frameencfile

February 9, 2017

1 Program - Frameencfile

Program to capture a video from the default camera (0), compute the 2D DCT on the Y component, quantize the lowest 3 coefficients of each DCT block and save them as 2 bit values in files framedim.txt, y00enc.bin, y01enc.bin, and y10enc.bin - Gerald Schuller, Dec. 2015

- Import relevant modules:

```
In [1]: import cv2
        import numpy as np
        import scipy.fftpack as sft
        #import our file functions:
        from writereadbits import *
        import blockdct
```

```
writereadbits.py
```

- Defining the variables:

```
In [2]: cap = cv2.VideoCapture(0)
        #Get size of frame:
        [retval, frame] = cap.read()
        [r,c,d]=frame.shape
        print(r,c)
        #Store dimensions in info file:
        np.savetxt('framedim.txt', [r,c])
        print "Record to compressed files with key 'q', show until key 'q' is pressed "
```

```
(480, 640)
Record to compressed files with key 'q', show until key 'q' is pressed
```

- only kep the 3 lowest frequencies coefficients of the 8x8 DCT,

```
In [3]: N=8

        while(True):
```

```
# Capture frame-by-frame
[retval, frame] = cap.read()
Y=(0.114*frame[:,:,0]+0.587*frame[:,:,1]+0.299*frame[:,:,2])/255;

key=cv2.waitKey(1) & 0xFF
if key== ord('c'):
    print "store frame encoded in files framedim.txt, y00enc.bin, y01enc.bin, and y1

    #compute magnitude of 2D DCT of blocks of 8x8 pixels of the Y component
    X=blockdct.dct8x8(Y)

    #print X[0,0]
    #Quantize:
    #print('Quantisieren')
    #Ausprobieren vom Bereich:
    #DC: 0..5
    #AC: -0.5..+0.5

    #Number of bits per pixel
    bits=2

    #resulting quantization step size for 2^bits steps:
    #Stufen fuer unterschiedliche Ortsfrequenzen:
    #DC Indices mit range 0...5:
    quantstufeDC=5.0/(2**bits-1)

    #Alle DC indices (anfangen mit Position 0 und dann jeder N'te Koeffizient:
    #Kleinsten Indexwert addieren um den ganzen range des coders zu nutzen:
    indices00=np.round(X[0::N,0::N]/quantstufeDC)-2

    #reshape into 1-D array:
    indices00=np.reshape(indices00,(1,-1))
    #print indices00.shape

    #convert to code string:
    codestring00=data2codestring(indices00[0,:])

    #write  to binary file
    writebinaryfile('y00enc.bin', codestring00)

    #Zwei AC Koeffizienten, mit range 0.5-(-0.5)
    #DCT Koeffizienten der Position (0,1):
    quantstufeAC=1.0/(2**bits-1)
    indices01=np.round(X[0::N,1::N]/quantstufeAC)

    #Reshape:
    indices01=np.reshape(indices01,(1,-1))
```

```
        #Store with 2 bits each value:
        #convert to code string:
        codestring01=data2codestring(indices01[0,:])

        #write to binary file
        writebinaryfile('y01enc.bin', codestring01)

        #DCT Koeffizienten der Position (1,0):
        indices10=np.round(X[1::N,0::N]/quantstufeAC)
        indices10=np.reshape(indices10,(1,-1))

        #convert to code string:
        codestring10=data2codestring(indices10[0,:])

        #write to binary file
        writebinaryfile('y10enc.bin', codestring10)


    cv2.putText(Y,"Frame Compression Demo,", (20,50), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (25
    cv2.putText(Y,"Press 'c' to take picture, 'q' to quit,", (20,80), cv2.FONT_HERSHEY_S
    cv2.imshow('Original Video, Y Komponente, 8bits/Pixel',Y)
    #Keep window open until key 'q' is pressed:
    if  key == ord('q'):
        break
```

```
store frame encoded in files framedim.txt, y00enc.bin, y01enc.bin, and y10enc.bin
```

- When everything done, release the capture

```
In [4]: cap.release()
        cv2.destroyAllWindows()
```