

Einführung
Aufgabe 1 - Bildverarbeitung

Anton Augsburg
2011



- 1 - Allgemeine Hinweise
- 2 - Grundlagen
- 3 - Aufgabenstellung
- 4 - C#
- 5 - Einführendes Beispiel
- 6 - Aufgabenhinweise

Übungsablauf

Einschreibung bei jExam!

Übungskalender mit allen Terminen verfügbar

5 Gruppen(je 30 Rechnerarbeitsplätze)

MO 4.DS (13:00-14:30) – ANTON AUGSBURG

MO 5.DS (14:50-16:20) – ANTON AUGSBURG

MI 2.DS (9:20-10:50) – MICHAEL MARTENS

MI 3.DS (11:10-12:40) – MICHAEL MARTENS

MI 4.DS (13:00-14:30) – TOBIAS REINSCH

Tutoren stehen für Fragen per email zur Verfügung

für selbstständiges Arbeiten die Raumbelegung im FRZ prüfen (E065 & E069)

bei Heimarbeit auf Kompatibilität mit denen im FRZ installierten Werkzeugen achten

Anwesenheitspflicht zu den Einführungen und zur Abschlusspräsentation (5 Termine)

Bewertung

jede Aufgabe wird mit Punktzahl bewertet, welche pro Aufgabe auf jedem Aufgabenblatt mit angegeben ist (Gesamtzahl 100 Punkte je Übung)

Zwischenabgabe bei jedem Aufgabenkomplex (orange gekennzeichnet)

- > digitale Abgabe auf den Server
- > Teilabgabe wird bewertet

Die Ergebnisse werden nach Abschluss der Bewertung für jede Aufgabe per email versandt – email-Adresse aus jExam wird verwendet, bitte prüfen!

Bewertung

Bewertung mit **NULL** Punkten, wenn:

- > keine Abgabe
- > Dateien mit vorgegebenen Werkzeugen nicht ausführbar/zu öffnen
 - > bei Arbeit zu Hause auf Kompatibilität achten, keine alternativen Werkzeuge erlaubt – **keine Ausnahmen!**
- > Ergebnis mit einem anderen identisch – jeder Teilnehmer muss eine eigenständige Lösung abgeben

Zulassungsvoraussetzung für die Klausur sind mindestens 200/400 Punkte (50%)

zusätzlich präsentieren einige Studenten in der Einführungsstunde ihre letzten Ergebnisse – keine Anwesenheit oder schlechter Vortrag gibt 25 Punkte Abzug (bei vorher bestandener Übung)

