Einführung in die Multimedia-Technik **WS 2013/2014**

Posch

Übungsblatt 2

Themen: Schallwellen, (D)PCM, Digitale Bilder, Farbräume, Subsampling

Ü 2.1 Schallwellen / PCM

Beantworten Sie folgende Fragen:

a. Was ist und wie entsteht Schall?
b. Mit welcher Geschwindigkeit breitet sich Schall in der Luft/ im Wasser aus?
c. Erläutern Sie folgende physikalische Parameter einer Schallwelle:
   - Frequenz
   - Amplitude
   - Phase
d. In welcher Verbindung stehen die folgenden Begriffe mit den zuvor genannten physikalischen Parametern:
   - Lautstärke
   - Tonhöhe

Erklären Sie was die Pulse Code Modulation (PCM) ist. Angenommen die Schallwellen die von einem Menschen beim Sprechen erzeugt werden, befinden sich im Frequenzbereich von 50 Hz bis 10 kHz. Bestimmen Sie mit Hilfe des *Nyquist-Shannon-Abtasttheorem* die notwendige Abtastrate mit der das analoge Signal (Sprache) abgetastet werden sollte, um es wieder korrekt rekonstruieren zu können. Wie hoch ist die Bitrate des entstehenden digitalen Signales, wenn Sie für die PCM eine Quantisierung des Wertebereichs von a) 8 bit b) 16 bit c) 24 bit vornehmen?

Ü 2.2 Predictive Coding / DPCM

Erklären Sie was man unter dem Begriff *Predictive Coding* versteht. Was erhofft man sich durch diese Technik? Erläutern Sie den Begriff Differential Pulse Code Modulation (DPCM) und finden Sie heraus welcher Zusammenhang zwischen DPCM, PCM und Predictive Coding besteht.

Angenommen das vorgegebene DPCM Schema nutze folgende Variablen-Bezeichnungen:

f sei eine Integer-Folge von Input Signalen.

f_n sei der n-te (tatsächliche) Wert in der Folge

$\hat{f}_n$ sei der n-te vorausgesagte (predicted) Wert der Folge

e_n sei der n-te Fehler zwischer f_n und $\hat{f}_n$

$\tilde{e}_n$ sei der n-te quantisierte Fehler

$\tilde{f}_n$ sei der n-te rekonsturierte Wert

Die Werte der Variablen ergeben sich im gegebenen Schema aus folgenden Gleichungen:

$$\hat{f}_n = \text{trunc}\left[\left(\tilde{f}_{n-1} + \tilde{f}_{n-2}\right)/2\right]$$

$$e_n = f_n - \hat{f}_n$$

$$\tilde{e}_n = Q[e_n] = 16 * \text{trunc}\left[(255 + e_n)/16\right] - 256 + 8$$

transmit $codeword(\tilde{e}_n)$

reconstruct: $\tilde{f}_n = \hat{f}_n + \tilde{e}_n$

Berechnen Sie welche Codewörter entstehen, wenn das oben genannte DPCM Schema genutzt wird und als Input die Folge: (100, 120, 150, 110, 105) vorliegt. Welche Werte müssen gespeichert werden, damit eine Rekonstruktion möglich ist? Welche Werte können rekonstruiert werden?

Hinweis(e):

- Die Funktion „trunc“ erzeugt einen ganzzahligen Integer, indem die Nachkommastellen einer Gleitkommazahl verworfen werden.
- Duplizieren Sie den ersten Wert der Folge zweimal bevor Sie mit der Generierung des DPCM Codes beginnen und initalisiern Sie den quantisierten Fehler $\tilde{e}_n$ für n:=1 mit 0. Dadurch erreichen Sie, dass der erste rekonstruierte Wert nicht verfälscht wird.

Ü 2.3 DPCM Implementierung

Schreiben Sie ein Java/C/C++/C# Programm, das folgende Aufgaben erfüllen kann:

a) Einlesen einer Zahlenfolge aus der Datei *sequence.csv* (im Moodle verfügbar).
b) Codieren der Zahlenfolge mittels DPCM. Speichern Sie das Ergebnis in eine Datei.
c) Einlesen der erzeugten Datei und Rekonstruktion der Folge ausschließlich aus den Informationen, die in der Datei abgespeichert wurden.

Ihr Programm arbeitet korrekt, falls die Inputfolge

```
130 150 140 200 230 232 238 250 230 220 210 207 200 100 90 70 76 80 82
```

in die Outputfolge

```
130 154 134 200 223 235 237 244 232 214 215 206 202 100 95 73 76 82 87
```

rekonstruiert wird.

Hinweis: Für das zu implementierende DPCM Schema gelten die selben Gleichungen wie in Aufgabe Ü 2.2. Beachten Sie auch den Hinweis aus Aufgabe Ü 2.2.

Ü 2.4 Digitale Bilder / Farbräume / Subsampling (Farbunterabtastung)

Definieren Sie folgende Begriffe und erläutern Sie deren Zusammenhänge, beziehungsweise grenzen Sie die Begriffe voneinander ab. Zeichnen Sie Skizzen wenn notwendig:

a. Farbraum
   - Farbaddition
   - Farbsubtraktion

b. Digitales Bild
   - Pixel
   - Farbtiefe
   - Auflösung

c. Subsampling

Erklären Sie wie folgende Farbräume aufgebaut sind, wie sie sich unterscheiden und für welches Einsatzgebiet sie praktisch genutzt werden: *RGB, YUV, HSV, CMYK*.

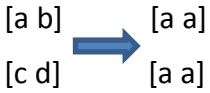
Konvertieren Sie die gegebenen Werte von dem angegebenen Ausgangsfarbraum in den festgelegten Zielfarbraum:

Ausgangsfarbraum	Werte	Zielfarbraum
RGB	R=100; G=150; B=80	YUV
HSV	H=218°; S=50%; V=90%	RGB
CMYK	C=0.65; M=0.3; Y=0.6; K=0.2	RGB

Ü 2.5 Subsampling – Implementierung

Schreiben Sie ein Java/C/C++/C# Programm, welches das Bild *lena_512x512.png* (im Moodle verfügbar) einliest. Greifen Sie auf die RGB Werte des Bildes zu und implementieren Sie folgendes einfaches Subsampling Verfahren. Ihre Funktion bekommt als Input nur das Bild und eine Blockgröße für das Subsampling übergeben. Unterteilen Sie die Bilddaten (Integer-Matrix) in quadratische Blöcke der übergebenen Blockgröße und weisen Sie allen Punkten in einem Block den Wert des ersten Elements dieses Blocks zu.

Beispiel für das Subsampling eines 2x2 großen Blocks:



Speichern Sie das erzeugte Bild nach dem Subsampling wieder ab. Vergleichen Sie die ursprüngliche Dateigröße mit der Dateigröße nach dem Subsampling. Führen Sie dies für folgende Blockgrößen durch: 2x2, 4x4, 8x8. Beschreiben Sie welche Veränderungen Sie am Bild wahrnehmen können. Überlegen Sie sich **(keine Implementierung notwendig)** wie man das Subsampling verbessern könnte. Machen Sie mindestens drei konkrete Vorschläge.

Hinweis: In der Sprache Java erlaubt die Klasse *BufferedImage* einen einfachen Zugriff auf die einzelnen Pixelwerte eines Bildes.

BITTE LADEN SIE IHRE LÖSUNGEN AUF DER MOODLE HOMEPAGE HOCH!