

Einführung
Aufgabe 3 - MPEG

Tobias Reinsch
2011



Ziel der Aufgabe

- Kennenlernen der Bildkodierungsverfahren des MPEG Standards
- Praktische Umsetzung dieser Techniken mit Java

Werkzeug: Eclipse 3.6.2 Helios

Bearbeitungszeitraum: 30. Mai – 20. Juni

Zwischenabgabe: 8. Juni

MPEG

- Moving Picture Experts Group
- Gruppe von Experten, die sich mit Standardisierung von Videokompression befassen (ca. 350 Personen aus 200 Unternehmen)
- Resultierende Formate u.A.: H.261, MPEG 1,2,3,4,7,21, H.264
- Standards für Audio und Videokompression

- In der Übung werden einige Kodierungs- und Kompressionsverfahren aus dem MPEG-1 Standard praktisch angewandt

MPEG -1 Videokodierung

- in den 80er Jahren entwickelt
- für Datenraten < 1.5 Mbit/s zur Speicherung auf CD (Video-CD)
- Audiokompression mit MP 1-3
- Bildformat ähnelt JPEG

- Farbkodierung im YCbCr Format
- Reduktion der Datenrate mit Unterabtastung der Cb und Cr Kanäle
- zur weiteren Reduzierung werden Bildfolgen differenzkodiert, mit Hilfe von 8x8 Makroblöcken

Group of Pictures (GOP)

- I-Frame : Ausgangsbild für die Differenzkodierung
- P-Frame : Differenzkodierung aus vorausgegangenen I und P Frames
- B-Frame : Bidirektionale Differenzkodierung aus I und P – Frames

MPEG -1 Makroblöcke und DCT-Transformation

- für die Kodierung der Makroblöcke wird bei MPEG die Diskrete-Cosinus-Transformation (DCT) verwendet
- diese wandelt die räumliche Information der Farbwerte in eine Frequenz-Darstellung
- dabei werden sich überlagernde Cosinus-Kurven genutzt um die Farbwerte zu approximieren
- große Kurven repräsentieren Farbverläufe von einer Blockseite zur Anderen
- kleine Kurven repräsentieren Farbunterschiede zwischen einzelnen Pixel
- In den P und B Frames werden die DC - Koeffizienten gespeichert
- Wesentlicher weniger Informationen als bei Pixeldarstellung benötigt
- Gute Resultate bei Kompression
 - Jedoch werden bei starker Kompression Kanten zwischen den Makroblöcken sichtbar, da nur noch die Koeffizienten der große Wellen gespeichert werden

Einfachheitshalber werden diese Verfahren in der Übung nicht betrachtet

Die Aufgabe im Überblick:

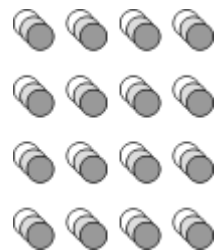
- RGB – Bild in YCbCr transformieren
- 4:2:0 Unterabtastung des YCbCr Bildes durchführen
- Bild in Makroblöcke aufteilen
- Blocksuche in einem 2.Bild und Differenzkodierung

Aufgaben:

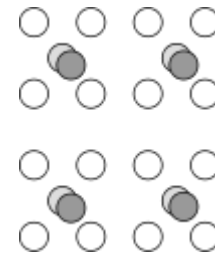
- Gegebenes Bild aus dem RGB Farbraum in YC_bC_r transformieren
- Für jeden Pixel muss folgende Operation durchgeführt werden:

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} + \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,1687 & -0,3313 & 0,5 \\ 0,5 & -0,4187 & -0,0813 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

- 4:2:0 Subsampling auf das YC_bC_r Bild anwenden



YC_bC_r 4:4:4



YC_bC_r 4:2:0

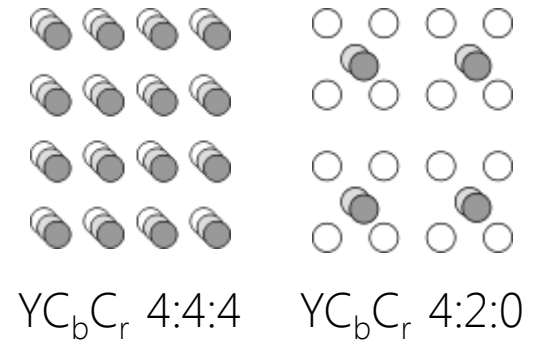
Zwischenabgabe bis: 8. Juni 13 Uhr !