Aufgabe 1 - Bildverarbeitung

FRZ – Zugriff

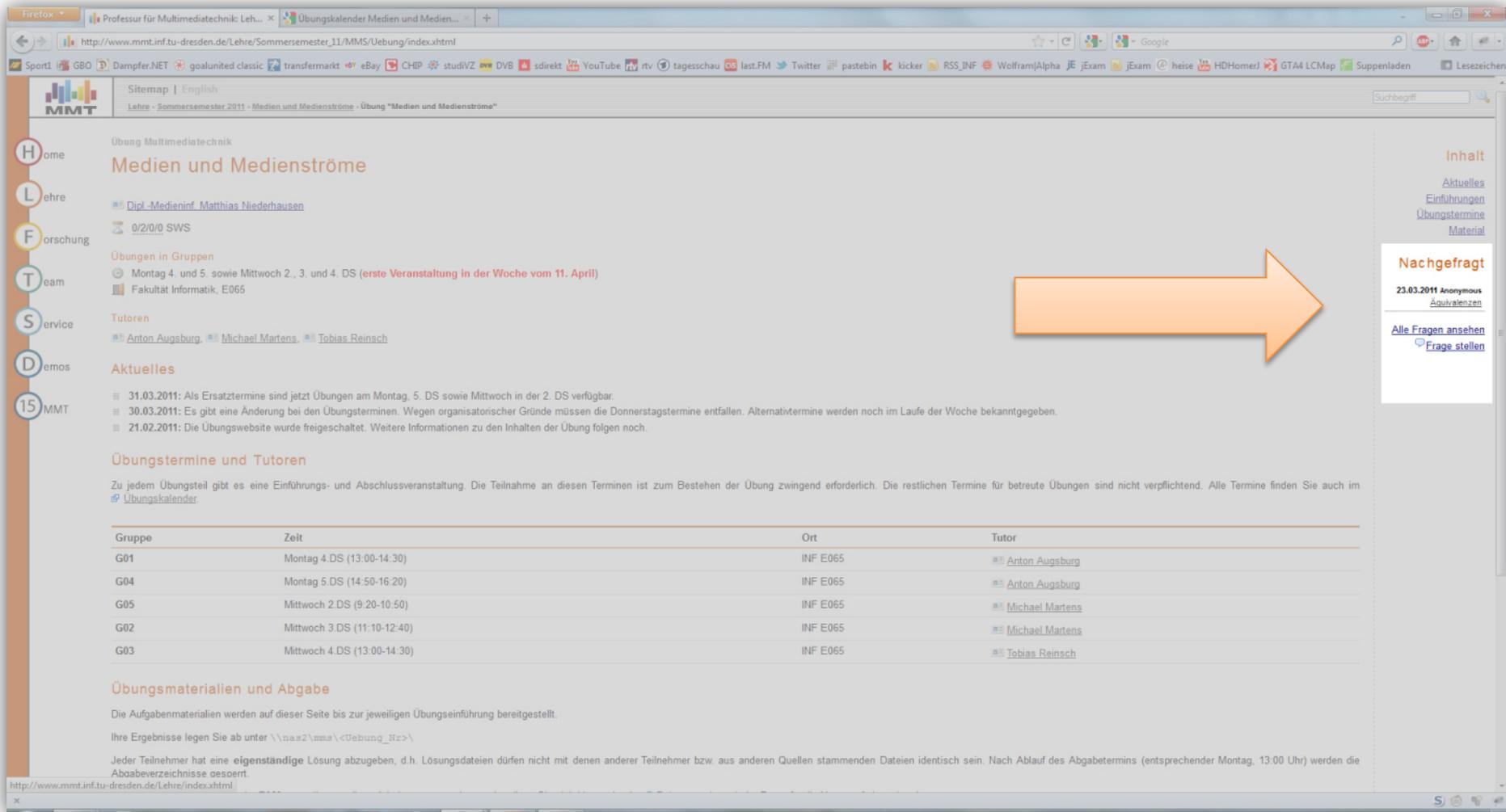
Explorer > Extras > Netzlaufwerk verbinden > Laufwerk: V,
Ordner: *\\nas2\mms*

SFTP – Zugriff

Empfehlung: WinSCP
Benutzername: ZIH – Login (sXXXXXXXX)
Kennwort: ZIH – Kennwort
Portnummer: 22
Servername (Rechnername): serv9.inf.tu-dresden.de
Verzeichnis: */zbv/mms*

Allgemeine Hinweise

Aufgabe 1 - Bildverarbeitung



Firefox | Professur für Multimedialechnik: Leh... | Übungskalender Medien und Medien... | +

http://www.mmt.inf.tu-dresden.de/Lehre/Sommersemester_11/MMS/uebung/index.shtml

MMT | Sitemap | English

Lehre - Sommersemester 2011 - Medien und Medienströme - Übung "Medien und Medienströme"

Übung Multimedialechnik

Medien und Medienströme

Dipl.-Medieninf. Matthias Niederhausen

0/2/0/0 SWS

Übungen in Gruppen

- Montag 4. und 5. sowie Mittwoch 2., 3. und 4. DS (erste Veranstaltung in der Woche vom 11. April)

Fakultät Informatik, E065

Tutoren

- Anton Augsburg, Michael Martens, Tobias Reinsch

Aktuelles

- 31.03.2011: Als Ersatztermine sind jetzt Übungen am Montag, 5. DS sowie Mittwoch in der 2. DS verfügbar.
- 30.03.2011: Es gibt eine Änderung bei den Übungsterminen. Wegen organisatorischer Gründe müssen die Donnerstagstermine entfallen. Alternativtermine werden noch im Laufe der Woche bekanntgegeben.
- 21.02.2011: Die Übungswebsite wurde freigeschaltet. Weitere Informationen zu den Inhalten der Übung folgen noch.

