

HÖHERE TECHNISCHE BUNDESLEHRANSTALT

HOLLABRUNN

Höhere Abteilung für Elektronik – Technische Informatik

Klasse/ Jahrgang: 5BHEL	Gruppe: 1 / b	Übungsleiter: Prof. Wihsböck
Übungsnummer: II/6	Übungstitel: Bildverarbeitung	
Datum der Übung: 17.11.2015	Teilnehmer: Dorner, Mötz, Prinz, Oberhamberger	
Datum der Abgabe: 01.12.2015	Schritfführer: Oberhamberger	Unterschrift:

	Beurteilung
Aufgabenstellung	
Dokumentation	
Messschaltungen	
Messtabellen	
Berechnungen	
Programmlistings	
Auswertung	
Diagramme	
Berechnungen	
Simulationen	
Schlußfolgerungen	
Kommentare	
Inventarliste	
Messprotokoll	
Form	
Summe	

Allgemeiner Teil

Titel der Übung: Bildverarbeitung
Übungsnummer: I/2
Übungsplatz: PAD
Schriftführer: Oberhamberger
Übungsteilnehmer: Bernhard Prinz
Florian Mötz
Fabian Dorner

Inhaltsverzeichnis

1. Eclipse	3
2. ImageJ Plug-In	3
3. Punktoperationen	4
3.1 Bildinvertierung	4
3.2 Helligkeit um 50% erhöhen	5
3.3 Grenzwertoperation	6
4. Filteroperation	7
4.1 3x3 Medianfilter	7
5. Bildanalyse	8
5.1 Objekte zählen	8
5.2 Messen des Flächeninhalts	9
6. Erkenntnis	10
7. Mitschrift	11
8. Inventarliste	12

1. Eclipse

Als Programmieroberfläche wurde das Programm "Eclipse Luna" verwendet. Dieses Programm dient zur Softwareentwicklung verschiedenster Art. Ist aber Ursprünglich als integrierte Entwicklungsumgebung für die Programmiersprache Java genutzt worden. Desweiteren gibt es für Eclipse eine Vielzahl an quelloffenen Erweiterungen. In der Laborübung wurde das sogenannte ImageJ Plug-In verwendet.

2. ImageJ Plug-In

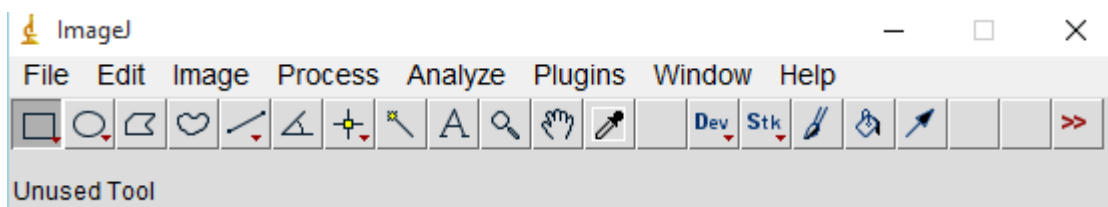
ImageJ ist ein Plug-In zur Bildverarbeitung indem man eigene Plug-Ins hinzufügen kann.

Zuerst musste ImageJ in Eclipse eingebunden werden. Das geht wie folgt:

File -> Import -> General -> Existing Project into Workspace -> ImageJPlugin auswählen -> Finish

Nachdem ImageJ eingebunden ist kann man auf das Run-Symbol in Eclipse drücken und ImageJ startet sich.

Danach öffnet sich dieses Fenster:



Um ein Bild einzufügen klickt man auf File -> Open und wählt dann ein Bild aus. Danach öffnet sich in einem separaten Fenster das Bild das nun bearbeitet werden kann.

Desweiteren ist zu beachten, wenn ein Bild bereits geöffnet ist und dann das Programm "ImageJ" geschlossen wird kann man das Bild nur mehr über den Task Manager schließen!

