Digital Image Processing Notes

MATLAB Plots of LPF and HPF

Convolution Thm:

Assuming Linear Spatially invariant system a system is characterized by it's impulse response. For the cases of images which we denote with a 2dim grey scale matrix, we define the impulse response to be a function of the number of rows and columns in the matrix h(n1,n2)

The output of the LSI is the convolution of the input and the impulse response

y(n1,n2)=x(n1,n2)**h(n1,n2)

To find an impulse response of a system input a delta function and the output is the impulse response. The equivalent of a 2d delta function for images is the matrix with a 1 at 0,0 and zeros everywhere else.

0 0 0
0 1 0
0 0 0

```
\begin{pmatrix}
0 &0  &0 \\
0 &1  &0 \\
0 &0  &0
\end{pmatrix}
```

Edge Effects Convolution

A FFT is a way to compute the DFT. There are butterfly algorithms to calculate the FFT fast; this is different than what the GPU can do.

Find DFT of (-1)^(n1+n2)

5. $x(n_1, n_2)$ is defined as $x(n_1, n_2) = (-1)^{n_1+n_2}$ when $0 \leq n_1, n_2 \leq 2$ and zero elsewhere. Denote by $X(k_1, k_2)$, where $0 \leq k_1, k_2 \leq 2$, the DFT of $x(n_1, n_2)$. What is the value of $X(1, 2)$

Defn of FFT/DFT:

X(k1,k2)=sum(sum(x(n1,n2)exp(-j2pi/N1*k1)exp(-j2pi/N2*k2)))

X(k1,k2)=sum(n1 from 0 to N1-1)(sum(n2 from 0 to N2-1)(x(n1,n2)exp(-j2pi/N1*n1*k1)exp(-j2pi/N2*n2*k2)))

X(k1,k2)=
x(0,0)exp(-j2pi/N1*0*k1)exp(-j2pi/N2*0*k2)+
x(0,1)exp(-j2pi/N1*0*k1)exp(-j2pi/N2*1*k2)+
x(0,2)exp(-j2pi/N1*0*k1)exp(-j2pi/N2*2*k2)+
x(1,0)exp(-j2pi/N1*1*k1)exp(-j2pi/N2*0*k2)+
x(1,1)exp(-j2pi/N1*1*k1)exp(-j2pi/N2*1*k2)+
x(1,2)exp(-j2pi/N1*1*k1)exp(-j2pi/N2*2*k2)+
x(2,0)exp(-j2pi/N1*2*k1)exp(-j2pi/N2*0*k2)+
x(2,1)exp(-j2pi/N1*2*k1)exp(-j2pi/N2*1*k2)+
x(2,2)exp(-j2pi/N1*2*k1)exp(-j2pi/N2*2*k2)

Set the exp terms with 0 to 1

X(k1,k2)=
x(0,0)+
x(0,1)exp(-j2pi/N2*1*k2)+
x(0,2)exp(-j2pi/N2*2*k2)+
x(1,0)exp(-j2pi/N1*1*k1)+
x(1,1)exp(-j2pi/N1*1*k1)exp(-j2pi/N2*1*k2)+
x(1,2)exp(-j2pi/N1*1*k1)exp(-j2pi/N2*2*k2)+
x(2,0)exp(-j2pi/N1*2*k1)+
x(2,1)exp(-j2pi/N1*2*k1)exp(-j2pi/N2*1*k2)+
x(2,2)exp(-j2pi/N1*2*k1)exp(-j2pi/N2*2*k2)

X(1,2)=
x(0,0)+
x(0,1)exp(-j2pi/N2*2)+
x(0,2)exp(-j2pi/N2*4)+
x(1,0)exp(-j2pi/N1)+
x(1,1)exp(-j2pi/N1)exp(-j2pi/N2*2)+
x(1,2)exp(-j2pi/N1)exp(-j2pi/N2*4)+
x(2,0)exp(-j2pi/N1*2)+
x(2,1)exp(-j2pi/N1*2)exp(-j2pi/N2*2)+
x(2,2)exp(-j2pi/N1*2)exp(-j2pi/N2*4)

x(0,0)=(-1)^0=1
x(1,0)=x(0,1)=(-1)^1=-1
x(1,2)=x(2,1)=(-1)^3=-1
x(2,2)=(-1)^4=1

X(1,2)=
1
-exp(-j2pi/N2*2)
+exp(-j2pi/N2*4)
-exp(-j2pi/N1)
+exp(-j2pi/N1)exp(-j2pi/N2*2)
-exp(-j2pi/N1)exp(-j2pi/N2*4)
+exp(-j2pi/N1*2)
-exp(-j2pi/N1*2)exp(-j2pi/N2*2)
+exp(-j2pi/N1*2)exp(-j2pi/N2*4)

