

Lecture_4_code_snippet

March 16, 2017

1 Python example:

The DCT Type 2 is defined as

$$T_{n,k} = \frac{1}{\sqrt{N}} \cdot \cos\left(\frac{\pi}{N} \cdot (n + 0.5) \cdot k\right)$$

for $n, k = 0, \dots, N-1$.

Take a 4x4 block of an image, for instance a part of a somewhat dark and even background:

```
In [18]: import numpy as np
         x = 0.3*np.ones((4,4))
         x
```

```
Out[18]: array([[ 0.3,  0.3,  0.3,  0.3],
                [ 0.3,  0.3,  0.3,  0.3],
                [ 0.3,  0.3,  0.3,  0.3],
                [ 0.3,  0.3,  0.3,  0.3]])
```

Now transform this block x . First compute the transform matrix T , which in this example is a so-called DCT type 2. First we obtain the transform matrix T by applying the DCT to the identity matrix I (norm='ortho' includes the square root term):

```
In [19]: import scipy.fftpack
         I = np.eye(4)
         I
```

```
Out[19]: array([[ 1.,  0.,  0.,  0.],
                [ 0.,  1.,  0.,  0.],
                [ 0.,  0.,  1.,  0.],
                [ 0.,  0.,  0.,  1.]])
```

```
In [20]: T = scipy.fftpack.dct(I, norm='ortho')
         T
```

```
Out[20]: array([[ 0.5,  0.65328148,  0.5,  0.27059805],
                [ 0.5,  0.27059805, -0.5, -0.65328148],
                [ 0.5, -0.27059805, -0.5,  0.65328148],
                [ 0.5, -0.65328148,  0.5, -0.27059805]])
```

Here we can see that each column represents a filter. For instance our filter impulse response (the coefficients) for the lowest subband, (including flipping it up-down) is,

```
In [21]: h0=np.flipud(T[:,0])
         h0
```

```
Out[21]: array([ 0.5,  0.5,  0.5,  0.5])
```

Now we can transform our block x to obtain our 2-dimensional subband values y ,

```
In [22]: y=np.dot(np.dot(np.transpose(T),x),T)
         y
```

```
Out[22]: array([[ 1.2,  0. ,  0. ,  0. ],
                 [ 0. ,  0. ,  0. ,  0. ],
                 [ 0. ,  0. ,  0. ,  0. ],
                 [ 0. ,  0. ,  0. ,  0. ]])
```

Observe:

We have only **1 large value**, the 1.2, in the upper, left hand corner. This corresponds to filter $h_0(n)$ horizontally and vertically. The rest of the coefficients are zero. For natural images, the DCT usually has this effect, that the large values appear in the upper, left hand corner, and the rest of the values are much smaller. This property is also called "**energy compaction**", and is used to reduce the bit rate, by taking advantage of redundancies in our signal, the image block. For instance, many consecutive zeros can be efficiently and easily encoded using the run-length coding (an escape code, followed by the value and the number of repetitions).