Lab 7 – image processimg – Octave version

Resources

Running OCTAVE
- Boot into Linux (SuSE).
- from the start menu run "Terminal"
- in the terminal type `octave`

Pager When displaying a long matrix, Octave automatically pipes the output to a pager program `less`. If you see ":" at the bottom line, Use `q` to exit `less`.

Plots With a plot, use right mouse button to zoom; press "u" to undo zoom.

Octave image processing toolbox and local additions

x=imread('filename.ext') read a {gif,bmp} file; grayscale file will give an MxN matrix, color file will give an MxNx3 (RGB) matrix.

xgray=rgb2gray(xrgb) convert a n RGB (MxNx3) image to grayscale (MxN); use it only if your image is not grayscale.

C6_IMAGE(x) or C6_IMAGE(r,g,b) show an image

colormap(mapname) set a colormap for showing; maps: hsv, gray, hot, cool, copper, pink, jet ...

colorbar display a colorbar showing value-to-color mapping

FREQZ2, filter2, fft2, ifft2 2-D analogues for respective 1-D functions

IMFFT, IMIFFT 2-D fft pair with zero frequency shifted into center $((-\pi,\ \pi >$ displays nicer than $(0,\ 2\pi >)$

m=IMMASK(r,type,image) makes a 'lowpass' or 'highpass' type filter mask in the frequency domain, sized for the given image; $(0 < r < min(size(image))/2$ is a cutoff radius)

FFT_ED(img) interactive spectrum editor

MED_FILT2(img[,M,N]) 2-D median filter with MxN mask (default: 3x3)

Look for test images in `/home/CYPS2011_C6/`. Available grayscale files are: LILIE.BMP, MIROW.BMP, ZACHOD.BMP, Zima.bmp. (LANIER.TIF is a colour image for exercise 8)

Experiments

1. display and sketch a 2-D FFT of a 16x16 image consisting of:
   (a) horizontal stripes of 1 pixel width; of 2 pixels width
   (b) vertical stripes of 1 pixel width; of 2 pixels width
   (c) diagonal stripes of 1 pixel width `d1=eye(4); d1=[d1 d1 d1 d1]; d1=[d1;d1;d1;d1];` of 2 pixels width `d2=d1+shift(d1,-1)`
   (d) checkerboard with 1x1 and 2x2 fields

   hint: `C6_IMAGE(x); colormap(gray); X=imfft(x); figure; C6_IMAGE(abs(X)); colormap(jet);`
2. Read a photo image `xo=imread('filename.ext'); C6_IMAGE(x); colormap(gray); size(x)`
3. Display an FFT of image x (hint: use `abs()` or `log10(abs())`, set colormap to "jet")

4. experiment with linear filters in "time" domain:
   `h1=ones(3,3)`
   and `h2=-ones(3,3); h2(2,2)=8;`

   With each filter:

   (a) filter your image
   (b) check frequency characteristics (hint: `FREQZ2(h); colormap(jet);`);
   (c) describe filter type and visible effects with your own words

5. experiment with linear filters in frequency domain
   Show image FFT and compute maximum mask radius:
   `X=imfft(x); C6_IMAGE(log10(abs(X))); r=min(size(X))/2;` then compute filters:

   (a) `h1mask=IMMASK(0.3*r,'highpass',X);C6_IMAGE(h1mask);`
   (b) `h2mask` - same but lowpass, radius `0.7*r`

   With each filter:

   (a) filter your image by multiplying image FFT with mask; display filtered image
   (b) describe filter type and visible effects with your own words

6. experiment with special linear filters

   (a) (choose one) `lapx=ones(3,1)*[-1 2 -1]; lapy=lapx';`
   (b) (choose one) `sobh=[-1 0 1]'*[1 2 1]; sobv=sobh';`

   Try to describe filter characteristics and effects on your image.

optional Removing periodic distortion
Distort image: `[xst]=IMSTRIPE(x [,angle,density])`. Then test and describe efficiency of:

   (a) lowpass linear filter of order 3 (`h1` from previous experiments)
   (b) median filter (a nonlinear one: `MED_FILT2(xst)`;
   (c) spectrum editor (`FFTED`) – remove (set to zero) spectrum fragments distored most;

7. Linear and nonlinear filtering of additive and impulse-type interference
   Distort your image with:

   (a) "salt and pepper" noise `xsalt=IMNOISE_SP(x);`
   (b) gaussian noise (choose the variance) `xgauss=IMNOISE_GAUSS(x, 0, 0.01);`

   Describe effects of using following filters with each distortion type:

   (a) lowpass linear filter of order 3 (`h1` from previous experiments)
   (b) median filter (a nonlinear one: `MED_FILT2(xst)`;

8. Space images processing

   (a) read a 3-channel image from earth observing satellite
       `[r g b]=TIFFREAD('/home/CYPS2011_C6/lanier.tif');` Display the image `C6_IMAGE(r,g,b);`.
   (b) compute histograms for each channel `imhist(r);`. Why is the picture "dim"?
   (c) try to rescale each channel `r1=r*a+c;` to obtain full 0.0-1.0 scale.
   (d) try statistical rescaling - to cover $\mu - 2\sigma$ till $\mu + 2\sigma$ with display scale of 0.0-1.0. You'll need to cut too small and too big values: `r1=min(1,max(0,r*a+c));`

   optional Use NDVI (normalized digital vegetation index) to display vegetation areas.
   NDVI is defined as $NDVI = \frac{IR-R}{IR+R}$. Infrared channel (IR) is our "red", true red channel (R) is our "green".