N2=N1=3

X(1,2)=
1
-exp(-j2pi/3*2)
+exp(-j2pi/3*4)
-exp(-j2pi/3)
+exp(-j2pi/3)exp(-j2pi/3*2)
-exp(-j2pi/3)exp(-j2pi/3*4)
+exp(-j2pi/3*2)
-exp(-j2pi/3*2)exp(-j2pi/3*2)
+exp(-j2pi/3*2)exp(-j2pi/3*4)

combine constants in exp

X(1,2)=
1
-exp(-j4pi/3)
+exp(-j8pi/3)
-exp(-j2pi/3)
+exp(-j2pi/3)exp(-j4pi/3)
-exp(-j2pi/3)exp(-j8pi/3)
+exp(-j4pi/3)
-exp(-j4pi/3)exp(-j4pi/3)
+exp(-j4pi/3)exp(-j8pi/3)

Combine exp terms into single exp term

X(1,2)=
1
-exp(-j4pi/3)

+exp(-j8pi/3)
-exp(-j2pi/3)
+exp(-j6pi/3)
-exp(-j10pi/3)
+exp(-j4pi/3)
-exp(-j8pi/3)
+exp(-j12pi/3)

X(1,2)=
1
-exp(-j4pi/3)
+exp(-j8pi/3)
-exp(-j2pi/3)
+exp(-j6pi/3)
-exp(-j10pi/3)
+exp(-j4pi/3)
-exp(-j8pi/3)
+exp(-j12pi/3)

e^npi = cos(npi) + isin(npi)

X(1,2)=1-exp(-j4pi/3)+exp(-j8pi/3)-exp(-j2pi/3)+exp(-j2pi)-exp(-j10pi/3)+exp(-j4pi/3)-exp(-j8pi/3)+exp(-j4pi)

1
-(cos(-4pi/3)+jsin(-4pi/3))
+(cos(-8pi/3)+jsin(-8pi/3))
-(cos(-2pi/3)+jsin(-2pi/3))
+(cos(-2pi)+jsin(-2pi))
-(cos(-10pi/3)+jsin(-10pi/3))
+(cos(-4pi/3)+jsin(-4pi/3))
-(cos(-8pi/3)+jsin(-8pi/3))
+(cos(-4pi)+jsin(-4pi))

distribute - signs

1-cos(-4pi/3)-jsin(-4pi/3)+cos(-8pi/3)+jsin(-8pi/3)-cos(-2pi/3)-jsin(-2pi/3)+cos(-2pi)+jsin(-2pi)-cos(-10pi/3)-jsin(-10pi/3)+cos(-4pi/3)+jsin(-4pi/3)-cos(-8pi/3)-jsin(-8pi/3)+cos(-4pi)+jsin(-4pi)

combine cos and sin terms

1-cos(-4pi/3)+cos(-8pi/3)-cos(-2pi/3)+cos(-2pi)-cos(-10pi/3)+cos(-4pi/3)-cos(-8pi/3)+cos(-4pi)-jsin(-4pi/3)+jsin(-8pi/3)-jsin(-2pi/3)+jsin(-2pi)-jsin(-10pi/3)+jsin(-4pi/3)-jsin(-8pi/3)+jsin(-4pi)

1
-cos(-4pi/3)

+cos(-8pi/3)
-cos(-2pi/3)
+cos(-2pi)
-cos(-10pi/3)
+cos(-4pi/3)
-cos(-8pi/3)
+cos(-4pi)
-jsin(-4pi/3)
+jsin(-8pi/3)
-jsin(-2pi/3)
+jsin(-2pi)
-jsin(-10pi/3)
+jsin(-4pi/3)
-jsin(-8pi/3)
+jsin(-4pi)

cos(-4pi/3) = -cos(-pi/3)=cos(pi/3)
cos(-8pi/3) = cos(-2pi/3)=cos(2pi/3)
cos(-2pi/3) = cos(2pi/3)
cos(-2pi)= 1
cos(-10pi/3)=cos(-4pi/3)=-cos(pi/3)
cos(-4pi)=1

sin(-4pi/3)= sin(pi/3)
sin(-8pi/3)=sin(-2pi/3)=-sin(2pi/3)
sin(-2pi/3)= - sin(2pi/3)
sin(-2pi)=0
sin(-10pi/3)=sin(-4pi/3)= sin(pi/3)
sin(-4pi)=0