YC_bC_r

- Wurde für das PAL Farbfernsehen entwickelt
 - Y – Grundhelligkeit
 - C_b – Blau-Gelb Farbkomponente
 - C_r – Grün-Rot Farbkomponente
- Entspricht besser dem Sehsinn der Menschen
- Unterabtastung ohne Verringerung der Ortsauflösung möglich

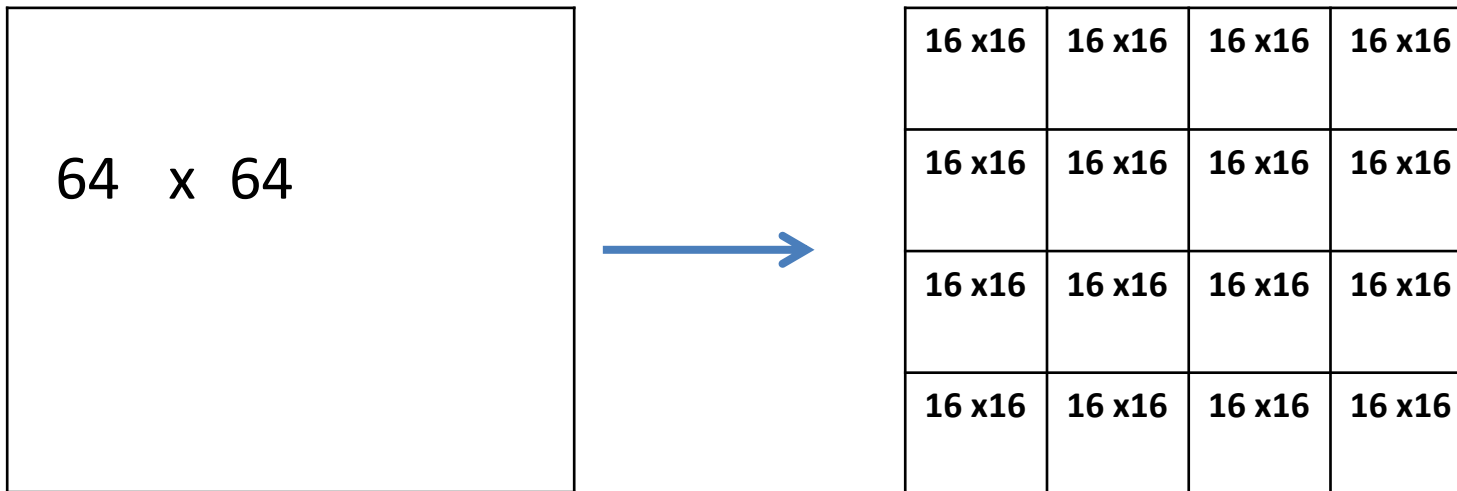
Subsampling (Unterabtastung)

- Reduzierung der Auflösung der C_b C_r Kanäle
- 4:2:0 → Halbe Datenrate
- Kaum sichtbare Qualitätseinbußen



Aufgaben:

- Gegebenes Bild in kleinere Bilder mit 16x16 Pixel unterteilen (Makroblöcke)
- Verwendung des Makroblock Datenformat im Framework
- Beachtung der Unterabtastungsaufösung
- Speichern der Makroblöcke für beide Frames in .blk Dateien mit dem Framework

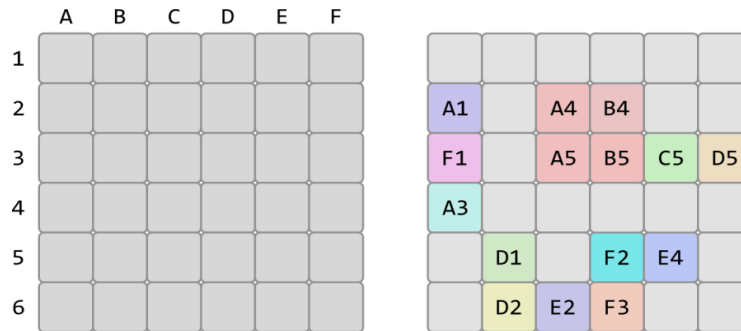


Aufgaben:

- Von Makroblöcken aus Aufgabe 3.2 soll Frame 2 als P-Bild kodiert werden
- *P-Bild*: differenz-kodiertes Bild welches nur Differenzinformationen zu dem vorausgehenden Bild(ern) enthält
- Implementierung eines Blocksuch-Algorithmus (freie Wahl), welcher Makroblöcke aus Frame 1 in Frame 2 sucht
- Als Ähnlichkeitsmaß soll die summierte Differenz der Farbwerte mit einem Schwellwert von 9000 verwendet werden

