**Seminar Bildverarbeitung**



Reto Rezzonico

Dozent: Georg Brügger

Hochschule für Technik Zürich

Frühjahrssemester 2009

Inhaltsverzeichnis

[1 Graustufen 4](#_Toc233278254)

[1.1 Theorie 4](#_Toc233278255)

[1.1.1 Umrechnung 4](#_Toc233278256)

[1.1.2 Farbkreis 4](#_Toc233278257)

[1.2 Umsetzung 5](#_Toc233278258)

[2 Bildbereich drehen 6](#_Toc233278259)

[2.1 Theorie 6](#_Toc233278260)

[2.1.1 Koordinaten 6](#_Toc233278261)

[2.1.2 Berechnung Drehung 7](#_Toc233278262)

[2.1.3 Aufgetretene Probleme 7](#_Toc233278263)

[2.2 Umsetzung 9](#_Toc233278264)

[3 Formerkennung 10](#_Toc233278265)

[3.1 Theorie 10](#_Toc233278266)

[3.1.1 Objekterkennung 10](#_Toc233278267)

[3.1.2 Objektdefinition 10](#_Toc233278268)

[3.1.3 Ermittlung der Eckpunkte 10](#_Toc233278269)

[3.1.4 Berechnung der Winkel in den Eckpunkten 12](#_Toc233278270)

[3.1.5 Vergleich von zwei Objekten 13](#_Toc233278271)

[3.1.6 Aufgetretene Probleme 14](#_Toc233278272)

[3.2 Umsetzung 15](#_Toc233278273)

[4 Erweiterungsmöglichkeiten 17](#_Toc233278274)

[4.1 Bessere Eckpunkterkennung 17](#_Toc233278275)

[4.2 Farbvergleich 17](#_Toc233278276)

[4.3 Mehrere Bildobjekte 17](#_Toc233278277)

Abbildungsverzeichnis

[Figure 1 - Farbkreis 4](#_Toc233278278)

[Figure 2 - Farbkreis Graustufen 4](#_Toc233278279)

[Figure 3 - Program Graustufen 5](#_Toc233278280)

[Figure 4 – Koordinaten 6](#_Toc233278281)

[Figure 5 - Unkorrigierte Drehung 8](#_Toc233278282)

[Figure 6 - Korrigierte Drehung 8](#_Toc233278283)

[Figure 7 - Program Drehung 9](#_Toc233278284)

[Figure 8 - Definition Objekt 10](#_Toc233278285)

[Figure 9 - Ermittlung Eckpunkte 11](#_Toc233278286)

[Figure 10 - Ermittlung Eckpunkte 11](#_Toc233278287)

[Figure 11 – Winkelmessung 12](#_Toc233278288)

[Figure 12 - Sortierung Eckpunkte 13](#_Toc233278289)

[Figure 13 - Vergleich Grösse 13](#_Toc233278290)

[Figure 14 - Vergleich Lage 14](#_Toc233278291)

[Figure 15 - Regelwerk Eckpunktermittlung 14](#_Toc233278292)

[Figure 16 - Program Formerkennung 15](#_Toc233278293)

[Figure 17 - Neue Form 16](#_Toc233278294)

[Figure 18 - Bekannte Form 16](#_Toc233278295)

[Figure 19 - Eckpunkterkennung 17](#_Toc233278296)

# Graustufen

## Theorie

### Umrechnung

Um das Bild in Graustufen umzuwandeln wird der Durchschnitt der RGB Farbwerte ermittelt. Die drei RGB Werte werden alle auf den ermittelten Durchschnittswert gesetzt.

**Beispiel:**

Ursprüngliche Farbe:

R 60; G 100; B 200

Durchschnittswert:

(60+100+200)/3 = 120

Grauwert:

R 120; G 120; B 120

### Farbkreis

Wird ein Farbkreis in ein Graustufenbild umgewandelt. Ist nur noch die Helligkeit der Farbe ersichtlich. Durch die Umwandlung in Graustufen wird die Farbe eliminiert, die Helligkeit bleibt erhalten.

|  |  |
| --- | --- |
| farbkreis.jpg  Figure 1 - Farbkreis | farbkreis_gray.jpg  Figure 2 - Farbkreis Graustufen |

## Umsetzung

In diesem Abschnitt wird die Handhabung des Programs für die Graustufenumwandlung erläutert.