Übungstermine und Tutoren

Zu jedem Übungssteil gibt es eine Einführungs- und Abschlussveranstaltung. Die Teilnahme an diesen Terminen ist zum Bestehen der Übung zwingend erforderlich. Die restlichen Termine für betreute Übungen sind nicht verpflichtend. Alle Termine finden Sie auch im [Übungskalender](#).

Gruppe	Zeit	Ort	Tutor
G01	Montag 4.DS (13:00-14:30)	INF E065	Anton Augsburg
G04	Montag 5.DS (14:50-16:20)	INF E065	Anton Augsburg
G05	Mittwoch 2.DS (9:20-10:50)	INF E065	Michael Martens
G02	Mittwoch 3.DS (11:10-12:40)	INF E065	Michael Martens
G03	Mittwoch 4.DS (13:00-14:30)	INF E065	Tobias Reinsch

Übungsmaterialien und Abgabe

Die Aufgabenmaterialien werden auf dieser Seite bis zur jeweiligen Übungseinführung bereitgestellt.

Ihre Ergebnisse legen Sie ab unter `\\nas2\mms\<Übung_Nr>`

Jeder Teilnehmer hat eine **eigenständige** Lösung abzugeben, d.h. Lösungsdateien dürfen nicht mit denen anderer Teilnehmer bzw. aus anderen Quellen stammenden Dateien identisch sein. Nach Ablauf des Abgabetermins (entsprechender Montag, 13:00 Uhr) werden die Abgabeverzeichnisse oesuert.

Navigation: Home, Lehre, Forschung, Team, Service, Demos, MMT

Suchbegriff

Inhalt: Aktuelles, Einführungen, Übungstermine, Material

Nachgefragt: 23.03.2011 Anonymous, Äquivalenzen

Alle Fragen ansehen, Frage stellen

Bildaufbau

Sequenz von Bildpunkten, angeordnet in
X Spalten + Y Zeilen

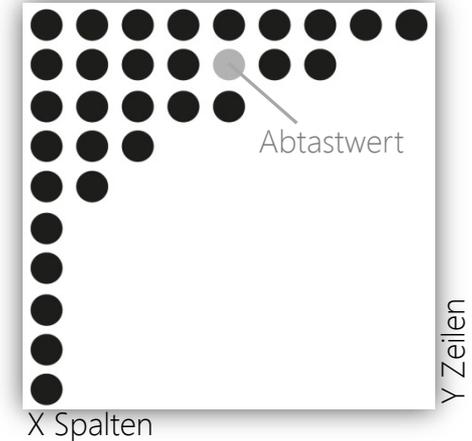
Bildpunkt charakterisiert durch Farbwert

Farbwert ist Punkt im Farbraum des
jeweiligen Systems (RGB, CMY, $Y C_R C_B$)

Farbtiefe ist Anzahl Bits/Bildpunkt

Bildebenen sind abhängig von Attributen der Bildpunkte
bei Farbbildern i.A. 3 Ebenen (z.B. R,G,B)
Graustufenbildern genügt 1 Ebene

Punktoperationen betreffen jeweils nur einzelnes Pixel
homogen, wenn die Operation für alle Bildpixel gleich ist (z.B. Aufhellen)
inhomogen, wenn selektiv angepasst wird (z.B. selektive Kontrastanpassung)



