videorecdct0idctdisp

March 8, 2017

1 Videorecdct0idctdisp

Program to capture a video from the default camera (0), compute the 2D DCT on the Green component, take the magnitude (phase) and display it live on the screen and then low pass filter it and finally inverse transform it.

-Gerald Schuller, Nov. 2014

- **Import relevant modules:**

```
In [1]: import cv2
        import numpy as np
        import scipy.fftpack as sft
```

- **Define the variables:**

```
In [2]: cap = cv2.VideoCapture(0)
        #Get size of frame:
        [retval, frame] = cap.read()
        [r,c,d]=frame.shape
        print(r,c)
```

```
(480, 640)
```

- **Mask to set to zero the 3/4 highest frequencies, only kep the 1/4 lowest frequencies in each direction for the DCT, because of the DCT no longer symmetric about the center:**

- **For rows:**

```
In [3]: Mr = np.ones((r,1))
        Mr[(r/4.0):r,0] = np.zeros((3.0/4.0*r))
        #For columns:
        Mc=np.ones((1, c))
        Mc[0, (c/4.0):c] = np.zeros((3.0/4.0*c));
        #Together:
        M = np.dot(Mr,Mc)
```

- ** Start looping over captured frames, analyse it and compute the 2D-DCT for the green component of those frames and then apply inverse DCT to display the coded frame simultaneously:**
- **After applying the 2D-DCT removing the High frequencies i.e., 7/8th of the frequencies and considering only low frequency componenets**
- **Once done press 'q' to quit all windows:**

```
In [4]: while(True):
            # Capture frame-by-frame
            [retval, frame] = cap.read()
            cv2.imshow('Original Video, Gruen Komponente',frame[:,:,1])

            #compute magnitude of 2D DCT of green component
            #with suitable normalization for the display,
            #with norm='ortho' for "energy conservation" in the subbands and for
            #invertibiltity without factor:
            X=sft.dct(frame[:,:,1]/255.0,axis=1,norm='ortho')
            X=sft.dct(X,axis=0,norm='ortho')
            #Set to zero the 7/8 highest spatial frequencies in each direction:
            X=X*M
            frame=np.abs(X)

            # Display the resulting frame
            cv2.imshow('2D-DCT mit Null Setzen der hoechsten Ortsfrequenzen',frame)
            #Inverse 2D DCT:
            X=sft.idct(X,axis=1,norm='ortho')
            x=sft.idct(X,axis=0,norm='ortho')
            cv2.imshow('Inverse 2D DCT ohne die hoechsten Ortsfrequenzen', x)

            #Keep window open until key 'q' is pressed:
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
```

- **When everything is done then release the capture:**

```
In [5]: cap.release()
        cv2.destroyAllWindows()
```