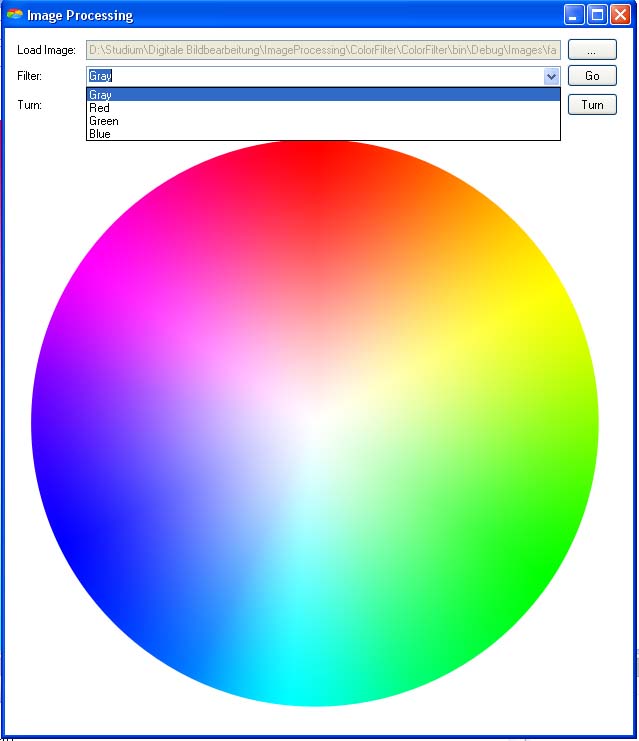


Figure 3 - Program Graustufen

**Schritt 1:**

Als erstes kann ein beliebiges Bild geladen werden. Dafür wird mit dem “…“ Button ein Dialog geöffnet mit dem das gewünscht Bild ausgewählt werden kann.

**Schritt 2:**

Auf das geladene Bild können nun verschiedene Farbfilter angewandt werden. In der Auswahl kann der gewünschte Farbfilter gewählt werden.

**Schritt 3:**

Mit einem Klick auf den “Go“ Button wird der Filter ausgeführt.

# Bildbereich drehen

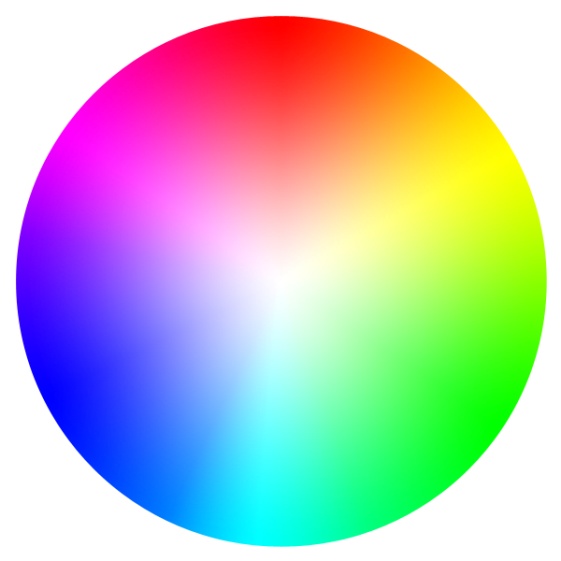
## Theorie

### Koordinaten

Ein Bild in .NET hat den Koordinatenpunkt 0/0 in der linken oberen Ecke. Die Koordinaten werden von diesem Punkt aus berechnet.

