

Einführung in Matlab

Übung 9

Aufgabe 1: Benutzen Sie die einfache 2D-HWT und 2D-IHWT zur **Kantendetektion** an einem Bild Ihrer Wahl, analog zur Vorlesung. Bei einem **Farbbild** empfiehlt es sich, dies zuerst **zu einem Grauwertbild aus double-Werten zu machen**. (Denken Sie daran, **geradzahlige Spalten- und Zeilenanzahlen** zu verwenden.)

Aufgabe 2: Benutzen Sie die einfache 2D-HWT und 2D-IHWT zur **Datenkompression** an einem Bild Ihrer Wahl (auch hier Farbbild zu Grauwertbild mit double-Werten). Machen Sie dazu folgendes:

- (a) Schreiben Sie eine Funktion `[e,ind]=energie(x)`, welche zu einem Signal `x` sowohl die **kumulierte Energie** `e` als auch den Indexvektor `ind` der Sortierung liefert, so dass gilt `v=abs(x(ind))` (siehe Vorlesung).
Hinweis: Damit dies auch für Matrizen (Bilder) funktioniert, sollte der erste Befehl `x=x(:)` lauten. Dies macht aus einer Matrix einen langen Spaltenvektor, indem alle Spalten der Matrix untereinander gehängt werden.
- (b) **Zunächst per Augenmaß und Hand analog zur Vorlesung:** Geben Sie sich einen Prozentsatz `p` der Gesamtenergie vor, den Sie erhalten möchten und schauen Sie sich die kumulierten Energien des Originals und der HW-Transformierten an. Identifizieren Sie die Mindestanzahl der Daten, die benötigt werden um `p` zu erreichen, setzen Sie die restlichen auf Null, und machen Sie dann beim komprimierten transformierten Bild wieder die Rücktransformation. Vergleichen Sie das so komprimierte Bild (`imagesc`) mit dem Original.
- (c) Nun **automatisch auf Null setzen**: Schreiben Sie eine Funktion `[x,q]=aufNull(x,p)`, die in (b) die Schritte “Identifizieren” und “auf Null setzen” für Sie erledigt, und zusätzlich den Prozentsatz $q = \frac{\text{Anzahl Nichtnull-Einträge}}{\text{Gesamte Pixelzahl}}$ des so komprimierten Bildes zurückgibt.
- (d) **Alles automatisch**: Schreiben Sie eine Funktion `[X,q]=komprimiere(X,p)`, die zuerst (c) macht, dann beim komprimierten transformierten Bild die Rücktransformation, und diese dann auf den ursprünglichen Intensitätsbereich zurückskaliert als `uint8`-Bild zurückgibt (analog zur Vorlesung; dann kann auch wieder mit `image` geplottet werden).

Aufgabe 3: Implementieren Sie jeweils eine Funktion `y=HWT1DK(x,k)` und entsprechend `IHWT1DK`, `HWT2DK`, `IHWT2DK` für die **K-fache 1D/2D Haar-Wavelet-(Rück)Transformation** und testen Sie diese. Dazu lassen sich die schon in der Vorlesung geschriebenen Funktionen `HWT1D`, `IHWT1D`, `HWT2D`, `IHWT2D` geeignet verwenden. Mit der zwei- oder dreifachen Trafo lässt sich oftmals noch besser komprimieren als mit der einfachen. Testen Sie das auch wie in Aufgabe 2.