

## Hauptdiplomklausur Informatik Juli 2002: Multimedia-Technik

Name: ..... Vorname: .....

Matrikel-Nr.: ..... Semester: ..... Fach: .....

*Hinweise:*

1. Bitte füllen Sie sofort den Kopf des Deckblattes aus.
2. Überprüfen Sie bitte Ihr Klausurexemplar auf Vollständigkeit (**10** Seiten).
3. Tragen Sie die Lösungen – soweit möglich – direkt in die Klausur ein.
4. Zugelassene Hilfsmittel: nicht programmierbarer Taschenrechner
5. Bearbeitungszeit: 100 Minuten.

Aufgabe	max. Punktzahl	Punkte
1	15	
2	13	
3	8	
4	12	
5	19	
6	18	
7	15	
Summe	100	

## **Aufgabe 1: Verlustfreie Kompression [5+2+2+2+4=15 Punkte]**

- (a) [5 Punkte] Gegeben sei eine Zuordnung von Zeichen zu Bitmustern. Die Bitmuster haben variable Länge und keinen einheitlichen Trenner, so wie z. B. bei Huffman Codes. Beschreiben Sie als Pseudocode (auch Java oder C), wie man die Eindeutigkeit der Zuordnung prüfen kann, ohne einen Baum aufzubauen und ohne einen Text zu kodieren.
- (b) [2 Punkte] Konstruieren Sie einen beispielhaften Datenstrom von mindestens zehn Zeichen, der mehr von der Huffman als von der Lempel-Zip Kodierung profitiert. Die zur Verfügung stehenden Zeichen seien die Buchstaben von A-Z.
- (c) [2 Punkte] Gehen Sie nun umgekehrt vor. Konstruieren Sie einen Datenstrom von mindestens zehn Zeichen, der mehr von Lempel-Ziv als von Huffman profitiert.
- (d) [2 Punkte] Warum wird die theoretisch mögliche Kodierungseffizienz von Huffman Codes meistens nicht erreicht? Liegt dies in der Natur des Huffman Codes oder an den Daten? Begründen Sie Ihr Urteil.
- (e) [4 Punkte] In einem Text kommen sehr viele sehr lange Runs (Läufe) desselben Zeichens vor, es soll aber dennoch keine Lauflängenkodierung verwendet werden. Ist in diesem Fall eine Huffman- oder Lempel-Ziv Kodierung von Vorteil? Begründen Sie Ihr Urteil.

## Aufgabe 2: Kompressionsalgorithmen [13 Punkte]

(a) [2+3+2+3+1+2=13 Punkte] DCT-Transformationen

Die eindimensionale 8-Punkt-DCT ist definiert als

$$S(u) = \frac{C(u)}{2} \sum_{x=0}^7 s(x) \cos\left(\frac{(2x+1)u\pi}{16}\right)$$

mit  $C(0) = 1/\sqrt{2}$  und  $C(u) = 1$  für  $u > 0$ .

- (i) [2 Punkte] Wie muss das Eingangssignal  $s(x)$  lauten, damit die Ausgabe  $S(u) = 0$  erzeugt wird?
  
- (ii) [3 Punkte] Wie muss das Eingangssignal  $s(x)$  lauten, damit die Ausgabe  $S(0) = 1$ ,  $S(u) = 0$  für  $u > 0$  erzeugt wird?
  
- (iii) [2 Punkte] Wie lautet der Basisvektor der DCT für  $u = 1$ ? Die Cosinusterme brauchen nicht vereinfacht zu werden.
  
- (iv) [3 Punkte] Gibt es zwei verschiedene Eingangssignale  $s_1(x)$ ,  $s_2(x)$ , welche die gleiche Ausgabe  $S(u)$  erzeugen? Wenn ja, welche? Wenn nein, warum nicht?
  
- (v) [1 Punkt] Wofür steht der Wert  $S(0)$  eines beliebigen Eingangssignals anschaulich?
  