x

0/0



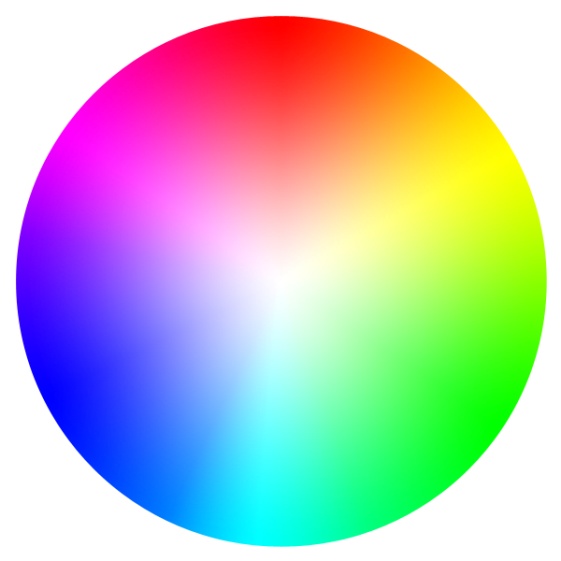
**100/150**

y

Figure 4 – Koordinaten

Um einen Bildausschnitt zu drehen wird ein neues Koordinatensystem über das Bild gelegt. Im neuen System ist der Drehpunkt der Koordinatenursprung 0/0.

**Beispiel mit dem Drehpunkt 100/150 (ursprüngliche Koordinaten):**



**-x**

**x**

**-y**

**y**

**0/0**

### Berechnung Drehung

Um den Bildbereich zu drehen wird für jeden Bildpunkt die neue Position ermittelt. Dafür wird der Ortsvektor des Pixels mit einer Drehmatrix multipliziert.

|  |  |
| --- | --- |
|  | [[1]](#footnote-1) |

Alpha ist ein beliebiger Drehwinkel, mit dem der Bildausschnitt gedreht werden soll. Für jeden Bildpunkt werden intern neben den beiden Koordinaten auch zwei Farbwerte gespeichert. Der ursprüngliche Farbwert muss bis zum Ende des Drehprozesses behalten werden, da der Farbwert bereits von einem anderen verarbeiteten Punkt überschrieben worden sein kann.

### Aufgetretene Probleme

#### Fehlende Punkte

Durch die Multiplikation mit der Drehmatrix entstehen bei den meisten Winkeln Koordinatenwerte mit Kommazahlen. Diese Koordinaten müssen auf eine gerade Koordinate gerundet werden. Gewisse Punkte im Zielbereich werden dadurch nicht erreicht und behalten die ursprüngliche Farbe.



Figure 5 - Unkorrigierte Drehung

**Lösung:**

Statt die Koordinaten streng mathematisch zu runden, werden X- und Y-Koordinate auf- sowie abgerundet. Damit werden statt nur einem, vier Zielkoordinaten ermittelt.

**Beispiel:**



Figure - Korrigierte Drehung

## Umsetzung

In diesem Abschnitt wird die Handhabung des Programs für die Drehung eines Bildausschnittes erläutert.

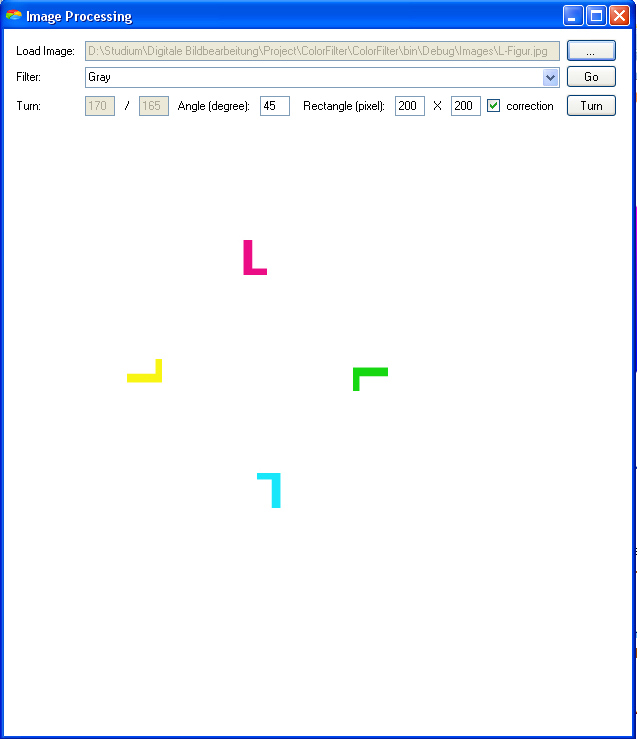


Figure 7 - Program Drehung

**Schritt 1:**

Mit der Maus kann der gewünschte Drehpunkt im Bild gewählt werden. Die gewählte Drehkoordinate wird im oberen Bereich angezeigt.