3. Punktoperationen

Definition: Der Farbwert eines Pixels wird nur von der Abhängigkeit des bisherigen Farbwertes des Pixels berechnet.

3.1 Bildinvertierung

Aufgabenstellung: Es musste ein Plug-In realisiert werden welches ein 8-bit Graustufenbild invertiert.

Programmcode:

```
public class Inverter_PointOperation implements PlugInFilter
{
    @Override
    public int setup(String arg, ImagePlus imp)
    {
        return DOES_8G;
    }

    @Override
    public void run(ImageProcessor ip)
    {
        int w = ip.getWidth();
        int h = ip.getHeight();

        for(int x=0;x<w;x++)
        {
            for(int y=0;y<h;y++)
            {
                int pixelValue = ip.getPixel(x,y);
                ip.putPixel(x,y, 255 - pixelValue);
            }
        }
    }
}
```

Ergebnis:

Vorher:



Nachher:



3.2 Helligkeit um 50% erhöhen

Aufgabenstellung: Es musste ein Plug-In realisiert werden in dem die Helligkeit eines 8-Bit Graustufenbild um 50% erhöhen soll.

Programmcode:

```
public class Brightness_50 implements PlugInFilter
{
    @Override
    public int setup(String arg, ImagePlus imp) {

        return DOES_8G;
    }
    @Override
    public void run(ImageProcessor ip) {

        int w = ip.getWidth();
        int h = ip.getHeight();

        for(int x=0;x<w;x++)
        {
            for(int y=0;y<h;y++)
            {
                int pixelValue = ip.getPixel(x,y);
                double newpixelvalue = pixelValue * 1.5;
                ip.putPixel(x,y, (int)newpixelvalue);
            }
        }
    }
}
```

Ergebnis:

Vorher:



Nachher:



3.3 Grenzwertoperation

Aufgabenstellung:

Es musste ein Plug-In realisiert werden welches ein 8-Bit Graustufenbild in ein Schwarz-Weiß Bild umwandelt indem der Grenzwert die Hälfte der 2^8 Bit genommen wird und alles das unter dem Wert ist, ist Schwarz und alles über diesem Wert ist Weiß .

Programmcode:

```
public class Grenzwert_128 implements PlugInFilter {

    @Override
    public int setup(String arg, ImagePlus imp)
    {
        return DOES_8G;
    }

    @Override
    public void run(ImageProcessor ip) {
        int w = ip.getWidth();
        int h = ip.getHeight();

        for(int x=0;x<w;x++)
        {
            for(int y=0;y<h;y++)
            {
                int pixelValue = ip.getPixel(x,y);
                if (pixelValue <= 128)
                {
                    pixelValue = 255;
                    ip.putPixel(x,y, pixelValue);
                }
                else
                {
                    pixelValue = 0;
                    ip.putPixel(x,y, pixelValue);
                }
            }
        }
    }
}
```

Ergebnis:

Vorher:



Nachher:



4. Filteroperation

Definition: Der neue Farbwert eines Pixels wird oft von den Farbwerten der umliegend Pixels berechnet.

4.1 3x3 Medianfilter

Aufgabenstellung: Es musste ein 3x3 Median-Filter realisiert werden mit dem man ein 8-bit Graustufenbild mit einem implementierten Salz und Pfeffer Rauschen filtern kann.

