

Übung 2 – Digitale Medien

Inhalt

- Umrechnung Binär <-> Dezimal
- LZW-Codierung mit Ausnahmebehandlung,
- Digitalisierung.

Aufgaben

Aufgabe 1: Umrechnung zwischen Binär und Dezimal (2 Punkte)

- a) Konvertieren sie die folgende Zahl (dezimal) ins Binärformat. Berechnen sie dabei zuerst die Vorkommastellen (r1), dann die Nachkommastellen (r2) und geben sie das Ergebnis so an: r1, r2. In der Lösung müssen alle Rechenschritte mit angegeben werden.
3110,71875
- b) Konvertieren sie die folgende Zahl (Binärformat) ins Dezimalsystem. In der Lösung müssen alle Rechenschritte mit angegeben werden.
1001 0110

Aufgabe 2: Codierung nach Lempel-Ziv-Welch (4 Punkte)

Gegeben sei die Codetabelle des ASCII-Zeichensatzes (siehe S. 4).

Neue Einträge können erst ab Index 256 erfolgen.

Weiterhin gegeben sei folgende Nachricht:

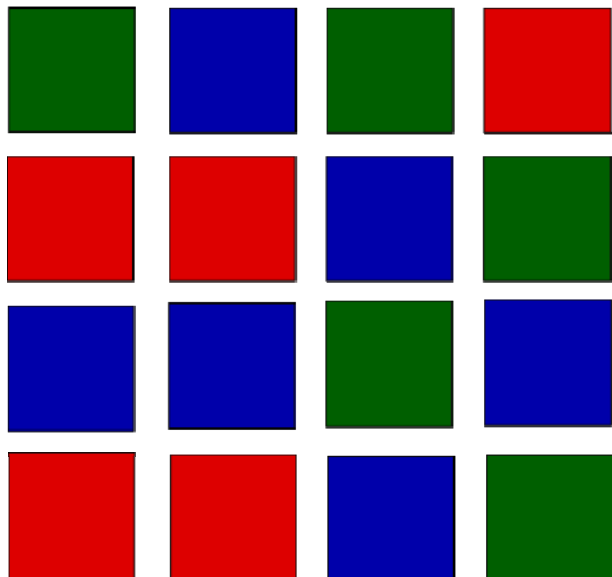
rabarabalelele

- a) Codieren Sie die Nachricht mittels LZW-Codierung. Gehen Sie dabei entsprechend dem in der Vorlesung behandelten Algorithmus vor. Verwenden Sie zur Darstellung der Zwischenschritte des Algorithmus eine Tabelle mit den Spalten: „Lesen“, „Codetabelle schreiben“, „Ausgabe“ und „Puffer füllen“. Das Wörterbuch wird mit der oben angegebenen Codetabelle initialisiert.
- b) Decodieren Sie die Nachricht. Verwenden Sie zur Darstellung der Zwischenschritte des Algorithmus eine Tabelle mit den Spalten: „Lesen“, „Ausgabe“, „Puffer füllen“, „Codetabelle schreiben“ und „Merken“.
- c) Codieren Sie die Nachricht zusätzlich im unkomprimierten ASCII-Format. Wandeln Sie dann beide Codierungsformen (LZW und ASCII) in die jeweiligen Binärdarstellungen. Benutzen Sie für den ASCII-Code 8-Bit Darstellungen für jedes Zeichen (siehe Tabelle im Anhang) und für LZW 9-Bit Darstellungen (da Zahlen > 255). Um wie viel Prozent schrumpft die Nachricht durch Verwendung von LZW?

Aufgabe 3: Codierung eines Bildes mit LZW (3 Punkte)

Folgendes Bild mit einer Auflösung von 4 mal 4 Bildpunkten ist vorgegeben:

	Dezimal	Huffman
Rot (r)	0	01
Blau (b)	1	1
Grün (g)	2	00



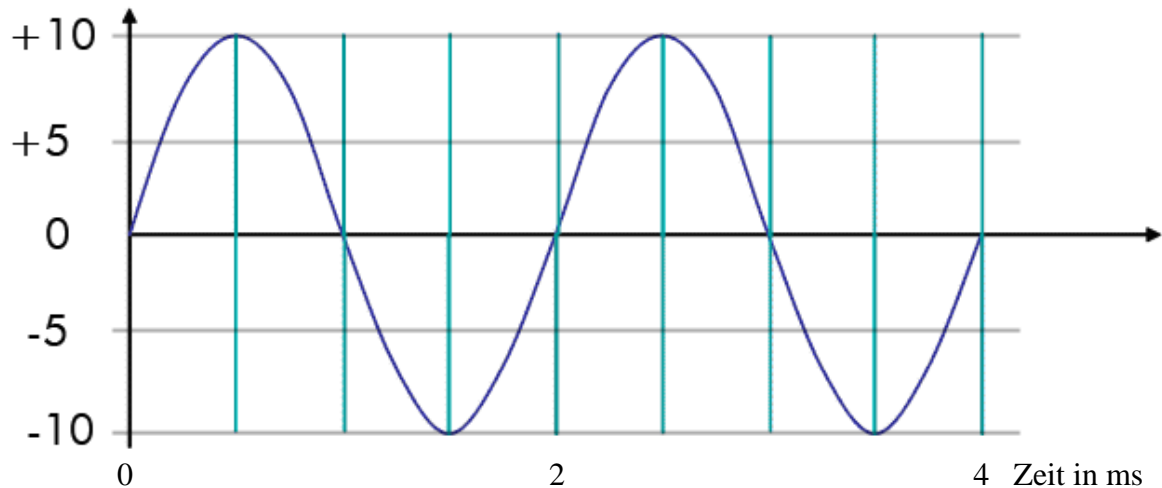
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Die vorhandenen Farben sollen mit der obenstehenden Tabelle nach den Werten die in der Spalte „Dezimal“ angegeben sind codiert werden. Das Bild wird zeilenweise von links nach rechts verarbeitet, d.h. nachdem die vier Pixel in Zeile 1 verarbeitet wurden, wird direkt mit Zeile 2 weitergemacht (Siehe kleinere Grafik).