Vorgehen

- ① Idee in eine Funktion überführen
Ziel: Bild als Schwarz-Weiß-Variante
Funktion: Grauwert p eines Pixels soll für dunkle Bereiche durch Schwarz und helle Bereiche durch Weiß ersetzt werden
- ② Algorithmus formulieren
 w = Bildbreite; h = Bildhöhe;
zähle x von 0 bis $w-1$
zähle y von 0 bis $h-1$
 p = Bild(x,y);
wenn ($p < 127$): Bild(x,y) = Schwarz;
sonst: Bild(x,y) = Weiß;
- ③ Implementieren

Zielstellung

Kennenlernen der Wirkungsweise eines Bildbearbeitungsprogramms

- > Bild spiegeln, invertieren, ...
- > Bildverbesserung (Kontrastverstärkung, Filter)
- > algorithmische Grundlagen

Bearbeitungszeitraum

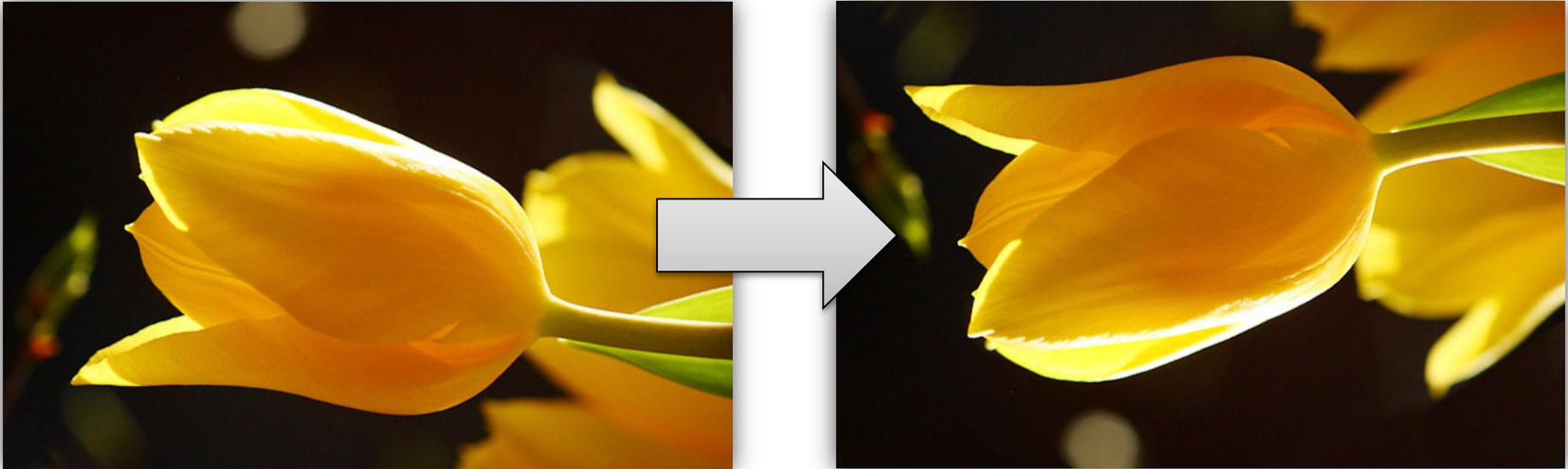
11.04. – 02.05.2011 (Abgabe bis 13.00 UHR)

Verwendete Software

Microsoft Visual Studio 2008 (im FRZ installiert)
auch Visual C# Express Version möglich

Aufgabe 1 – Bildspiegelung

Das Eingangsbild soll so verändert werden, dass es vertikal gespiegelt ausgegeben wird.



Aufgabe 2 – Dithering mit dem Floyd-Steinberg-Algorithmus

Auf das in Graustufen umgewandelte Ausgangsbild soll der *Floyd-Steinberg-Algorithmus* angewendet werden.