1
-cos(pi/3)
+cos(2pi/3)
-cos(2pi/3)
+1
+cos(pi/3)
+cos(pi/3)
-cos(2pi/3)
+1
-jsin(pi/3)
-jsin(2pi/3)
+jsin(2pi/3)
-jsin(pi/3)
+jsin(pi/3)
+jsin(2pi/3)

cancel terms

1+1+cos(pi/3)-cos(2pi/3)+1-jsin(pi/3)+jsin(2pi/3)

3+1/2+1/2+1=4

DFT example:

X(k1,k2)=sum(sum(x(n1,n2)exp(-j2pi/N1*k1)exp(-j2pi/N2*k2)))

arithmetic sum 16*17/2=136

Convolutoin Examples:i9;;

Upsampling Images

Downsampling Images

Gaussian Pyramid:
Laplace Pyramid:

week 4 questions

1 point

1. Check all the applications where motion estimation can be employed to improve the results:

- [x] Object tracking
- [x] Human-computer interaction
- [ ] Still image inpainting
- [x] Video compression
- [ ] Segmentation of a single image

---

1 point

2. We want to increase the frame rate of a video sequence by inserting a new frame between every two existing consecutive frames. Denote by y the new frame formed via linear interpolation of motion vectors between frames x_{t-1} and x_t in the original video. Assuming that a circular object is centered at pixel (i,j) in x_{t-1} and at pixel (p,q) in x_t, where will it be centered in y?

- ( ) $(p+i, q+j)$
- ( ) $((p+i)/2, (q+j)/2)$
- ( ) $(p-i, q-j)$
- ( ) $((p-i)/2, (q-j)/2)$

Test w 2 points; (2,3), (4,5). To interpolate half way point (3,4); (x+y)/2 to get 3 and 4 from ponts above

Motion Estimation:

MSE:

3. Calculate the Mean Square Error (MSE) between the two given image blocks (enter your answer to at least one decimal point):

1	1	2	2
1	1	2	2
2	2	3	4
2	2	5	6

Block 1

2	2	1	1
2	2	2	2
2	2	6	4
2	2	5	3

Block 2

```
%calculate MSE between 2 matrices
X=[1 1 2 2; 1 1 2 2; 2 2 3 4; 2 2 5 6]
Y = [2 2 1 1; 2 2 2 2; 2 2 6 4; 2 2 5 3]
distxy=abs(X-Y).^2
msexy=sum(distxy(:)/numel(X))


%msexy =

%    1.5000
```

1 point

4. Assume that we want to perform block matching for the image block [Math Processing Error] given below. Which of the following image blocks is a better match in the Mean Absolute Error (MAE) sense?

10	20	10	10
20	40	10	10
30	40	20	20
50	60	20	20

Block x

- ○

10	20	10	10
20	40	10	10
20	20	30	40
20	20	50	60

- ○

20	30	20	20
30	50	20	20
40	50	30	30
60	70	30	30

- ○

10	20	30	40
20	40	50	60
10	10	20	20
10	10	20	20

- ○

1	2	1	1
2	4	1	1
3	4	2	2
5	6	2	2

```
%MAE calculation example
mref=[10 20 10 10; 20 40 10 10; 30 40 20 20; 50 60 20 20]
m1=[10 20 10 10; 20 40 10 10; 20 20 30 40; 20 20 50 60]
m2=[20 30 20 20; 30 50 20 20; 40 50 30 30; 60 70 30 30]
m3=[10 20 30 40; 20 40 50 60; 10 10 20 20; 10 10 20 20]
```

```
m4=[1 2 1 1; 2 4 1 1; 3 4 2 2; 5 6 2 2]

dist1 = abs(mref-m1)
dist2=abs(mref-m2)
dist3 = abs(mref-m3)
dist4=abs(mref-m4)

mae1=sum(dist1(:))
mae2=sum(dist2(:))
mae3=sum(dist3(:))
mae4=sum(dist4(:))
```

```
mae1 =

   200


mae2 =

   160


mae3 =

   280


mae4 =

   351
```

Smallest is MAE2@160

---

1 point

5. (True or False) Sub-pixel motion estimation is used in applications where a faster and hence less accurate estimation of motion is needed.

