DCT - Übung

Felix Rohrer

>

Aufgabe 1: Diskrete Cosinus Transformation

Mit Hilfe der DCT (Diskrete Cosinus Transformation) kann ein Vektor $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1})^T$ [1] wie folgt auf den Vektor $\mathbf{y} = (y_0, y_1, \ldots, y_{N-1})^T$ abgebildet werden:

$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

wobei die Matrix $\mathbf{C}$ definiert ist durch

$$C_{ik} = c_i \cos\left(\frac{\pi i\,(2k+1)}{2N}\right) \quad 0 \le i, k \le N-1$$

mit $c_0 = \frac{1}{\sqrt{N}}$, $c_i = \sqrt{\frac{2}{N}}$ für $i = 1, 2, \ldots, N-1$.

Definieren Sie für beliebiges $N \in \mathbb{N}$ die Matrix $\mathbf{C}$ und zeigen Sie, dass die Inverse von $\mathbf{C}$ gleich der Transponierten ist

$$\mathbf{C}^{-1} = \mathbf{C}^T$$

Dazu müssen Sie zeigen, dass gilt:

$$\mathbf{C}\mathbf{C}^T = \mathbf{C}^T\mathbf{C} = 1$$

Kontrollieren Sie für $N = 8$ und $N = 100$!

Mit Matlab gelöst:

```
function [m] = dct(n) ;
% ----------------
% erzeugt die Matrix der Grösse n x n
% ----------------
m = zeros(n,n);
for j = 1:n
    m(1,j) = 1/sqrt(n);
end;

for k = 2:n
    for j = 1:n
        m(k,j) = sqrt(2/n) * cos((k-1)*pi*(2*j-1)/(2*n));
    end;
end;
%
```

N = 8

```
>> C = dct(8)

C =

    0.3536    0.3536    0.3536    0.3536    0.3536    0.3536    0.3536    0.3536
    0.4904    0.4157    0.2778    0.0975   -0.0975   -0.2778   -0.4157   -0.4904
    0.4619    0.1913   -0.1913   -0.4619   -0.4619   -0.1913    0.1913    0.4619
    0.4157   -0.0975   -0.4904   -0.2778    0.2778    0.4904    0.0975   -0.4157
    0.3536   -0.3536   -0.3536    0.3536    0.3536   -0.3536   -0.3536    0.3536
    0.2778   -0.4904    0.0975    0.4157   -0.4157   -0.0975    0.4904   -0.2778
    0.1913   -0.4619    0.4619   -0.1913   -0.1913    0.4619   -0.4619    0.1913
    0.0975   -0.2778    0.4157   -0.4904    0.4904   -0.4157    0.2778   -0.0975

>> CT = transpose(C)

CT =

    0.3536    0.4904    0.4619    0.4157    0.3536    0.2778    0.1913    0.0975
    0.3536    0.4157    0.1913   -0.0975   -0.3536   -0.4904   -0.4619   -0.2778
    0.3536    0.2778   -0.1913   -0.4904   -0.3536    0.0975    0.4619    0.4157
    0.3536    0.0975   -0.4619   -0.2778    0.3536    0.4157   -0.1913   -0.4904
    0.3536   -0.0975   -0.4619    0.2778    0.3536   -0.4157   -0.1913    0.4904
    0.3536   -0.2778   -0.1913    0.4904   -0.3536   -0.0975    0.4619   -0.4157
    0.3536   -0.4157    0.1913    0.0975   -0.3536    0.4904   -0.4619    0.2778
    0.3536   -0.4904    0.4619   -0.4157    0.3536   -0.2778    0.1913   -0.0975

>> C*CT

ans =

    1.0000    0.0000   -0.0000    0.0000    0.0000    0.0000   -0.0000   -0.0000
    0.0000    1.0000    0.0000   -0.0000    0.0000   -0.0000    0.0000    0.0000
   -0.0000    0.0000    1.0000    0.0000   -0.0000    0.0000    0.0000         0
    0.0000   -0.0000    0.0000    1.0000    0.0000    0.0000   -0.0000    0.0000
    0.0000    0.0000   -0.0000    0.0000    1.0000    0.0000   -0.0000   -0.0000
    0.0000   -0.0000    0.0000    0.0000    0.0000    1.0000    0.0000   -0.0000
   -0.0000    0.0000    0.0000   -0.0000   -0.0000    0.0000    1.0000    0.0000
   -0.0000    0.0000         0    0.0000   -0.0000   -0.0000    0.0000    1.0000
```