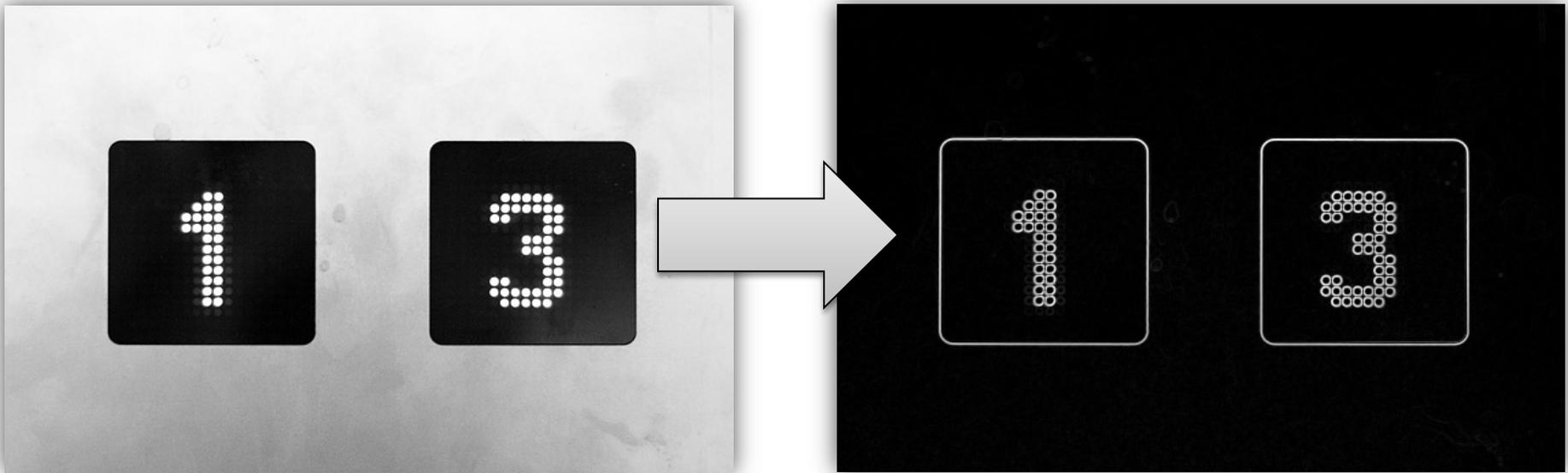
Dafür nötig ist unter anderem die nebenstehende Berechnungsmatrix zur Verteilung des Quantisierungsfehlers.

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & P & \frac{7}{16} \\ \frac{3}{16} & \frac{5}{16} & \frac{1}{16} \end{pmatrix}$$



