

Praktikum Aufgabe 3

(Abgabe bis 04. Juni 2007)

12 Punkte

Informatik für den Studiengang Physik
Dr. Herbert Süße, Institut für Informatik
8.5.2007

1 Allgemeines

Wenn Sie das ICE-Textterminal benutzen, so sind folgende Funktionen von Interesse:

Ein- und Ausgabefunktionen

int PutChar(int c)

Es wird ein Zeichen c an das Textfenster ausgegeben.

int GetChar(void)

Es wird die Eingabe eines Zeichens erwartet und der Zeichencode der gedrücketen Taste zurückgegeben.

int GetKey(void)

Es wird getestet, ob eine Taste gedrückt wurde. Ist das der Fall, wird der zugehörige Zeichencode zurückgegeben, wenn keine Taste gedrückt wurde, ist der Rückgabewert 0.

int Printf(const char *format,...)

Ausgabe in das Textfenster. Der Formatstring und die restlichen Parameter sind aufgebaut wie bei der Standardfunktion printf().

int Input(char *p)

Eingabe einer Integerzahl. Der String p wird als Prompt ausgegeben. Bei fehlerhafter Eingabe wird die Anforderung wiederholt.

double InputD(char *p)

Eingabe einer Gleitkommazahl. Der String p wird als Prompt ausgegeben. Bei fehlerhafter Eingabe wird die Anforderung wiederholt.

void InputS(unsigned char *p, unsigned char *s)

Eingabe einer Zeichenkette. Der String p wird als Prompt ausgegeben.

2 Aufgabe a(10 Punkte)

Lesen Sie ein Bildfile ein. Berechnen Sie das Grauwertistogramm. Berechnen Sie aus dem Grauwertistogramm die Entropie des Bildes und geben Sie diesen Wert auf dem Textterminal aus. Berechnen Sie aus dem Bild ein neues Bild nach folgender Vorschrift: Die erste Zeile des Bildes bleibt erhalten. Die folgenden Zeilen enthalten einfach die modulo 256 Differenzen der Zeile zur vorhergehenden Zeile, also $g'(i, j) = g(i, j) - g(i, j - 1) \bmod 256$ für alle i einer Zeile. Damit ist das

Originalbild wieder eindeutig aus dem transformierten Bild rekonstruierbar, Sie müssen dann ebenfalls wieder modulo 256 rechnen. Berechnen Sie nun ebenfalls das Grauwert histogramm und daraus die Entropie des transformierten Bildes. Geben Sie diese aus und vergleichen die Entropien. Berechnen Sie, basierend auf den ausgerechneten Entropien der eindimensionalen Grauwertstatistik, die theoretischen verlustfreien Kompressionsraten des Original- und des transformierten Bildes. Rekonstruieren Sie das Originalbild aus dem transformierten Bild. Visualisieren Sie das Originalbild, das transformierte Bild und das rekonstruierte Bild.

3 Aufgabe b (2 Punkte)

Zusätzlich ist ein Bild (der gleichen Bilddimension wie in 2a) mittels Zufallszahlen zu generieren, benutzen Sie Gaußsches weißes Rauschen, d.h. für jedes Pixel ist unabhängig voneinander eine Zufallsgrauwert zu generieren, der Gauß-verteilt ist. Für dieses Bild ist ebenfalls die Aufgabe 2a) zu realisieren. Vergleichen Sie die Entropieergebnisse und versuchen Sie diese Ergebnisse zu deuten. Blenden Sie mittels ICE-Funktionen stets die Histogramme in die Bilder ein.

Erzeugung von Zufallszahlen und Histogrammen in ICE

void Randomize()

Initialisierung des Zufallszahlengenerators

int Random(int val)

Es wird eine Realisierung einer im Intervall $[0, \text{val}]$ gleichverteilten Zufallsvariablen erzeugt.

double RandomD()

Es wird eine Realisierung einer im Intervall $[0, 1]$ gleichverteilten Zufallsvariablen erzeugt.

double GaussRandom(double sigma)

Es wird eine Realisierung einer normalverteilten Zufallsvariablen mit dem Erwartungswert 0 und der Standardabweichung σ erzeugt

Histogramme sind spezielle Datentypen und sind mit

Hist name1, name2, ..

zu vereinbaren.

Hist HistImg(Image img, int diff)

Es wird ein Grauwert histogramm für das Bild `img` angelegt, wobei für jeden möglichen Grauwert eine Klasse vorgesehen wird. Für das Histogramm werden Bildpunkte aus einem Raster mit dem Abstand `diff` ausgewertet.

int Hist::Vis(val v, Image img)

In das Bild `img` wird das Histogramm eingezeichnet. Bitte mit z.B. `name1.Vis(v, img);` aufrufen