N = 100
Wie man sofort sieht, ist die Matrix 100x100 etwas gros - nichts destotrotz stimmt sie ;-)

```
>> C = dct(100);
>> CT = transpose(C);
>> C*CT

ans =

  Columns 1 through 9

    1.0000   -0.0000   -0.0000   -0.0000   -0.0000
   -0.0000    1.0000    0.0000    0.0000    0.0000
   -0.0000    0.0000    1.0000    0.0000   -0.0000
   -0.0000    0.0000    0.0000    1.0000   -0.0000
   -0.0000    0.0000   -0.0000   -0.0000    1.0000
    0.0000    0.0000    0.0000   -0.0000   -0.0000
   -0.0000   -0.0000    0.0000   -0.0000   -0.0000
   -0.0000    0.0000    0.0000   -0.0000    0.0000
    0.0000   -0.0000   -0.0000    0.0000   -0.0000
    0.0000   -0.0000    0.0000    0.0000   -0.0000
   -0.0000    0.0000   -0.0000    0.0000    0.0000
```

Aufgabe 2

Wenn Sie die Aufgabe 1 verstanden haben, können Sie sofort sagen, wie man den Vektor x aus dem Vektor y berechnet! Wie sieht die entsprechende Gleichung in möglichst einfacher Form aus (Matrixform!).

Antwort

Da die Matrix Orthogonal ist, ist sie invertierbar und ihre Transponierte ist gleichzeitig ihre Inverse

$$x = C^T \cdot y$$

Aufgabe 3

(1) Erzeugen Sie ein Graustufenbild mit 8×8 Pixel, in welches Sie einen Kreis mit einem Radius R von 2 Pixel zeichnen. Dabei soll nur die Randlinie des Kreises schwarz sein, der Rest des Bildes weiss. Erzeugen Sie ein Bild der Matrix (mit *matrixplot(..)*).

(2) Führen Sie die Kosinus Transformation durch und komprimieren Sie das Bild, indem Sie nur 90, 70, 50 oder 30 Prozent aller Koeffizienten berücksichtigen.

(3) Führen Sie die Rücktransformation durch. Machen Sie die entsprechenden Matrizen mit *matrixplot(..)* sichtbar.

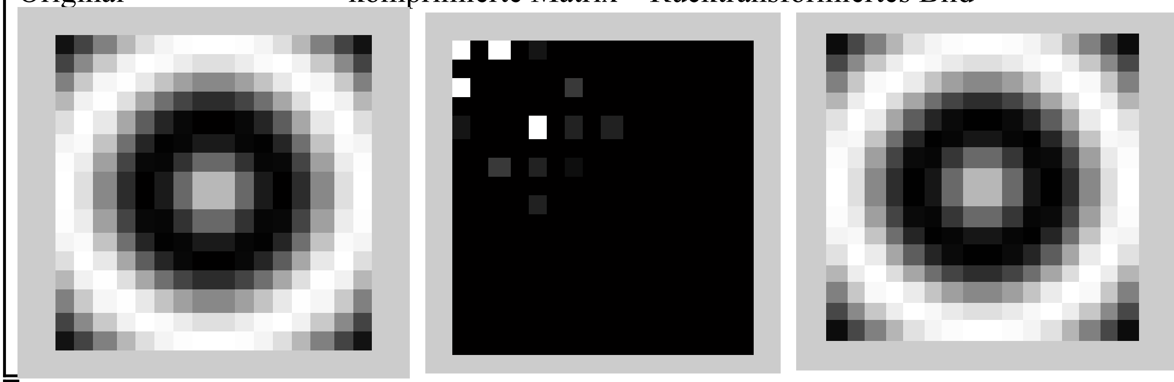
Antwort

```
function f = drawcircle(n);
% Zeichnet einen Kreis
f = zeros(2*n,2*n);
for i = 1:n,
    for k = 1:n,
        f(n-i+1,n-k+1) = 0.5 + 0.5 * cos(sqrt(i*i + k*k) / n*2*pi);
        f(n+i,n-k+1)   = 0.5 + 0.5 * cos(sqrt(i*i + k*k) / n*2*pi);
        f(n-i+1,n+k)   = 0.5 + 0.5 * cos(sqrt(i*i + k*k) / n*2*pi);
        f(n+i,n+k)     = 0.5 + 0.5 * cos(sqrt(i*i + k*k) / n*2*pi);
    end;
end;
```

```
function [m] = mydct(n) ;
% DCT
m = zeros(2*n,2*n);
for j = 1:2*n,
    m(1,j) = 1 / sqrt(2*n);
end;
for k = 2:2*n,
    for j = 1:2*n,
        m(k,j) = sqrt(1/n) * cos((k-1)*pi*(2*j-1)/(4*n));
    end;
end;
```

```
>> img = drawcircle(8);
>> imshow(img);
>> c = mydct(8);
>> ftr = c * img * c';
>> r = abs(ftr) < 0.05;
>> ftrcomp = (1 - r).*ftr;
>> fcomp = c' * ftrcomp * c;
>> imshow(ftrcomp)
>> imshow(fcomp)
```

