Assignment 3 COMP 423

Instructions

- Posted Wed. March 26, 2008. Hardcopy due **in class** on Wed. April 9, 2008.
- Assignment will be marked out of 20. Late penalty is 2 points per day (e-copy is due by midnight).
- Submit a hardcopy of answers to the questions below.

  You do not need to submit a hard copy of the source code this time. Instead, use `handin` to submit source code. Include a README file that explains the various programs/data you are handing in.

  Complete instructions on how to use handin are given at `http://www.cs.mcgill.ca/socsinfo/handin/`
- *You may use whatever programming language you like for this assignment.* However, we suggest using a scripting language such as Matlab or Python. We have provided starter code for these (`imageread.m` and `imageread.py`). If you need help, please see me or Rob.
- Be neat and organized. Points will be taken off for excessive sloppiness.

Introduction

A color image consists of an image triplet, namely a red, green, and blue component image. So we can think of color images as matrices `I[HEIGHT][WIDTH][3]`. Color images typically can be compressed very well since, at any pixel, if the intensity value is large (or small) in the red channel then it also tends to be large (or small) in the blue and green channels.

Traditional video formats for broadcasting (e.g. NTSC) typically do not encode RGB images separately, but rather encode transformed images, where each RGB value has been transformed to a *luminance* and two *chrominance* channels. Roughly speaking, luminance corresponds to brightness and the two chrominance channels indicate the "color", which is related to the differences of intensities in the R,G,B images.

In this assignment, you will encode and decode two different color images: `Mikeintree.jpg` and `rockface.jpg`.

Question 1 (lossless - 10 points)

For each of the two images, suppose we were to transform each I_{RGB} pixel as follows:

$$\begin{bmatrix} C_L \\ C_{RG} \\ C_{BY} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix}$$

where the subscripts L, RG, BY stand for "luminance" (brightness), "red vs. green", and "blue vs. yellow" (yellow = red + green), respectively.

1. Compute the frequencies of values in each I_R, I_G, I_B, and estimate the entropy per pixel for each RGB component - i.e. $H(I_R)$, $H(I_G)$, $H(I_B)$. Note that if the intensities were uniformly random, then the entropy would be 8 bits per pixel. You should find that the entropy is less than this. If there are any differences between the two images (`Mikeintree` vs `rockface`), *briefly* explain why – say, half a page.

   In Matlab, you can computer frequencies using `hist` – see starter code. For info on the python histrogram command, see:
   `http://matplotlib.sourceforge.net/matplotlib.pyplot.html#-hist`

2. Compute the frequencies of values in each C_L, C_{RG}, C_{BY} channel. Estimate the entropy of each *i.e.* $H(C_L)$, $H(C_{RG})$, $H(C_{BY})$. Compare the values to each other, and to those of (1.). If there are any differences between the two images, *briefly* explain why.

3. Compute the horizontal differences of values

   $$I_{RGB}(j_1, j_2 + m) - I_{RGB}(j_1, j_2)$$

   for interpixel distances $m = 1, 2, 3$ and for each RGB channel. Estimate the entropy of these differences for each m and each image. i.e. compute 18 entropies (2 images $\times$ 3 channels (RGB) $\times$ 3 m values). Explain your findings.

4. Same as in (3.), but now compute the horizontal differences of values of C_L, C_{RG}, C_{BY}. Again, compute 18 entropies (2 images $\times$ 3 channels $\times$ 3 m values). Compare to (3) and explain why/if there are systematic differences.

NOTE: $I_R, I_G, I_B, C_L, C_{RG}, C_{BY}$ could have a large range of values, so you should *not* print out the frequencies in numerical form (e.g. as an enormous table). Instead, you should provide histogram plots. For example, use Matlab or MS Excel (or Open Office equivalent).

Question 2 (lossy - 10 points)

In Question 1, the rows of the transform matrix were orthogonal but not of unit length. Here we will use an orthonormal matrix, namely

$$\begin{bmatrix} \tilde{C}_L \\ \tilde{C}_{RG} \\ \tilde{C}_{BY} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix}$$

and then use the same quantization width Δ for each of $\tilde{C}_L,\ \tilde{C}_{RG},\ \tilde{C}_{BY}$.

1. Quantize these $\tilde{C}_{L,RG,BY}$ values using a midtread quantizer for two values of Δ, namely $\Delta = 1, 10$. The encoder will encode the levels

   $$l_{L,RG,BY} = round(\frac{\tilde{C}_{L,RG,BY})}{\Delta}) \ .$$

   For each Δ and for each component $\tilde{C}_L, \tilde{C}_{RG}, \tilde{C}_{BY}$, compute frequencies of the levels l and estimate the entropies $H(\tilde{C}_L),\ H(\tilde{C}_{RG}),\ H(\tilde{C}_{BY})$ based on these frequencies.

   As in Question 1, you should not provide the frequency values in tabular form. Instead, give a histogram.

2. Using the levels from (1.), decode the image by apply the inverse transform to obtain an estimate of I_{RGB} at each pixel. Note that the inverse transform is the transpose of the above matrix.

   You'll need to round off each value in order to get integer intensities. Furthermore, if any I_{RGB} value is less than 0 or greater than 255, then you'll need to clip to 0 or 255, respectively, to get an 8 bit image.

   Provide the decoder's reconstructed RGB image, one for each Δ.

   `Mikeintree-Delta1.jpg`, `Mikeintree-Delta10.jpg`, `rockface-Delta1.jpg`, `rockface-Delta10.jpg`.

3. For each image and for each Δ, compute the mean squared pixelwise error in each R, B, G component between the original image and the decoded image.

4. Briefly discuss your findings. In particular, your answer should consider how the pixelwise error and the estimated entropy vary with Δ.