**Elie Weintraub and Hillel Weintraub ECE418 - Digital Video Spring 2013**

**Index of Files for Superresolution Project:**

**I. Functions used to Generate Low Resolution Observation Images**

**A. Down Sampling**

*1) D.m*

* Calling syntax: [b] = D( a,m,n )

[b] = D( a,m )

* Description: Implements downsampling matrix D in the image domain. Decimates image *a* by a factor of m\*n. If n is left out decimates by m.
* Returns: Decimated image b

*2) Dtrans.m*

* Calling syntax: [ out ] = Dtrans( b,m,n )

[ out ] = Dtrans( b,m )

* Description: Implements multiplication by DT (transpose of the downsampling matrix) in the image domain, effectively expanding the matrix b by replicating pixel values in m\*n blocks. If n is left out, it expands in sqrt(m)\*sqrt(m) blocks.
* Returns: out, the image resulting from this operation

**B. Blurring**

*3) circconv2.m*

* Calling syntax: [ b ] = mycircconv2( a,h )
* Description: Circularly convolve a with h. The returned result is same size as a. The center of h is treated as the origin. Used in *H.m* to convolve image with blurring kernel.
* Returns: The image resulting from the circular convolution of a and h.

*4) H.m*

* Calling syntax: [ b ] = H( a,h )
* Description: Implements blurring matrix H in the image domain.
* Returns: The blurred image, b, resulting from the circular convolution of a and h.

*5) Htrans.m*

* Calling syntax: [ b ] = Htrans( a,h )
* Description: Implements multiplication by HT (transpose of the blurring matrix) in the image domain.
* Returns: The image, b, resulting from this operation.

**C. Geometric Warping**

*6) Warp.m*

* Calling syntax: [ im2 ] = Warp(im, shift\_x,shift\_y )
* Description: Translates an image by (shift\_x, shift\_y) pixels using bilinear interpolation for sub-pixel shifts.
* Returns: The image, im2, resulting from the translation.

*7) WarpT.m*

* Calling syntax: [ im2 ] = WarpT(im, shift\_x,shift\_y )
* Description: Performs WT (the transpose of the warp matrix) effectively spreading out the intensity of a pixel proportional to the interpolation coefficients.
* Returns: The image, im2, resulting from this operation.

**D. Image Generation Functions**

*8) imGen.m*

* Calling syntax: [y] = imGen(x,shft\_x,shft\_y,h,noise\_var,dwn\_m,dwn\_n)

[y] = imGen(x,shft\_x,shft\_y,h,noise\_var,dwn)

* Description: Generates a blurred, warped, noisy, downsampled observation image, y, from an original image, x. Conceptually performs DHWx+n, where D is a down sample matrix, H is a blur matrix, W is a warp matrix, and n is a noise vector.
* Returns: The resulting observation image, y.

*9) genObsSeq.m*

Calling syntax: [ y ] = GenObsSeq(x,shft\_x,shft\_y,h,noise\_var,dwn\_m,dwn\_n)

[ y ] = GenObsSeq(x,shft\_x,shft\_y,h,noise\_var,dwn)

* Description: Generates an observation sequence of blurred, warped, noisy, and downsampled observation images, y, from an original image, x. Conceptually performs DHrWrx+nr , where D is a down sample matrix, Hr is a blur matrix for the rth observation, Wr is a warp matrix for the rth observation, and nr is a noise vector for the rth observation.(Note: shft\_x, shft\_y are vectors of length r. h is a cell of blur kernels. noise\_var, dwn\_m, and dwn\_n are scalars.)
* Returns: The resulting sequence of observation images, y.

**E. Wrapper Functions**

*10) DHW.m*

* Calling syntax: [ y ] = DHW(x,shft\_x,shft\_y,h,dwn\_m,dwn\_n)

[ y ] = DHW(x,shft\_x,shft\_y,h,dwn)

* Description: Generates a blurred, warped, and downsampled image from an original image,x. Conceptually performs DHWx, where D is a down sample matrix, H is a blur matrix, W is a warp matrix. Performs this by calling D.m ,H.m, and Warp.m.
* Returns: The resulting image, y.

*11) DHWtrans.m*

* Calling syntax: [ y ] = DHWtrans(x,shft\_x,shft\_y,h,dwn\_m,dwn\_n)

[ y ] = DHWtrans(x,shft\_x,shft\_y,h,dwn)

* Description: Conceptually performs (DHW) T{x}, where D is a down sample matrix, H is a blur matrix, and W is a warp matrix. Performs this by calling Dtrans.m , Htrans.m, and WarpT.m.
* Returns: The resulting image, y.

**II. Functions used to Generate the Superresolved Images**

*12) GNC.m*

* Calling syntax: [X,Xinit,itr\_data]=GNC(Y,prior,MAX\_ITR,shft\_x,shft\_y,h,noise\_var,dwn\_m,dwn\_n)

[X,Xinit,itr\_data]=GNC(Y,prior,MAX\_ITR,shft\_x,shft\_y,h,noise\_var,dwn)

* Description: Implementation of GNC a deterministic annealing algorithm used here to find the MAP estimate of the superresolved image.
* Returns: The superresolved image, X; the initial estimate, Xinit; and itr\_data, a multi-dimensional array containing the estimation of the image after each iteration.

*13) grad\_n.m*

* Calling syntax: [grd\_n]=grad\_n(xhat,y,prior,gamma, shft\_x,shft\_y,h,noise\_var,dwn\_m,dwn\_n)

[grd\_n]=grad\_n(xhat,y,prior,gamma, shft\_x,shft\_y,h,noise\_var,dwn)

* Description: Computes the gradient of the cost function used in the GNC algorithm to find the MAP estimate of the superresolved image. For the prior either a DAMRF or a GMRF can be used.
* Returns: The computed gradient.

**III. Functions used for Analysis of Results**

*14) ISNR.m*

* Calling syntax: [ isnr ] = ISNR(Xinit,itrData )
* Description: Takes the original image, the initial guess, and the estimates of each iteration and returns the ISNR (improvement in SNR) from initial guess to each iteration. Optionally plots the ISNR as a function of iteration number.
* Returns: The computed ISNR.

*15) PSNR.m*

* Calling syntax: [psnr, mse] = PSNR( im1,im2, peakval)
* Description: Takes the original image, im1, and the image, im2, and computes the MSE and PSNR of im2.
* Returns: The computed PSNR and MSE.

**IV. Scripts used for Displaying Results**

*16) demo.m*

A script used to demonstrate the achieved results. A sequence of low resolution images are generated on which the superresolution algorithm is performed. Results are displayed and some statistical analysis is performed.

*17) colordemo.m*

A script used to demonstrate the achieved results when the problem is extended to that of color images. Sequences of low resolution images are generated for each of the RGB channels. The superresolution algorithm is then performed on each channel and the results are combined. Results are displayed and some statistical analysis is performed.