**Schritt 2:**

Der Drehwinkel und der zu drehende Bildausschnitt können in den Textfeldern eingegeben werden. Weiter kann gewählt werden ob die Drehung korrigiert (-> 2.1.3.1 Fehlende Punkte) ausgeführt werden soll.

**Schritt 3:**

Mit einem Klick auf den “Turn“ Button wird die Drehung ausgeführt.

# Formerkennung

## Theorie

### Objekterkennung

Um das Objekt des Bildes zu ermitteln werden die Pixel des Bildes anhand ihres Farbwertes analysiert und als Objektpixel oder Hintergrundpixel abgespeichert. Das Bild wird sozusagen in ein Schwarz-Weiss-Bild umgewandelt (Schwarz für Objektpixel/Weiss für Hintergrundpixel).

### Objektdefinition

Um Objekte zu maschinell zu vergleichen muss als erstes eine Definition zur Beschreibung von Objekten erstellt werden. Anfangs wollte ich die Objekte anhand der Anzahl Ecken und der Seitenverhältnisse vergleichen. Im Verlaufe der Realisierung habe ich mich dann entschieden anstatt der Seitenverhältnisse die Winkel in den Ecken zu verwenden.

**Beispiel:**

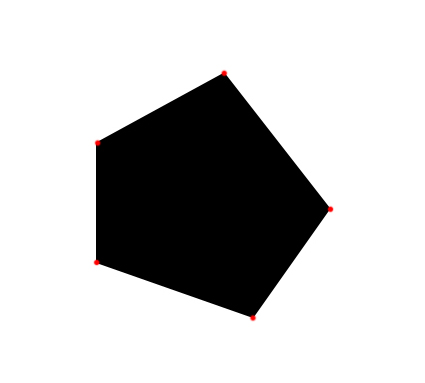


Figure 8 - Definition Objekt

Für das Fünfeck werden fünf Eckpunkte und deren Winkel gespeichert:

Punkt 1: 99°; Punkt 2: 118°; Punkt 3: 109°; Punkt 4: 105°; Punkt 5: 106°

### Ermittlung der Eckpunkte

Da ich mich in momentan auf die Erkennung von einfachen geometrischen Formen beschränke habe ich für die Erkennung der Eckpunkte ein sehr einfaches Verfahren verwendet.

Um die Eckpunkte zu ermittel werden die verschiedenen Kombinationen aus kleinsten X-/Y-Werten und grössten X-/Y-Werten ermittelt. Im folgenden Beispiel wird das Verfahren erläutert.

**Beispiel:**

In einem ersten Schritt werden die Objektpixel mit den kleinsten X-Koordinatenwerten ermittelt (Pixel am weitesten links):

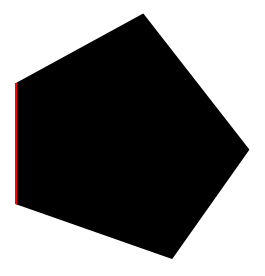


Figure - Ermittlung Eckpunkte

In diesem Fall wurden mehrere Pixel mit kleinstem X-Koordinatenwert ermittelt worden. Somit sind zwei Eckpunkte mit dem gleichen X-Koordinatenwert vorhanden. Um diese Punkte zu erhalten werden die Pixel mit dem Grössten und dem Kleinsten Y-Koordinatenwert ermittelt. Diese Punkte sind nun die Eckpunkte.

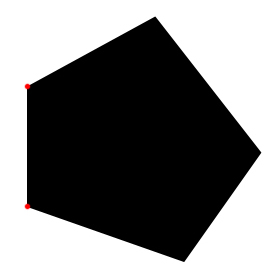
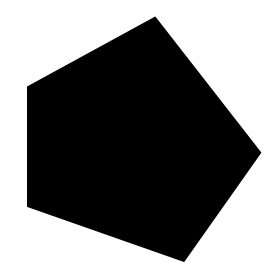


Figure 10 - Ermittlung Eckpunkte

Weiter werden die Punkte für den Kleinsten Y-Koordinatenwert, den Grössten Y-Koordinatenwert und den Grössten X-Koordinatenwert ermittelt.