Programmcode:

```
public class Filter_Median implements PlugInFilter {
    @Override
    public int setup(String arg, ImagePlus imp) {
        return DOES_8G;
    }
    @Override
    public void run(ImageProcessor ip) {
        int width = ip.getWidth();
        int height = ip.getHeight();

        int[] filterArray = new int[9];

        ImageProcessor copyImage = ip.duplicate();

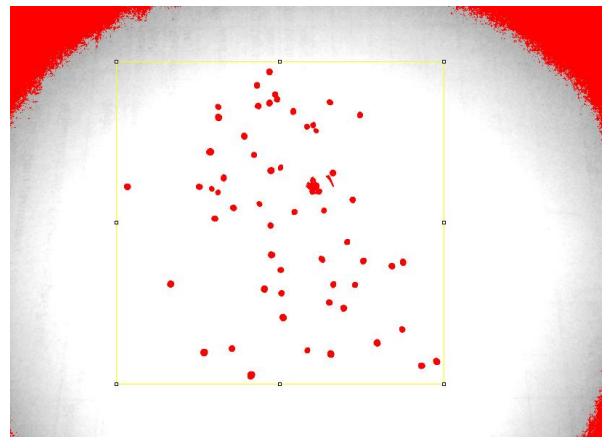
        for(int x = 0 ; x < width ; x++)
        {
            for(int y = 0 ; y < height ; y++)
            {
                for(int innerX = 0 ; innerX < 3 ; innerX++)
                {
                    for(int innerY = 0 ; innerY < 3 ; innerY++)
                    {
                        if(x+innerX-1 > 0 && x+innerX-1 < width &&
                           y+innerY-1 > 0 && y+innerY-1 < height)
                        {
                            filterArray[(innerX*3)+innerY] = copyImage.get(x+innerX-y+innerY-1);
                        }
                        else
                        {
                            filterArray[(innerX*3)+innerY] = 0;
                        }
                    }
                }
                Arrays.sort(filterArray);
                ip.putPixel(x, y, filterArray[5]);
            }
        }
    }
}
```

Ergebnis:**Vorher:****Nachher:****5. Bildanalyse**

Mit ImageJ ist es nicht nur möglich Bilder zu bearbeiten, man kann auch aus Bildern gewisse Informationen raus lesen, wie zum Beispiel Objekte zählen oder den Flächeninhalt eines Objektes.

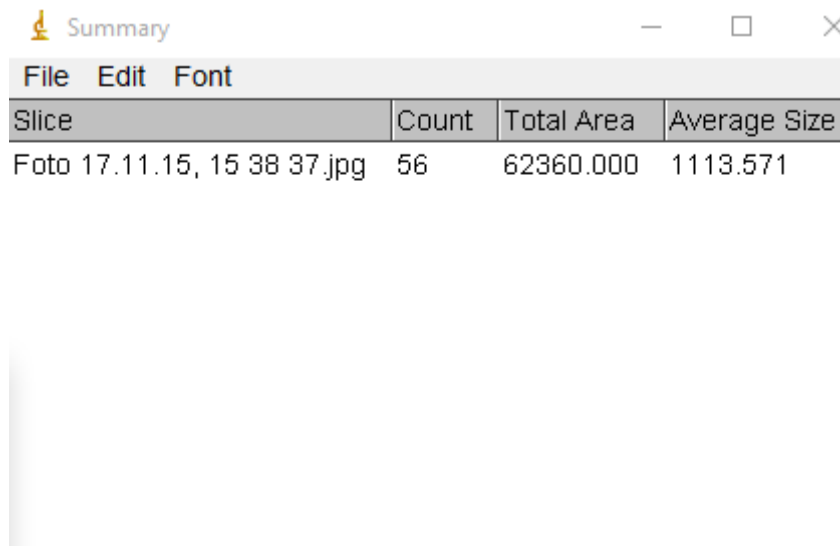
5.1 Objekte zählen**Aufgabenstellung:**

Es musste ein Foto gemacht werden und über ImageJ geöffnet werden. Danach wurde das Foto noch bearbeitet und dann wurden die Punkte im Foto mit dem Programm gezählt.

Foto:**Vorher:****Nach der Verarbeitung:**

Wie im Bild oben zu sehen ist muss das Foto so bearbeitet werden damit die Funktion feststellen kann was zu zählen ist. Danach wird noch ein Rechteck über die gewünschte Positionen ziehen. Alles was innerhalb diesem Rechteck ist wird gezählt

Zum Schluss wird ein Fenster angezeigt indem die Anzahl angezeigt wird.