- a) Codieren Sie das komplette Bild analog zur ersten Aufgabe anhand einer Tabelle mit dem LZW-Algorithmus. Der erste neue Tabelleneintrag ist bei Index 3.
- b) Codieren Sie das Bild dann mit dem vorgegebenen Huffman-Code Pixel für Pixel.
- c) Konvertieren Sie das Ergebnis aus Aufgabe a) ins Binärformat (Für LZW gilt diesmal eine 4-Bit Darstellung) und bestimmen Sie, welches der beiden Codierungsverfahren für dieses spezielle Bild besser geeignet ist (d.h. den kürzeren Code liefert).

Aufgabe 4: Digitalisierung (3 Punkte)

In folgender Abbildung ist ein 4 Millisekunden langer Ausschnitt eines periodischen Signals dargestellt. Die linke Skala bezeichnet die Amplitude des Signals:



- Welche Frequenz in Hertz hat das Signal?
- Das Signal wird mit einer Rate von 0,5 kHz abgetastet. Geben Sie für jedes Sample, das dadurch innerhalb der 4 ms erstellt wird, den Zeitpunkt der Messung in ms und die gemessene Amplitude an. Die Messungen beginnen zum Zeitpunkt 0.
- Die Abtastrate wird jetzt auf 2 kHz erhöht. Geben Sie wiederum die Anzahl der Samples, deren Zeitpunkte und gemessene Amplituden an.
- Mit welcher der beiden Abtastraten kann das Signal wieder rekonstruiert werden? Wie groß muss die Abtastrate mindestens sein (mit Begründung)?

Abgabe

Bitte geben Sie Ihre Lösung als ZIP-Datei bis zum 13.11.09 12:00 Uhr im UniWorx Portal (<http://www.pst.ifi.lmu.de/uniworx>) ab.

Anhang: ASCII – Tabelle (Ausschnitt)

Dezimal	Hex	Binär	Zeichen	Dezimal	Hex	Binär	Zeichen
032	020	00100000	(Leer)	080	050	01010000	P
033	021	00100001	!	081	051	01010001	Q
034	022	00100010	"	082	052	01010010	R
035	023	00100011	#	083	053	01010011	S
036	024	00100100	\$	084	054	01010100	T
037	025	00100101	%	085	055	01010101	U
038	026	00100110	&	086	056	01010110	V
039	027	00100111	'	087	057	01010111	W
040	028	00101000	(088	058	01011000	X
041	029	00101001)	089	059	01011001	Y
042	02A	00101010	*	090	05A	01011010	Z
043	02B	00101011	+	091	05B	01011011	[
044	02C	00101100	,	092	05C	01011100	\
045	02D	00101101	-	093	05D	01011101]
046	02E	00101110	.	094	05E	01011110	^
047	02F	00101111	/	095	05F	01011111	_
048	030	00110000	0	096	060	01100000	`
049	031	00110001	1	097	061	01100001	a
050	032	00110010	2	098	062	01100010	b
051	033	00110011	3	099	063	01100011	c
052	034	00110100	4	100	064	01100100	d
053	035	00110101	5	101	065	01100101	e
054	036	00110110	6	102	066	01100110	f
055	037	00110111	7	103	067	01100111	g
056	038	00111000	8	104	068	01101000	h
057	039	00111001	9	105	069	01101001	i
058	03A	00111010	:	106	06A	01101010	j
059	03B	00111011	;	107	06B	01101011	k
060	03C	00111100	<	108	06C	01101100	l
061	03D	00111101	=	109	06D	01101101	m
062	03E	00111110	>	110	06E	01101110	n
063	03F	00111111	?	111	06F	01101111	o
064	040	01000000	@	112	070	01110000	p
065	041	01000001	A	113	071	01110001	q
066	042	01000010	B	114	072	01110010	r
067	043	01000011	C	115	073	01110011	s
068	044	01000100	D	116	074	01110100	t
069	045	01000101	E	117	075	01110101	u
070	046	01000110	F	118	076	01110110	v
071	047	01000111	G	119	077	01110111	w
072	048	01001000	H	120	078	01111000	x
073	049	01001001	I	121	079	01111001	y
074	04A	01001010	J	122	07A	01111010	z
075	04B	01001011	K	123	07B	01111011	{
076	04C	01001100	L	124	07C	01111100	
077	04D	01001101	M	125	07D	01111101	}
078	04E	01001110	N	126	07E	01111110	~
079	04F	01001111	O	127	07F	01111111	DEL