Course IPC7, C3: Filter, Bauanleitung mit C++/MFC7.0

Projekt filter1 mit leerem Fenster

`Microsoft Visual Studio.NET starten`
`File – New – Project – Project Types: Visual C++ Projects, Templates: MFC Application`
Name: `filter1`
Location: `C:\temp`
Button `OK` unten Mitte klicken
Links unter `Overview` auf `Application Type` klicken.
Schritt1: **`Single document`** einschalten
Schritt 2: **`Document/View architecture support`** Checkbox **einschalten**
Schritt 3: `Finish`

Präprozessorbefehle und Deklarationen in filter1Doc.h

Klicken Sie im Hauptmenu von VS.NET auf den Menupunkt `View` und dann auf den Unterpunkt `Class View Ctrl+Shift+C`.
Sie sehen im rechten Bereich des VS.NET-Hauptfensters das Unterfenster `Class View - filter1`.
An dessen unteren Rand die Registerkarte Klassen klicken.
Darin sehen Sie die Zeile `+ filter1` , klicken Sie das Pluszeichen.
Sie sehen u.a. `+ Cfilter1Doc`, doppelklicken Sie diesen Klassennamen.
Sie editieren jetzt das File `filter1Doc.h`, Schreiben Sie dort vor die Klasse `class Cfilter1Doc : public CDocument` am besten unter die Zeile `#pragma once` die Präprozessoranweisungen:

```
#include < vector >
#include < math.h >
```

Schreiben Sie in die Klasse direkt unter die sich öffnende geschweifte Klammer die Deklarationen:

```
public:
  BITMAPFILEHEADER    FH;
  BYTE                IBytes[1200];//bytes for BitmapInfoHeader+Palette
  BITMAPINFOHEADER*   pIH;         //pointer on BitmapInfoHeader
  BITMAPINFO*         pI;          //pointer on BitmapInfo
  std::vector< BYTE > Original;
  std::vector< BYTE > Noise;
  std::vector< BYTE > Lowpass;
  std::vector< BYTE > HighVertical;
  std::vector< BYTE > HighHorizont;
  std::vector< BYTE > HighGradient;
```

Doppelklicken Sie auf den Konstruktor der Klasse: `Cfilter1Doc()`. Sie müssen dort drei Variable initialisieren:

```
Cfilter1Doc::Cfilter1Doc()
{  memset( IBytes, 0, sizeof(IBytes) );
   pIH = (BITMAPINFOHEADER*) IBytes;
   pI  = (BITMAPINFO*)       IBytes;
}
```

Prüfen, ob alles soweit in Ordnung ist: `Debug -> Start Without Debugging`. Beenden.

Code für Lesen und Noise in Serialize in Cfilter1Doc

Doppelklicken Sie auf `Serialize(CArchive& ar)`. Ersetzen Sie die fast leere Funktionshülse durch folgenden Code:

```
void Cfilter1Doc::Serialize(CArchive& ar)
{ if (ar.IsStoring()) {}
  else
  {  ar.Read( & FH, sizeof(BITMAPFILEHEADER) );
     if ( FH.bfType   != 'MB') { forget_it(); return; }
     if ( FH.bfSize    <= 54 ) { forget_it(); return; }
     if ( FH.bfOffBits <  54 ) { forget_it(); return; }

     int nBytesInfo  = FH.bfOffBits - sizeof(BITMAPFILEHEADER);
     int nBytesPixel = FH.bfSize    - FH.bfOffBits;

     ar.Read( IBytes, nBytesInfo ); //BitmapInfoHeader+Palette

     if ( !(pIH->biBitCount == 8 ) ) { forget_it(); return; }

     Original    .resize( nBytesPixel );
     Noise       .resize( nBytesPixel );
     Lowpass     .resize( nBytesPixel );
     HighVertical.resize( nBytesPixel );
     HighHorizont.resize( nBytesPixel );
     HighGradient.resize( nBytesPixel );

     std::vector< BYTE >::iterator p, p1, p2, p3, p4;
     ar.Read( &Original.front(), nBytesPixel );

#define AMPLITUDE 128
     for ( p1=Original.begin(), p2=Noise.begin();
           p1 < Original.end();
           p1++, p2++ )
     {  int noise = *p1 + AMPLITUDE/2 - rand()%AMPLITUDE;
        if ( noise <   0 )      *p2 = 0;
        else if ( noise > 255 ) *p2 = 255;
        else                    *p2 = BYTE(noise);
     }
   } //end of else IsStoring
}
```

Klicken Sie im `Class View - filter1` - Fenster mit der rechten Maustaste auf die Klasse `Cfilter1Doc`. Es öffnet sich ein Kontextmenü. Stellen Sie die Maus auf den 4 Menüpunkt `Add` und klicken Sie dann auf `Add Function...` . Es erscheint der "`Add Member Function Wizard - filter1`". Tragen Sie in seine Dialogfelder ein:

```
Return type : void
Function name: forget_it
Access : private
.cpp file : filter1doc.cpp
```

Alle weiteren Felder und Checkboxen bleiben leer. Sie verlassen den Wizard mit dem Button `Finish`.