### Berechnung der Winkel in den Eckpunkten

Der Winkel wird zwischen den beiden Vektoren des Eckpunktes zu seinen Nachbarpunkten gemessen.



α

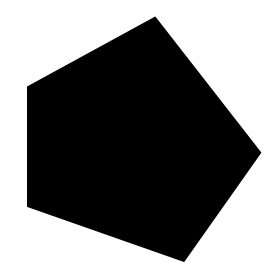
V

W

Figure 11 – Winkelmessung

Winkelmessung zwischen den beiden Vektoren V und W:

Um die Nachbarpunkte zu ermittel müssen die Eckpunkte anhand ihrer Position geordnet werden. Dafür wird vom Mittelpunkt des Objektes zu jedem Eckpunkt ein Vektor gelegt und den Winkel zum horizontalen Vektor (1/0) ermittelt. Wobei die Winkel für alle Vektoren in die gleiche Richtung gemessen werden (Gegenuhrzeigersinn). Die Eckpunkte werden den Winkelwerten nach aufsteigend geordnet.



Eckpunkt 5

Eckpunkt 4

Eckpunkt 3

Eckpunkt 2

Eckpunkt 1

Figure - Sortierung Eckpunkte

Für die Sortierung der Eckpunkte wird ebenfalls die oben beschriebene Methode zur Winkelberechnung zwischen zwei Vektoren verwendet.

### Vergleich von zwei Objekten

Beim Vergleich von zwei Objekten werden die Anzahl Eckpunkte sowie die Winkel in den Eckpunkten verglichen. Haben beide Objekte die gleiche Anzahl von Eckpunkten sowie die Winkelwerte in den Eckpunkten werden die Objekte als gleich betrachtet. Somit können Formen mit unterschiedlicher Grösse oder Ausrichtung erkannt werden.

**Beispiele:**

Die beiden Dreiecke werden trotz unterschiedlicher Grösse als gleich betrachtet.

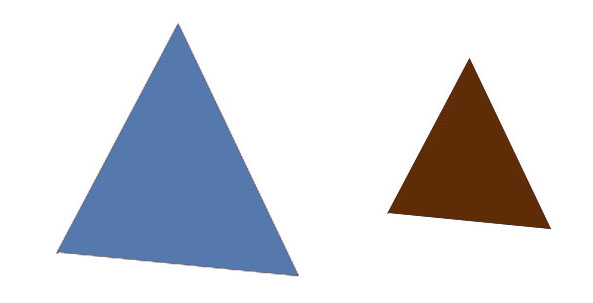


Figure - Vergleich Grösse

Auch die beiden Rechtecke werden als gleich betrachtet.

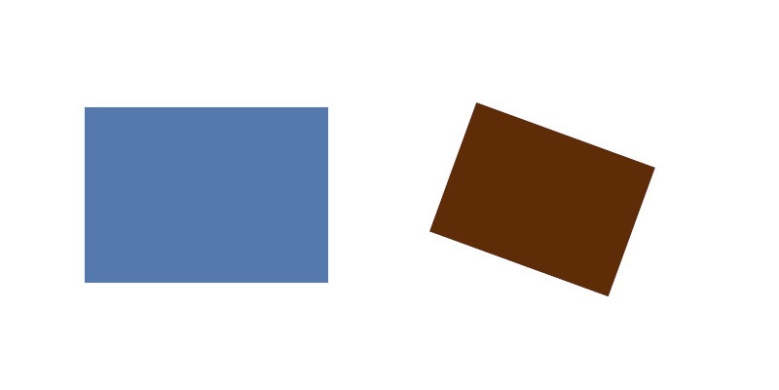


Figure - Vergleich Lage

### Aufgetretene Probleme

#### Erkennung der Eckpunkte

Das Erkennen der Eckpunkte bietet vor allem die Schwierigkeit, dass mehrere Pixel an den Extremwerten ermittelt werden. Es kann somit nicht einfach ein Pixel als Eckpunkt verwendet werden, allenfalls sind zwei Eckpunkte zu verwenden. (Beispiel Rechteck)

**Lösung:**

Bei der Ermittlung der Eckpunkte wird folgendes Regelwerk verwendet:



Figure - Regelwerk Eckpunktermittlung

## Umsetzung

In diesem Abschnitt wird die Handhabung des Programs für Formerkennung erläutert.

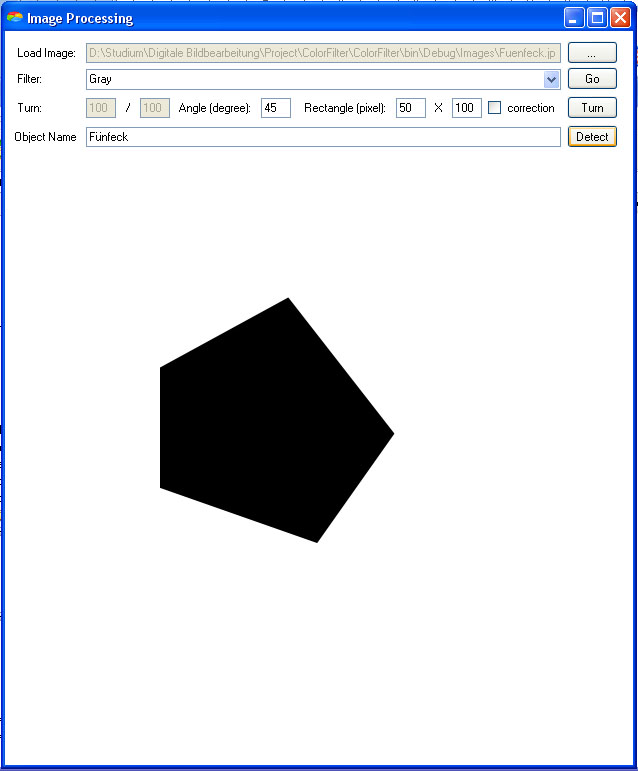


Figure - Program Formerkennung

**Schritt 1:**

Als erstes sollte eine sprechende Bezeichnung für die Form eingegeben werden.

**Schritt 2:**

Mit einem Klick auf Detect wird ermittelt ob bereits eine solche Form ermittelt wurde oder ob die Form neu ist und gespeichert werden muss.

Falls es sich um eine neue Form handelt erscheint folgende Meldung:

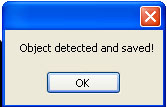


Figure 17 - Neue Form

Ist die Form bereits bekannt erscheint eine Meldung mit der Bezeichnung der bereits gespeicherten Form:



Figure - Bekannte Form

Nun können weitere Bilder geladen und Verglichen werden.

# Erweiterungsmöglichkeiten

Da für die Seminararbeit nur eine beschränkte Zeit von ca. 50 h eingesetzt werden sollten, musste ich viele Einschränkungen und Vereinfachungen anwenden. In diesem Kapitel möchte ich kurz einige Erweiterungs- und Verbesserungsmöglichkeiten aufführen.

## Bessere Eckpunkterkennung

Momentan verwende ich für die Erkennung der Eckpunkte einen sehr einfachen Mechanismus. Damit können viele Eckpunkte von Objekten nicht erkannt werden. Im folgenden Bild sind die erkannten und nicht erkannten Eckpunkte markiert (grün: erkannt; rot: nicht erkannt).

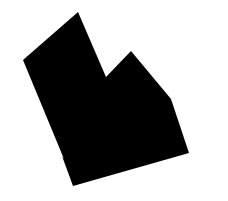


Figure 19 - Eckpunkterkennung

Mit einer Verbesserung der Eckpunkterkennung würde die Anzahl der erkennbaren Formen enorm anwachsen.

## Farbvergleich

Eine weitere Möglichkeit die Bilderkennung zu erweitern wäre die Objekte nicht nur anhand ihrer Form sondern auch anhand ihrer Farbe zu erkennen. Diese Erweiterung ist technisch nicht allzu schwierig zu implementieren.

## Mehrere Bildobjekte

Die jetzige Programversion kann nur mit einem Objekt pro Bild umgehen. Mit der Erweiterung auf mehrere Objekte pro Bild könnte eine weitere grössere Funktionserweiterung erreicht werden. Bei der Programmierung der aktuellen Version wurde eine solche Erweiterung bereits vorbereitet.

1. Stefan Adam: Matlab und Mathematik kompetent einsetzen, S. 198 [↑](#footnote-ref-1)