5.2 Messen des Flächeninhalts

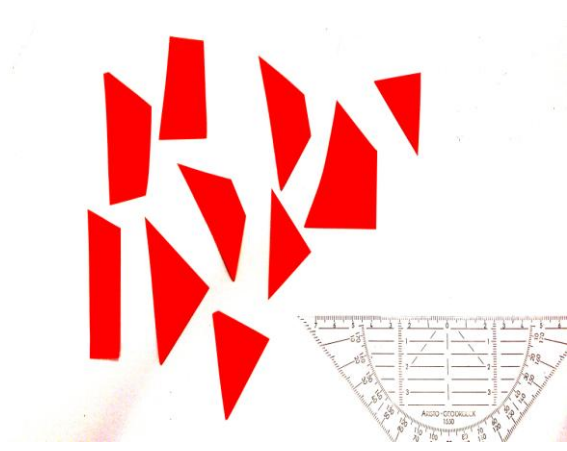
Aufgabenstellung: Es wurde wieder ein Foto gemacht nur das dieses Mal der Flächeninhalt gemessen werden. Hierbei musste das Foto vorher wieder bearbeitet werden und ein Maßstab gesetzt werden damit das Programm auf etwas beziehen kann.

Foto:

Vorher:

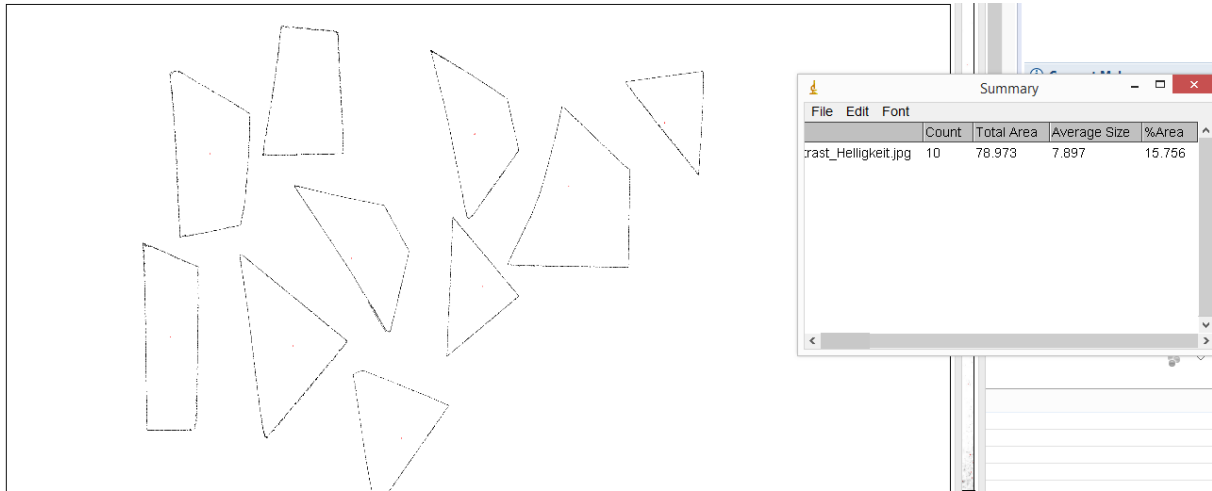


Nach der Verarbeitung:



Es musste zuerst eine Linie über dem Lineal gezogen werden damit das Programm eine Referenzlänge hat danach wurde die Funktion ausgeführt

Zum Schluss wird dann das Fenster angezeigt indem der Flächeninhalt steht.



Der Flächeninhalt beträgt 78,973 cm².

6. Erkenntnis

Es wurden die Grundlagen der objektorientierten Programmierung wiederholt.

Desweiteren wurde uns gezeigt was mit ImageJ alles machbar ist.

Zu guter Letzt wurden wir darauf hingewiesen das bei der Bildanalyse Fehler entstehen können.

7.Mitschrift

8. Inventarliste