gray < 0) gray = 0; else if (gray > 255) gray = 255;
            Noise.SetPixel( x, y, Color.FromArgb( gray, gray, gray ) );
        }
}

```

Klicken Sie Debug -> Start Without Debugging Ctrl F5. Erproben Sie das Verrauschen des Bildes.

Code für Tiefpass

Version3: Beenden Sie Ihr Programm filter1.

Schreiben Sie nach der Funktion void MenuFileRead(object obj, EventArgs ea) unterhalb der Klammer der inneren for-Schleife von Version 2 Noise, aber noch vor Cursor.Current = Cursors.Arrow;:

```
// Version 3: Lowpass without border handling
Int32 LowpassSize = 11; //insert an odd size: 3, 5, 7, ..., 29.
Int32 MidWeight = 1; //set a positive mid weight: 1, 2, 3, ..., 1000.
Int32 xx, yy, sum, d = LowpassSize/2;
float divisor = (2*d+1)*(2*d+1) + MidWeight - 1; //MidWeight == 1 means no weight
for ( y=d; y < ySize-d; y++ )
    for ( x=d; x < xSize-d; x++ )
        { sum = (MidWeight-1) * grayarray[x,y]; //extra mid weight
          for ( yy=-d; yy <= d; yy++ )
              for ( xx=-d; xx <= d; xx++ )
                  sum += grayarray[x+xx,y+yy];
          gray = Convert.ToByte( (float)sum / divisor );
          Lowpass.SetPixel( x, y, Color.FromArgb( gray, gray, gray ) );
        }
```

Klicken Sie Debug -> Start Without Debugging Ctrl F5. Erproben Sie die Verwaschung des Bildes.

Code für die Hochpässe

Version3: Beenden Sie Ihr Programm filter1.

Schreiben Sie in die Funktion void MenuFileRead(object obj, EventArgs ea) unterhalb der Klammer der inneren for-Schleife von Version 3 Lowpass, aber noch vor Cursor.Current = Cursors.Arrow;:

```
// Version 4 HighHorizontal + HighVertical + Highgradient
Int32 sumleft, sumright, sumtop, sumbottom, diff;
for ( y=1; y < ySize-1; y++ )
    for ( x=1; x < xSize-1; x++ )
        { sumleft = grayarray[x-1,y] + grayarray[x-1,y-1] + grayarray[x-1,y+1];
          sumright = grayarray[x+1,y] + grayarray[x+1,y-1] + grayarray[x+1,y+1];
          sumtop = grayarray[x,y-1] + grayarray[x-1,y-1] + grayarray[x+1,y-1];
          sumbottom = grayarray[x,y+1] + grayarray[x-1,y+1] + grayarray[x+1,y+1];
          diff = Math.Abs( sumleft - sumright );
          if ( diff > 255 ) HighHorizontal.SetPixel( x,y,Color.FromArgb( 255, 255, 255 ) );
          else HighHorizontal.SetPixel( x,y,Color.FromArgb( diff,diff,diff ) );
          diff = sumtop - sumbottom; if ( diff < 0 ) diff *= -1;
          if ( diff > 255 ) HighVertical.SetPixel( x,y,Color.FromArgb( 255, 255, 255 ) );
          else HighVertical.SetPixel( x,y,Color.FromArgb( diff, diff, diff ) );
          Double diff_h = sumleft - sumright;
          Double diff_v = sumtop - sumbottom;
          diff = Convert.ToInt32( Math.Sqrt( diff_h * diff_h + diff_v * diff_v ) );
          if ( diff > 255 ) HighGradient.SetPixel( x,y,Color.FromArgb( 255, 255, 255 ) );
          else HighGradient.SetPixel( x,y,Color.FromArgb( diff, diff, diff ) );
        }
```

Klicken Sie Debug -> Start Without Debugging Ctrl F5. Erproben Sie die Hochpassfilter.

Experimente

- (1) Variieren Sie die amplitude zwischen 10 und 1000.
- (2) Variieren Sie LowpassSize zwischen 3 und 29. Vorsicht mit höheren Werten: Die Rechenzeit steigt quadratisch !
- (3) Variieren Sie MidWeight zwischen 1 und 1000.
- (4) Nehmen Sie das Noise-Bild als Eingang für den Lowpass (neuer grayarray2 notwendig).
- (5) Nehmen Sie das Lowpassbild als Eingang für den Highpass (noch ein neuer grayarray3 notwendig).
- (6) Vereinfachen Sie HighGradient zu einem HighMaximum = Math.Max(HighpassVertical, HighpassHorizont).

Beispielbilder

Im Prinzip sollte das Programm alle Bildformate BMP, ICO, GIF, JPG, PNG, TIFF lesen und anzeigen. Falls Sie eine alte Graphikkarte mit 8 Bit benutzen und/oder falls Sie Ihren Desktop auf 256 Farben eingestellt haben, kann es sein, dass die Farben ganz schlecht aussehen. Falls Sie keine *.bmp - Dateien auf Ihrer Harddisk finden, benutzen Sie folgende Beispielbilder:

Download: [Butterfly.bmp 217 kB 24Bit-TrueColor-Bild](#)

Download: [Madonna.bmp 18 kB 8Bit-Grauwert-Bild](#)

Download: [Lena256.bmp 66 kB 8Bit-Grauwert-Bild](#)

Download: [Lena512.bmp 258 kB 8Bit-Grauwert-Bild](#)

Download: [Angiography.bmp 66 kB 8Bit-Grauwert-Bild](#)

Weitere Aufgaben

Klicken Sie auf `Help` in der Menüleiste von Visual Studio. Klicken Sie auf das Untermenü `Index`.

Gehen Sie in das Feld `Filtered by:` und wählen Sie dort `.NET Framework`. Dann geben Sie im Feld `Look for:` folgende Schlüsselwörter ein und lesen Sie die Texte:

`PixelFormat enumeration`

`Cursor class, all members`

`Math class, all members`

Erhöhen sie die Anzahl der Bilder durch anzeigen mehrerer Rauschbilder mit verschiedener `amplitude` und/oder mehrerer Tiefpassbilder mit diversen `LowpassSize` und `MidWeight`. Passen Sie Positionen und Höhen der Bilder an die Anzahl an.

Erproben Sie andere Mauszeiger als `Cursors.WaitCursor`.

Erproben Sie andere Hochpassfilter mit schräg über Eck stehenden (+45 Grad -45 Grad) Nachbarn.

Beenden Sie Visual Studio, starten sie den Explorer, löschen Sie die gesamte Directory `C:\temp\filter1`

Starten Sie Visual Studio wieder und erzeugen dasselbe Programm so oft, bis Sie das Programm ohne Anleitung und schnell von Null an erstellen, verändern und bedienen können.

Erfinden und erproben Sie neue Varianten des Programms in Form von neuen Projekten `filter2`, `filter3` usw. nach obigem Muster.