- (selected) True
- False

This is different than what he said in lecture on Pixel Subsampling. False is correct answer.

1 point

6. Refer to the RGB cube shown in the video lecture for this problem. Color magenta can be obtained by 1:1 mixing red and blue; yellow can be obtained by 1:1 mixing red and green; cyan can be obtained by 1:1 mixing blue and green. If magenta, yellow, and cyan are mixed at 1:1:1 proportion, what is the resulting color?

- ○ red
- ○ green
- ○ blue
- ○ white
- ◉ black

1 point

7. (True or False) Intensity in HSI color space is exactly the same as the Y-channel in YCbCr color space, as both represent the "brightness" of an image.

- ◉ True
- ○ False

1 point

8. In the next two problems you will perform block matching motion estimation between two consecutive video frames. Follow the instructions below to complete this problem.

(1) Download the two video frames from frame_1 and frame_2. The frames/images are of height 288 and width 352.

(2) Load the frame with file name "frame_1.jpg" into a 288×352 MATLAB array using function "imread", and then convert the array type from 8-bit integer to real number using function "double" or "cast" (note that the range of intensity values after conversion is between 0 and 255). Denote by I_1 the converted MATLAB array. Repeat this step for the frame with file name "frame_2.jpg" and denote the resulting MATLAB array by I_2. In this problem, I_2 corresponds to the current frame, and I_1 corresponds to the previous frame (i.e., the reference frame).

(3) Consider the 32×32 target block in I_2 that has its upper-left corner at $(65, 81)$ and lower-right corner at $(96, 112)$. Note this is MATLAB coordinate convention, i.e., the first number between the parenthesis is the row index extending from 1 to 288 and the second number is the column index extending from 1 to 352. This target block is therefore a 32×32 sub-array of I_2.

(4) Denote the target block by B_{target}. Motion estimation via block matching searches for the 32×32 sub-array of I_1 that is "most similar" to B_{target}. Recall in the video lectures we have introduced various forms of matching criteria, e.g., correlation coefficient, mean-squared-error (MSE), mean-absolute-error (MAE), etc.

In this problem, we use MAE as the matching criterion. Given two blocks B_1 and B_2 both of size $M \times N$, the MAE is defined as $MAE(B_1, B_2) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} |B_1(i,j) - B_2(i,j)|$. To find the block in I_1 that is most similar to B_{target} in the MAE sense, you will need to scan through all the 32×32 blocks in I_1, compute the MAE between each of these blocks and B_{target}, and find the one that yields the smallest value of MAE.

Note in practice motion search is only performed over a certain region of the reference frame, but for the sake of simplicity, we perform motion search over the entire reference frame I_1 in this problem and the next. When you find the matched block in I_1, enter the sum of the x and y coordinates of the upper-left corner of the matched block in MATLAB convention. For example, if the matched block has the upper-left corner located at $(10, 20)$ then you must enter 30.

Enter answer here

1 point

9. In the previous question, what was the corresponding MAE value (up to two decimal points)?

Enter answer here

Week 4: Optical Flow

From Forsythe/Ponce

http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_video/py_lucas_kanade/py_lucas_kanade.html

Feature Based Motion Estimation: SIFT/SURF
Kalman Filtering to predict motion of object

RGB -> YCrCB conversion:
rgb2gray supports the generation of C code using MATLAB® Coder™.
Algorithms
rgb2gray converts RGB values to grayscale values by forming a weighted sum of the R, G, and B components:

0.2989 * R + 0.5870 * G + 0.1140 * B
These are the same weights used by the rgb2ntsc function to compute the Y component.

convert RGB to HSV, is the grey scale in HSV same as Y?

RANSAC:
RANSAC used to
1) estimate the fundamental matrix of stereo vision
2) commonality between 2 sets of points for object detection
3) register sequential video frames for image stabilization(form of motion estimation)

What is a fundamental matrix?

Stereo Vision:

Disparity Calculation: