

16.581/16.484 Computer Vision

Assignment #1

Spring 2016

The purpose of this assignment is to acquaint you with the input, output, manipulation, and display of images. The purpose is to learn how to

1. read image files
2. write image files
3. reduce the size of an image by 4, and
4. reduce the size of the image information.
5. display the original and reduced images.

There two parts to this assignment. You should:

Part 1:

1. select one of the images you are provided with. Read the README file to get the image sizes. Use **xv** to display the image.
2. Use some statements of code fragment sample.c, as a starting point for a program that reads in an image of some known size. (the code is not a running code, but is given only to illustrate how to use the read() and write() system functions, to read and write binary images)
3. write a program to output an image that contains only 2 quadrants of your input image, that is the upper-left quadrant, and the bottom-left quadrant of your input image, and use **xv** to print the resulting image (that is, save new image as a .ps file and then send the postscript file to the printer, or save it electronically, i.e. in an image format that you desire, to include it in your electronic report).
4. Turn in the following:
 - a- your source code that reads in the image, cuts out the lower-right quadrant, and writes out the resulting new image
 - b- original input and output picture images
 - c- written report

Part 2:

1. use the code fragment I provided you in class (file sample.c, again, it is a non-running c-code, given only to illustrate the usage of the system commands, read() and write() binary images) as a starting point for a program that reads in a binary image of some known size.
2. write a program to sample the image every other pixel, e.g., keeping the even while discarding the odd pixels, in either horizontal or vertical direction. That is, if your original image is 256x256 pixels, you will then have a 128x128 pixels image. Do not hard-code the file names of the input or output images. Display the new image using **xv**.
3. submit a report explaining what happened to the new image output that you got. Does it look good? if not why?
4. Turn in the following
 - a- your source code that reads in the image, and samples every other pixel.
 - b- original and reduced picture images
 - c- written report