Aufgabe 3a – Sobelfilter

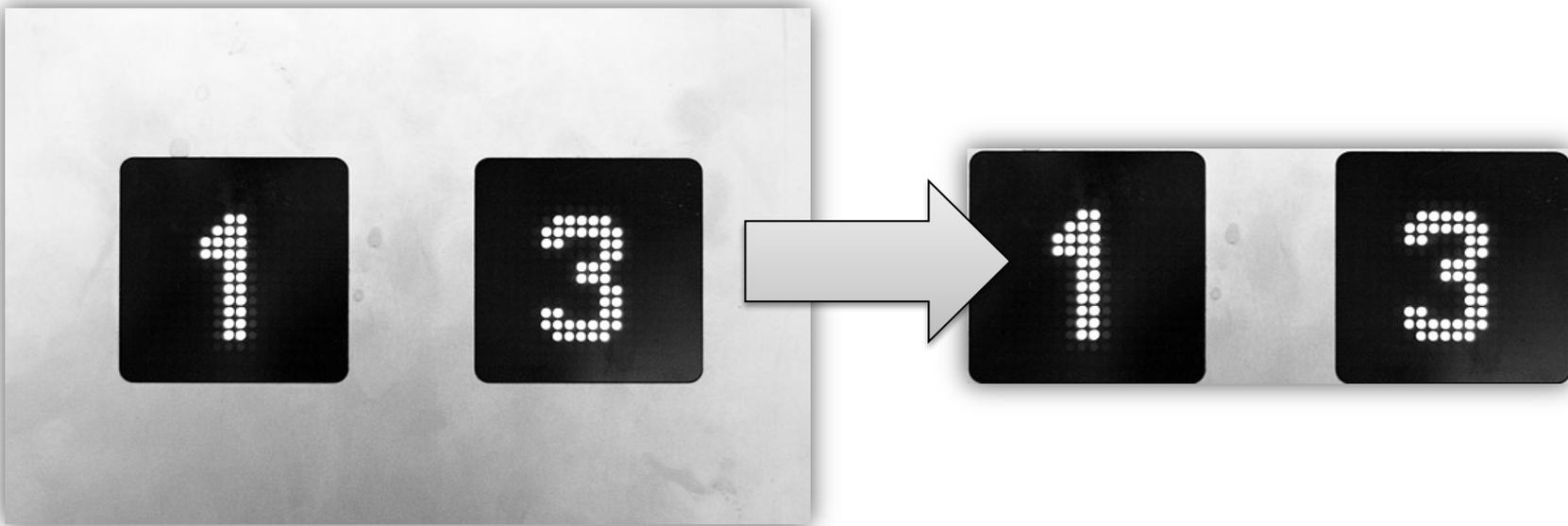
Auf das Eingangsbild soll der zweidimensionale Sobelfilter angewendet werden. Mit dem Sobelfilter lassen sich Kanten im Bild finden und darstellen. Nutzen Sie dazu die vorgegebenen 3x3 Sobel-Filtermatrizen für die horizontale (S_x) sowie vertikale (S_y) Filterung. Vereinen Sie beide Filtereffekte in einem Bild.



Aufgabe 3b – Zuschnitt

Das Eingangsbild soll nun auf Basis der berechneten Kanten zugeschnitten werden.

Wichtig ist dabei nicht der pixelgenaue Zuschnitt, die grundlegende Funktion des Algorithmus ist wichtig.



Objektorientierte Programmierung

„Die Grundidee dabei ist, Daten und Funktionen, welche auf diese Daten angewandt werden können, möglichst eng in einem sogenannten Objekt zusammenzufassen und nach außen hin zu kapseln, so dass Methoden fremder Objekte diese Daten nicht versehentlich manipulieren können.“ - Wikipedia

Methodenaufrufe, Variablendeklarationen, Schleifen, Operatoren, etc. wie in Java

Weiterführende Ressourcen

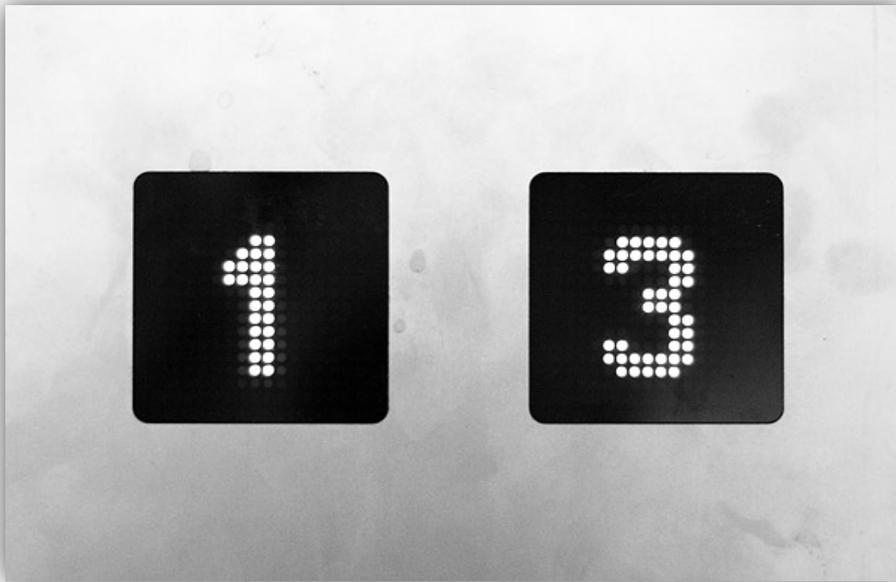
[Galileo Openbook - C#](#)

[Visual C# Developer Center](#)

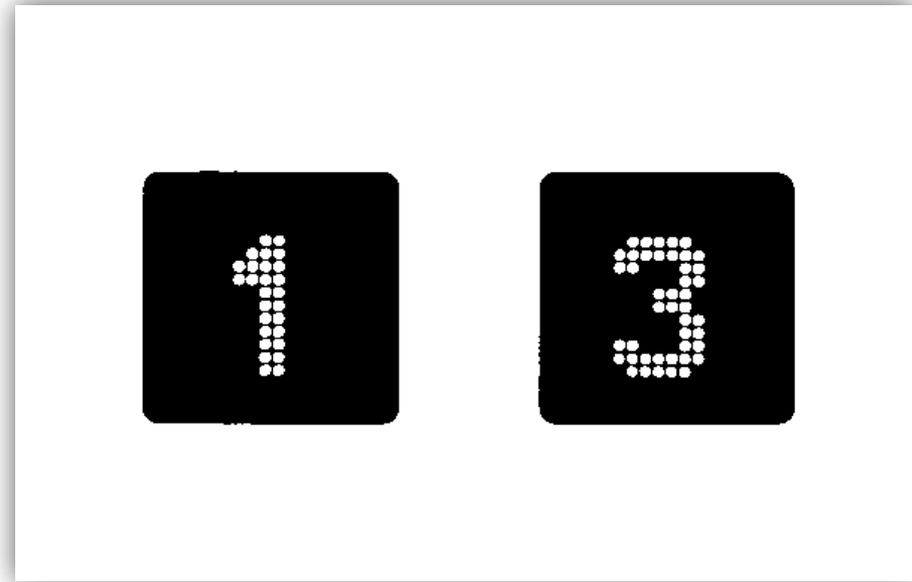
[C# Dokumentation](#)

Schwarz-Weiss-Filter

Umwandlung von hellen Pixeln in weiß, dunkle Pixel in schwarz
Schwellwert von 127 (Wertebereich 0...255)



Original



gefiltert

Aufgabe 1 - Bildverarbeitung

```
[...]  
{  
class Example  
{  
public Bitmap run(Bitmap inputimage)  
{  
    Bitmap outputimage = new Bitmap(inputimage.Width, inputimage.Height);  
    Color black = Color.FromArgb(0, 0, 0);  
    Color white = Color.FromArgb(255, 255, 255);  
  
    for (int x = 0; x < inputimage.Width; x++){  
        for (int y = 0; y < inputimage.Height; y++){  
            if ((inputimage.GetPixel(x, y).R + inputimage.GetPixel(x, y).G +  
                inputimage.GetPixel(x, y).B) / 3 < 128){  
                outputimage.SetPixel(x, y, black);}  
            else {  
                outputimage.SetPixel(x, y, white);}  
            }  
        }  
    }  
    return outputimage;  
}}}
```

Als Erstes..

Visual Studio 2008 oder Visual C# 2008 starten

im Materialordner findet man eine Solutiondatei (*MUMS_Uebung1.sln*),
diese in VS einladen. (File > Open Project > ...)

Was muss ich machen?

rechts im Solution-Explorer sind einige Dateien aufgelistet, für die Übung sind
nur vier *Exercise#.cs* Klassen relevant

in jeder dieser Klassendefinitionen befindet sich eine run-Methode, welche ein
Bitmapobjekt übergeben bekommt - dieses ist das Original- , also Eingangsbild

die Methode gibt ebenfalls ein Bitmapobjekt (*outputimage*) zurück - das bearbeitete Bild

nur #region implementieren

Zum Schluss..

muss jede der vier Exercise-Klassen die gegebene Aufgabe bewältigen
Transformation inputimage -> outputimage

Anton Augsburg
2011