- (vi) [2 Punkte] Was ist der Unterschied zwischen *Diskretisierung* und *Quantisierung*. Erläutern Sie kurz beide Begriffe.

### **Aufgabe 3: Kommunikationsunterstützung [8 Punkte]**

- (a) [3 Punkte] Nennen Sie die wesentlichen Unterschiede zwischen Netzwerk-kartenadressen in LAN's und Internetadressen?
- (b) [2 Punkte] Erläutern Sie die Funktionsweise des *Routing Information Protocol (RIP)*?
- (c) [3 Punkte] Warum ist Medienskalierung besonders wichtig bei Multicast-Übertragungen?

## Aufgabe 4: Betriebssystemunterstützung [6+6=12 Punkte]

- (a) [6 Punkte] Beschreiben Sie die Scheduling-Verfahren *Earliest deadline first*, *Rate-monotonic* und Scheduling nach *laxity*. Nennen Sie Vorteile und Nachteile dieser Verfahren.
- (b) [6 Punkte] Ein Server zur Übertragung von Video- und Audiodaten nutzt das Non-preemptive Scheduling. Seine Aufgabe ist es, Audio- und Videodaten gleichmäßig an seine Empfänger zu senden. Das Senden eines Stromes (Video oder Audio) geschieht jeweils in einem Prozess. Jeder Prozess soll mindestens einmal pro Sekunde aktiv sein. Durch Beschränkung der CPU-Kapazität kann der Server innerhalb von 6 ms 150 kByte Videodaten bzw. innerhalb von 5 ms 10 kByte an Audiodaten übermitteln. Ein Audiostrom besteht aus durchschnittlich 128 kBit/s, ein Videostrom aus durchschnittlich 2400 kBit/s. Zum Umschalten zwischen zwei Prozessen werden jeweils 2,5 ms benötigt.

Wieviele Ströme kann der Server maximal parallel versenden, wenn zu jedem Videostrom genau ein Audiostrom gesendet werden soll? Erläutern Sie Ihre Rechenschritte, so dass diese leicht nachvollziehbar sind.

## Aufgabe 5: Video-Inhaltsanalyse [6+1+3+2+4+3=19 Punkte]

- (a) [6 Punkte] Eine Anzuchtschale für Bakterien wird einmal pro Stunde digital fotografiert. Die Vergrößerung reicht dabei aus, um einzelne Bakterien eindeutig zu identifizieren. Die Biologen brauchen ein Verfahren, welches gleichzeitig einen Rückschluss darauf zulässt, ob Bakterien entstanden bzw. gestorben sind. Dabei kann angenommen werden, dass sich Bakterien nicht bewegen und diese nach dem Tod vollständig verschwinden. Erschwerend kommt hinzu, dass sich die Färbung des Nährbodens wie auch der Bakterien im Zeitablauf unvorhersehbar verändern können. Erläutern Sie, wie Entstehen und Vergehen detektiert werden kann. Sie können dabei auf ein Verfahren aus der Vorlesung zurückgreifen.
- (b) [1 Punkt] Was ist der optische Fluss in einem Video?
- (c) [3 Punkte] Wann ist es besonders schwer, den optischen Fluss zu bestimmen, wann funktioniert es besonders gut und warum?
- (d) [2 Punkte] Wie geeignet ist der optische Fluss für die Schnitterkennung? Welche Vor- oder Nachteile gibt es?
- (e) [4 Punkte] Zeichnen Sie zwei Bilder, in denen der optische Fluss...
1. ... in genau einem Pixel exakt ermittelt werden kann.
  2. ... in mindestens einem Pixel nur in eine Richtung genau berechnet werden kann.
  3. ... in allen Pixeln Null ist.
  4. ... in einigen Pixeln nicht berechnet werden kann.
- (f) [3 Punkte] Wie und warum wirkt sich die Anzahl der Bits/Pixel auf die Genauigkeit einer Flussberechnung aus?

## Aufgabe 6: Inhaltsanalyse [10+8=18 Punkte]

(a) [10 Punkte] Bildtransformationen

Gegeben seien fünf verschiedene Bildgeometrie-Transformationen durch:

$$(1) \vec{x}' = \vec{x} + \vec{t}$$

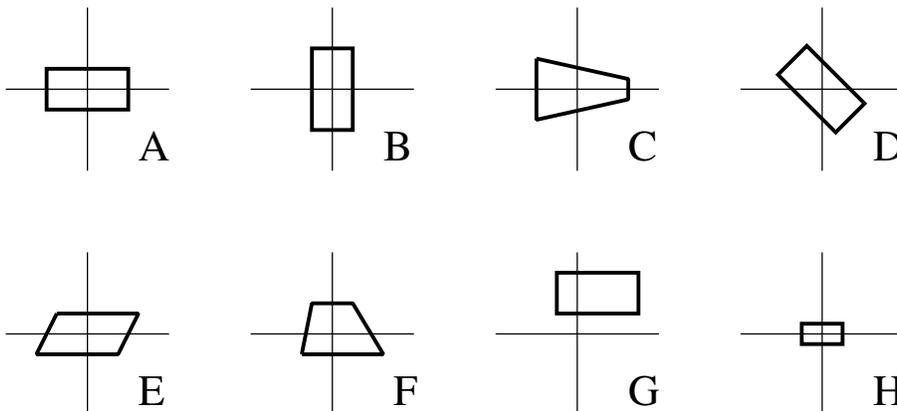
$$(2) \vec{x}' = A\vec{x} + \vec{t}$$

$$(3) \vec{x}' = \frac{A\vec{x} + \vec{t}}{b^T \vec{x} + 1}$$

$$(4) \vec{x}' = \begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix} \vec{x}$$

$$(5) \vec{x}' = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \vec{x}$$

Dabei sind  $\vec{t}, \vec{b}, a_i, \alpha$  und die  $2 \times 2$ -Matrix  $A$  beliebige Parameter. Durch die Transformation wird die Position  $\vec{x}$  auf  $\vec{x}'$  abgebildet.



(i) [8 Punkte] Welche der Bilder B-H können mit den obigen Transformationen jeweils aus Bild A erzeugt werden? Markieren Sie die möglichen Kombinationen in der unten vorbereiteten Tabelle!

(ii) [2 Punkte] Benennen Sie die Transformationen mit ihrem gebräuchlichen Namen.

Transformation	B	C	D	E	F	G	H	Name
1								
2								
3								
4								
5								

(b) [8 Punkte] Bild-Segmentierung

Implementieren Sie den *Region-Growing* Algorithmus in C, Java, oder einem vernünftigen Pseudocode. Benutzen Sie dazu das unten angegebene Code-Fragment. Bild-Ränder brauchen nicht berücksichtigt zu werden. In dem globalen zweidimensionalen Feld `image[][]` steht das Graustufenbild. Die Region-Growing Funktion soll im Feld `region[][]` alle Pixel-Positionen auf `true` setzen, die zur Region gehören. Zum Zeitpunkt des Aufrufs ist `region[][]` mit `false` initialisiert.

```
int image[][];
bool region[][];

void RegionGrowing(
    int x,        // x-Koordinate des Start-Pixels
    int y,        // y-Koordinate des Start-Pixels
    int sigma) // Aequivalenz-Schwelle
{

}
}
```



Codewörter (Datenbits)	Kanalbits
0000 0000	01001000100000
0000 0001	10000100000000
0000 0010	10010000100000
0000 0011	10001000100000
0000 0100	01000100000000
0000 0101	00000100010000
0000 0110	00010000100000
0000 0111	00100100000000
0000 1000	01001001000000
0000 1001	10000001000000
0000 1010	10010001000000

*Hinweis:* Beachten Sie, dass Füllbits zwischen den Datenbits liegen.

(iv) [1 Punkt] Wie gross ist die Kanalbitrate bei einer CD-DA?