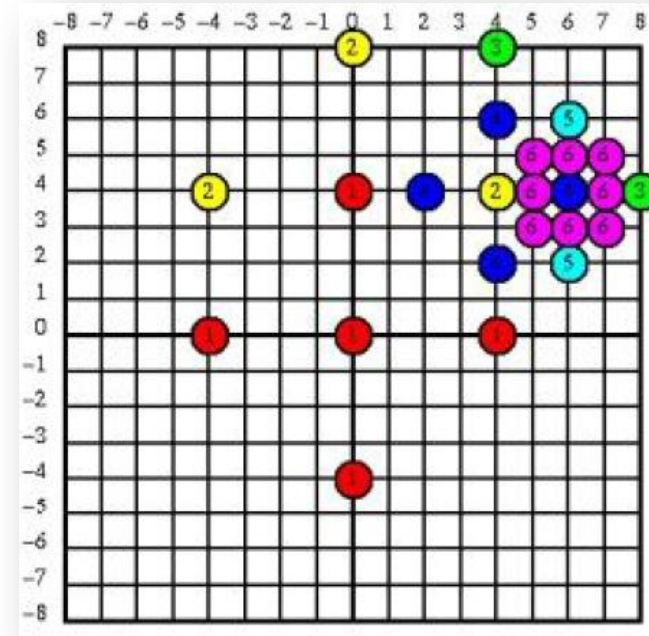
Aufgaben:

- Erzeugung einer Matrix mit Differenz-Makroblöcken mit folgenden Regeln:
 - *Wurde ein ähnlicher Block im vorherigen Bild gefunden, speichere die relativen Koordinaten des Originalblocks sowie eine Differenzmatrix für die YC_bC_r -Ebenen.*
 - *Wurde kein ähnlicher Block im vorherigen Bild gefunden, speichere den Makroblock „normal“, d.h. wie in Aufgabe 3.2 ermittelt.*
- Speichern des P-Frames als p-frame.blk mit Hilfe des Frameworks



Heuristische, zweidimensionale logarithmische Suche:

- Koordinatensystem stellt Suchbereich S dar
- Kreise sind Makroblöcke, Nummern Verarbeitungsschritte
- Im 1.Schritt (Rot) wird kreuzweise Ähnlichkeit ermittelt
- Wiederholung des 1.Schrittes mit ähnlichstem Block als Mittelpunkt (gelb)
- Stimmt mittlerer Block am besten überein, wird Schrittweite halbiert und wieder der 1.Schritt durchgeführt (Blau)
- Bei Schrittweite von 1 werden alle 8 umliegenden Blöcke untersucht (Pink) . Der ähnlichste Block wird dann gespeichert



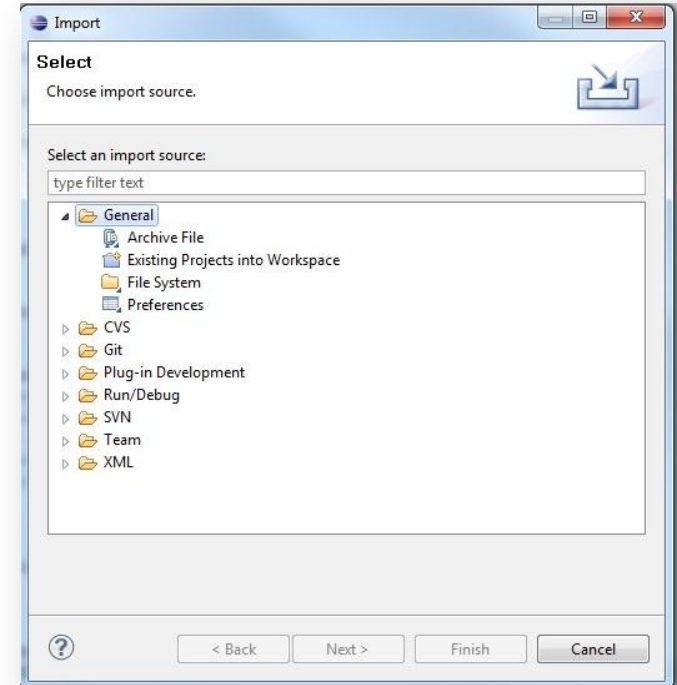
- Für die Lösung der Aufgaben steht ein **Java** - Framework zur Verfügung
- Bereitgestellt als Eclipse-Projekt

Funktionen:

- Initiale Klassen für die 3 Aufgaben
- Datenstrukturen:
 - MatrixTriple: speichert Frame mit 3 Farbkanälen für beide Farbmodelle
 - MacroBlock: speichert Makroblock - feste Blockgröße
- GUI zum Testen der Funktionen und speichern der Makroblock-Dateien

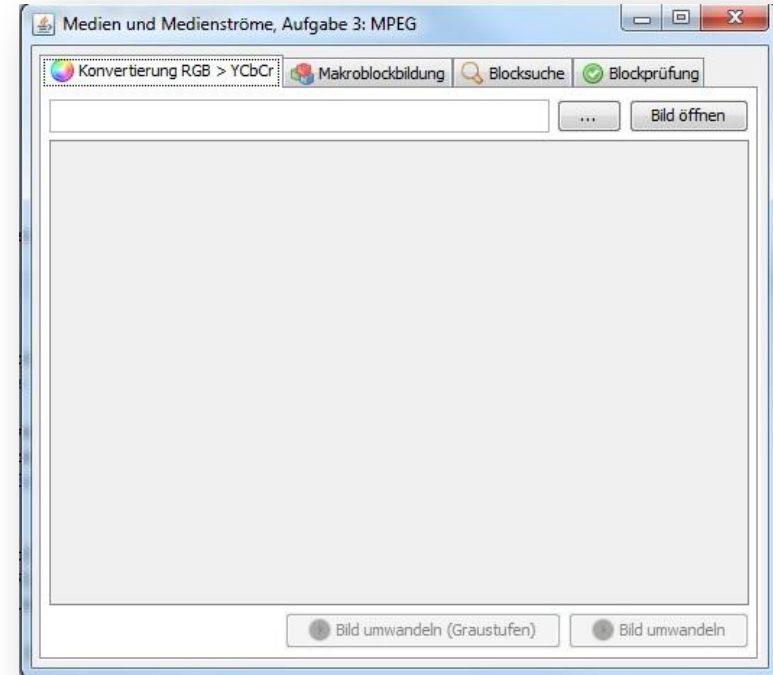
Installation:

- Download des Projekts
- Eclipse öffnen
- Im Package Explorer Importdialog öffnen
- Projekt entweder als .zip Datei oder Ordner einladen
- Die Aufgaben befinden sich vorbereitet als Java-Klasse im Package: **de.mmt.mpeg**



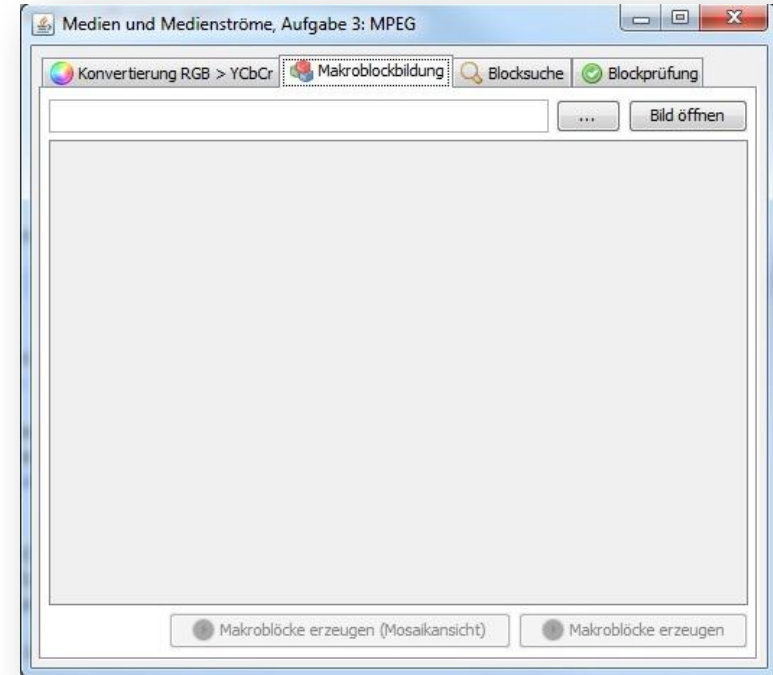
Konvertierung RGB zu YCbCr:

- Bild laden mit „*Bild öffnen*“
- „*Bild umwandeln*“ transformiert Bild in YCbCr (4:2:0) und wieder zurück (für die Darstellung notwendig)
- Bei Fehler im Algorithmus werden Bildfehler sichtbar
- „*Bild umwandeln (Graustufen)*“ wandelt Bild zu Testzwecken in Graustufen um



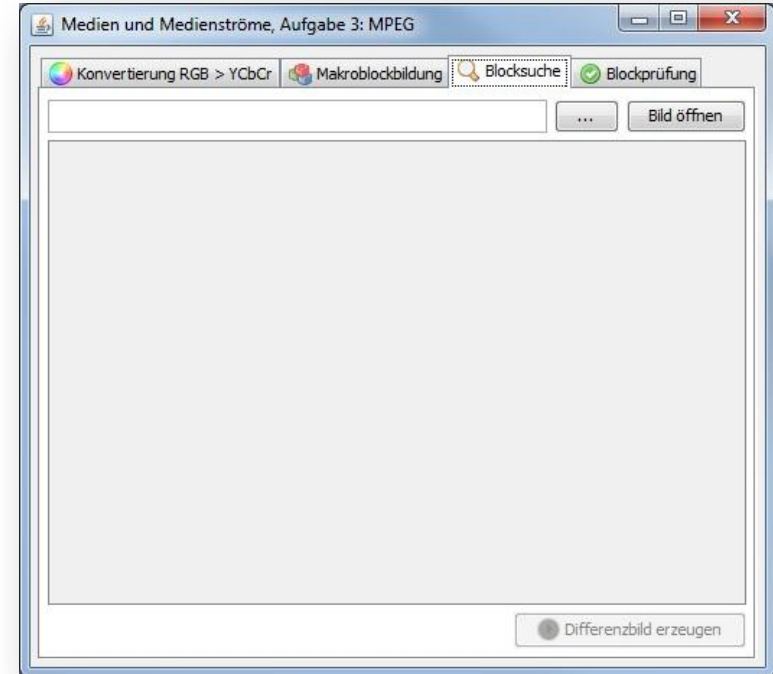
Makroblockbildung:

- Bild laden mit „*Bild öffnen*“
- „*Makroblock erzeugen*“ transformiert Bild in Makroblöcke und speichert als .blk Datei
- „*Mosaikansicht*“ stellt die Makroblöcke als übergroße Pixel dar



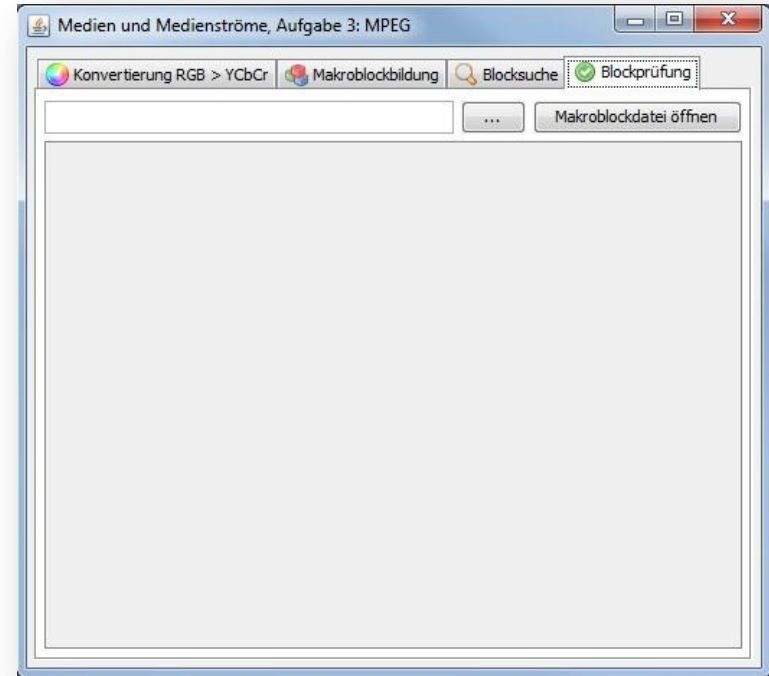
Blocksuche:

- zu kodierendes Bild laden mit „*Bild öffnen*“
(Frame 2)
 - „*Differenzbild*“ lädt Bild, gegen welches
differenzkodiert werden soll (Frame 1)
- Differenzblöcke werden als .blk Datei gespeichert



Blockprüfung:

- Makroblockdatei laden mit
„Makroblockdatei öffnen“
- Wandelt Makroblockdatei in Bild und stellt es dar
 - Bei Differenzblöcken muss noch das Ausgangsbild geladen werden (Frame 1)



Links

Java Einführung

- <http://openbook.galileocomputing.de/javainsel/Handbuch>

Eclipse Tutorial

- <http://www.del-net.com/download/Eclipse31Einfuehrung.pdf> Schnellreferenz

Tobias Reinsch
2011