Original komprimierte Matrix Rücktransformiertes Bild



>

Aufgabe 4

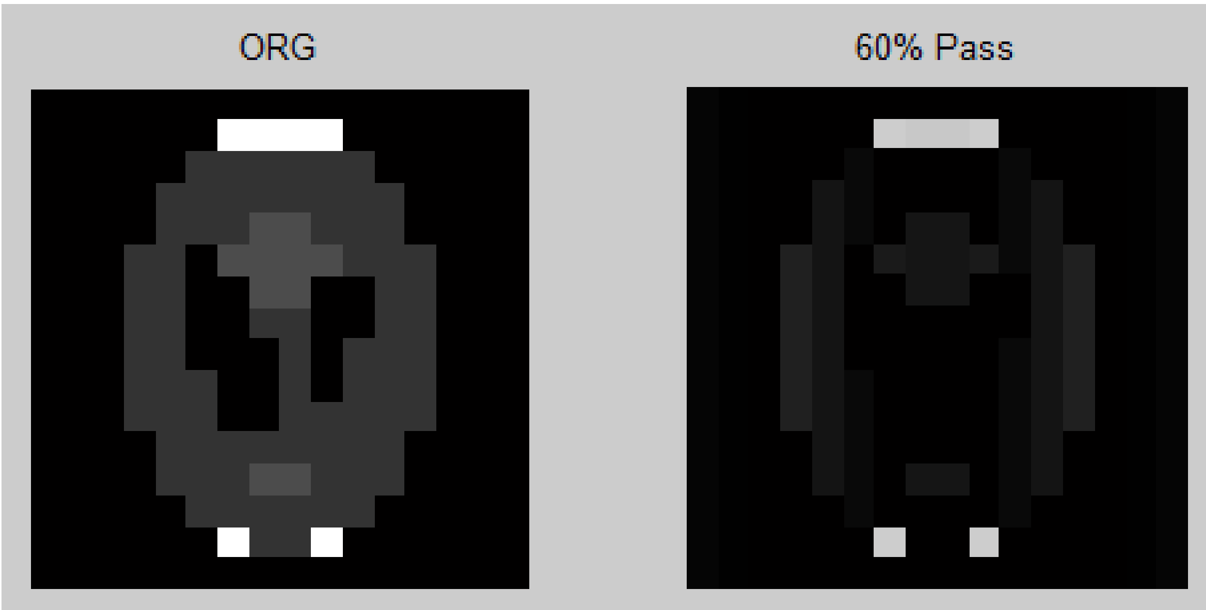
(1) Erzeugen Sie ein frei gewähltes Bild als Graustufen-Matrix der Grösse 16×16.

(2) Führen Sie die Kosinus Transformation durch, und unterdrücken Sie einige der hohen Frequenzen.

(3) Führen Sie die Rücktransformation durch. Zeichnen Sie die so entstehenden Bilder mit *matrixplot(..)*

Antwort

```
>> clear
>> org = phantom(16);
>> new = dct2(org);
>> new(abs(new) > 0.6 * max(new(:))) = 0;
>> new2 = idct2(new);
>> subplot(1,2,1), imshow(org), title('ORG')
>> subplot(1,2,2), imshow(new2), title('60% Pass')
```



>

► Aufgabe 5