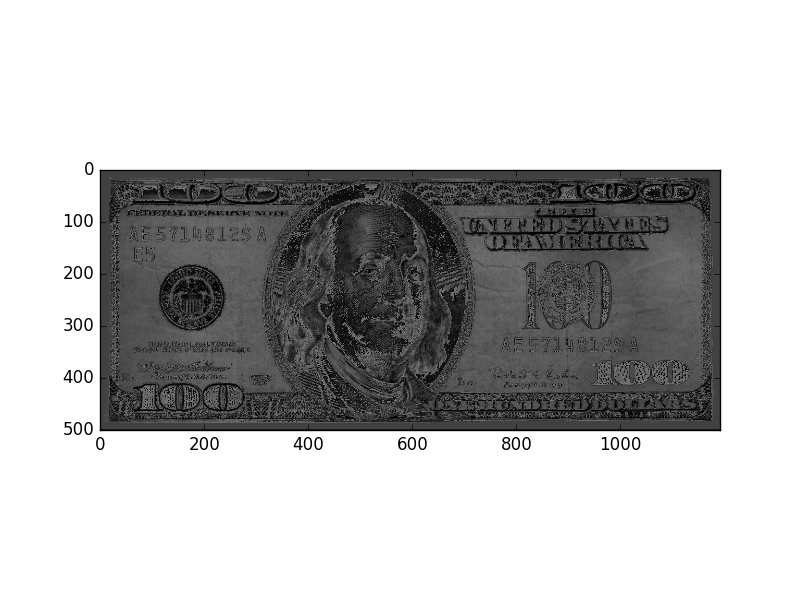
Resample

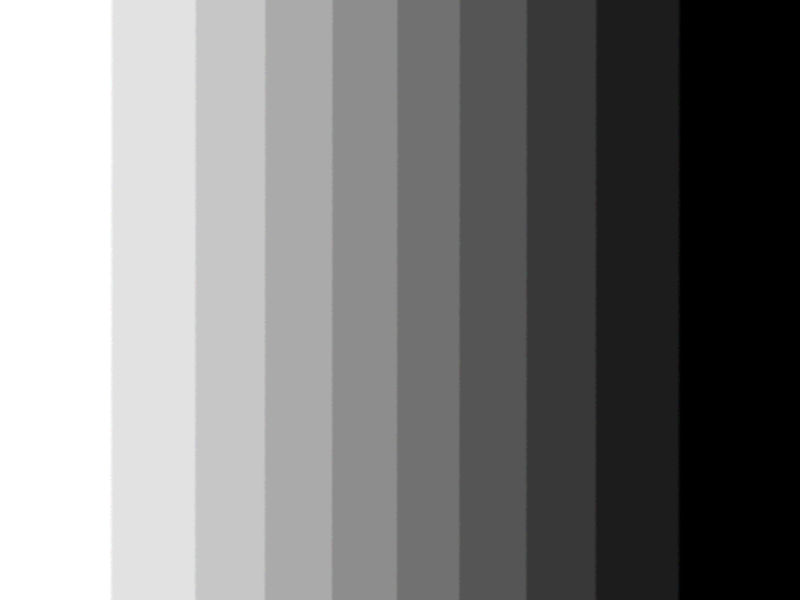
Resample size from 2 to 32(from left to right)



Bitplane



Gray-scale chart



Code for Q1

from skimage.viewer import ImageViewer as IV

import skimage.io as io

import numpy as np

import matplotlib.pyplot as plt

# code folows the idea of last lecture, which get a smaller img from original one and fill pixels in the skipped rows

# and colums from its previous row and col to get the same size picture which however has less information in it.

resample\_size = 32 # change this to different size

c = io.imread('chronometer.tif')

c\_smaller = np.ndarray(shape = (len(range(0,c.shape[0],resample\_size)),len(c[1,:][0:c.shape[1]:resample\_size])), dtype = "uint8")

for i in range(0,c.shape[0],resample\_size):

c\_smaller[i/resample\_size,:] = c[i,:][0:c.shape[1]:resample\_size]

c\_resize = np.ndarray(shape = (c\_smaller.shape[0]\*resample\_size,c\_smaller.shape[1]\*resample\_size), dtype = "uint8")

for i in range(0,c\_smaller.shape[0]):

for j in range(0,c\_smaller.shape[1]):

for h in range(i\*resample\_size,(i+1)\*resample\_size):

for k in range(j\*resample\_size,(j+1)\*resample\_size):

c\_resize[h,k] = c\_smaller[i,j]

v = IV(c\_resize)

v.show()

Code for Q2

d = io.imread("dollar.tif")

plt.imshow((d&(~(1<<0))),vmin = 0, vmax =255, cmap= plt.cm.gray)

plt.imshow((d&(~(1<<1))),vmin = 0, vmax =255, cmap= plt.cm.gray)

plt.imshow((d&(~(1<<2))),vmin = 0, vmax =255, cmap= plt.cm.gray)

plt.imshow((d&(~(1<<3))),vmin = 0, vmax =255, cmap= plt.cm.gray)

plt.imshow((d&(~(1<<4))),vmin = 0, vmax =255, cmap= plt.cm.gray)

plt.imshow((d&(~(1<<5))),vmin = 0, vmax =255, cmap= plt.cm.gray)

plt.imshow((d&(~(1<<6))),vmin = 0, vmax =255, cmap= plt.cm.gray)

plt.imshow((d&(~(1<<7))),vmin = 0, vmax =255, cmap= plt.cm.gray)

Code for Q3

a = np.empty(500)

a[0:100] = 0

a[100:200] = 50

a[200:300] = 100

a[300:400] = 200

a[400:500] = 255

a = a/255.0

b = np.ones(500)

mat = np.outer(b,a)

viewer= IV(mat)

viewer.show()