Der Wizard erzeugt Ihnen einen Funktionshülle, die Sie folgendermaßen füllen:

```
void Cfilter1Doc::forget_it()
{ memset( IBytes, 0, sizeof(IBytes) );
  for ( int i=0; i < 10; i++ ) MessageBeep(-1);
}
```

Ausführen, lesen muss funktionieren, aber man sieht noch nichts. Falls man eine Datei öffnet, die kein oder ein falsches Bitmap enthält, muss es piepsen.

Code für OnDraw in Cfilter1View

Nun programmieren Sie in `filter1View.cpp` die bereits vorhandene Funktion: `void Cfilter1View::OnDraw(CDC* /*pDC*/)` bis sie so aussieht:

```
void Cfilter1View::OnDraw(CDC* pDC)
{  Cfilter1Doc* pDoc = GetDocument();
   if ( !pDoc->pIH->biSize ) { pDC->TextOut(0,0,"Open an 8-Bit *.BMP file !");
return; }

   CRect R; GetClientRect( R ); //welche Größe hat die ClientArea?
   SetStretchBltMode( pDC->GetSafeHdc(), COLORONCOLOR );
   StretchDIBits( pDC->GetSafeHdc(),
                  0, 0, R.Width()/3, R.Height()/2,
                  0, 0, pDoc->pIH->biWidth, pDoc->pIH->biHeight,
                  &(pDoc->Original.front()), pDoc->pI,
                  DIB_RGB_COLORS, SRCCOPY );
   StretchDIBits( pDC->GetSafeHdc(),
                  R.Width()/3, 0, R.Width()/3, R.Height()/2,
                  0, 0, pDoc->pIH->biWidth, pDoc->pIH->biHeight,
                  &(pDoc->Noise.front()), pDoc->pI,
                  DIB_RGB_COLORS, SRCCOPY );
   StretchDIBits( pDC->GetSafeHdc(),
                  2*R.Width()/3, 0, R.Width()/3, R.Height()/2,
                  0, 0, pDoc->pIH->biWidth, pDoc->pIH->biHeight,
                  &(pDoc->Lowpass.front()), pDoc->pI,
                  DIB_RGB_COLORS, SRCCOPY );
   StretchDIBits( pDC->GetSafeHdc(),
                  0, R.Height()/2, R.Width()/3, R.Height()/2,
                  0, 0, pDoc->pIH->biWidth, pDoc->pIH->biHeight,
                  &(pDoc->HighVertical.front()), pDoc->pI,
                  DIB_RGB_COLORS, SRCCOPY );
   StretchDIBits( pDC->GetSafeHdc(),
                  R.Width()/3, R.Height()/2, R.Width()/3, R.Height()/2,
                  0, 0, pDoc->pIH->biWidth, pDoc->pIH->biHeight,
                  &(pDoc->HighHorizont.front()), pDoc->pI,
                  DIB_RGB_COLORS, SRCCOPY );
   StretchDIBits( pDC->GetSafeHdc(),
                  2*R.Width()/3, R.Height()/2, R.Width()/3, R.Height()/2,
                  0, 0, pDoc->pIH->biWidth, pDoc->pIH->biHeight,
                  &(pDoc->HighGradient.front()), pDoc->pI,
                  DIB_RGB_COLORS, SRCCOPY );
}
```

`filter1` ist vorläufig fertig, ausführen, `*.bmp`-File öffnen, , 8-Bit *.bmp-File öffnen, Sie sehen 6 Felder, davon 2 mit Bildern gefüllt, beenden.

Code für Lowpass in Serialize in Cfilter1Doc

Schreiben Sie folgenden Code unterhalb des in `Serialize` schon vorhandenen Codes, nach der geschweiften Klammer, die die for-Schleife von Noise abschließt, aber noch vor die abschließende geschweifte Klammer von `//end of else IsStoring` in `Serialize`:

```
#define LOWPASSSIZE 11
#define ADDMIDWEIGHT 0
   memset( &Lowpass.front(), 0, nBytesPixel );
   int xSize = pIH->biWidth;
   int ySize = pIH->biHeight;
   int x, y, xx, yy, sum, d = LOWPASSSIZE/2;
   int norm = (2*d+1)*(2*d+1)+ADDMIDWEIGHT;
   for ( y = d; y < ySize - d; y++ )
   {  p1 = Original.begin() + y * xSize;
      p  = Lowpass .begin() + y * xSize;
      for ( x = d; x < xSize - d; x++ )
      {  p2 = p1 + x;
         sum = ADDMIDWEIGHT * *p2;
         for ( yy = -d; yy <= d; yy++ )
         {  p3 = p2 + yy * xSize;
            for ( xx = -d; xx < d; xx++ )
            {  p4 = p3 + xx;
               sum += *p4;
            }
         }
         *(p+x) = (BYTE)( sum / norm );
      }
   }
```

Die 2. Version von `filter1` ist vorläufig fertig, ausfuehren, 8-Bit *.bmp-File öffnen, Sie sehen 6 Felder, davon 3 mit Bildern gefüllt, beenden.
Es vergeht viel Zeit vom dem Laden der Bilder bis man etwas sieht. Das Berechnen aller Filter geht viel schneller und synchron zur Mausbewegung, wenn Sie das Projekt ohne Debugger im `release` - mode übersetzen. Gehen Sie dazu im Hauptmenü von VS auf `Build -> Configuration Manager`. Es öffnet sich ein Fenster `Configuration Manager`. Sie setzen das Feld `Active Solution Configuration` auf "`Release`" und die Spalte `Configuration` auch auf "`Release`". Verlassen mit `Close`. Dann alles neu übersetzen, linken und ausführen mit `Debug -> Start Without Debugging`.
Erpoben, beenden, löschen, neu programmieren.

Code für Highpass in Serialize in Cfilter1Doc

Schreiben Sie folgenden Code unterhalb des in `Serialize` schon vorhandenen Codes, nach der letzten geschweiften Klammer, die Lowpass abschließt, aber noch vor die abschließende geschweifte Klammer von `//end of else IsStoring in Serialize`:

```
   memset( &HighVertical.front(), 0, nBytesPixel );
   memset( &HighHorizont.front(), 0, nBytesPixel );
   memset( &HighGradient.front(), 0, nBytesPixel );
   for ( y = 1; y < ySize - 1; y++ )
   {  p1 = Original.begin() + y * xSize;
      p2 = HighVertical.begin() + y * xSize;
      p3 = HighHorizont.begin() + y * xSize;
      p4 = HighGradient.begin() + y * xSize;
      for ( x = 1; x < xSize - 1; x++ )
      {  p = p1 + x;
         int sumleft   = *(p-1) + *(p-1-xSize) + *(p-1+xSize);
         int sumright  = *(p+1) + *(p+1-xSize) + *(p+1+xSize);
         int sumtop    = *(p-xSize) + *(p-xSize-1) + *(p-xSize+1);
         int sumbottom = *(p+xSize) + *(p+xSize-1) + *(p+xSize+1);
         int diff = sumleft - sumright;
         if ( diff < 0 ) diff *= -1;
         if ( diff > 255 ) *(p2+x) = 255;
         else              *(p2+x) = BYTE(diff);
         diff = sumtop - sumbottom;
         if ( diff < 0 ) diff *= -1;
         if ( diff > 255 ) *(p3+x) = 255;
         else              *(p3+x) = BYTE(diff);
         double diff_v = sumleft - sumright;
         double diff_h = sumtop - sumbottom;
         diff = int( 0.5 + _hypot( diff_v, diff_h ) );
         if ( diff > 255 ) *(p4+x) = 255;
         else              *(p4+x) = BYTE(diff);
      }
   }
```

Die 3. Version von `filter1` ist fertig, ausführen, 8-Bit *.bmp-File öffnen. Sie sehen 6 Bilder im Fenster, beenden.

Experimente

(1) Variieren Sie die `#define AMPLITUDE` zwischen 10 und 1000.
(2) Variieren Sie `#define LOWPASSSIZE` zwischen 3 und 100.
(3) Variieren Sie `#define ADDMIDWEIGHT` zwischen 0 und 1000.
(4) Nehmen Sie das Noise-Bild als Eingang für den Lowpass.
(5) Nehmen Sie das Lowpassbild als Eingang für den Highpass.
(6) Vereinfachen Sie HighpassGradient zu einem `HighpassMaximum = max( HighpassVertical, HighpassHorizont)`.
(7) Ersetzen Sie die 4 Pointer `p1 - p4` im Lowpass durch einen einzigen. Das macht den Filter zwar langsamer aber übersichtlicher.

Beispielbilder

Wenn Sie keine 8-Bit *.bmp - Dateien auf Ihrer Harddisk finden, benutzen Sie folgende Beispielbilder:

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