Assignment 1

Due September 20, 2017

This assignment contains six problems. The last one is optional. The first five are graded using a single grade as a group (Assignment 1). The optional problem is graded separately with another letter grade. To avoid “punishing” you for trying the optional problem, I will count it towards your final grade only if it improves your overall grade from Assignment 1. Example: say you get a B for the first five. If you get an A– for the optional one, I will count it because it improves upon B. However, if you get a B– for the optional problem, I will not count it unless you tell me otherwise.

Each problem is supposed to teach you something interesting if you think about what the result means. They require only basic probability skills and understanding of the concepts we introduced during the first two lectures. I will go through the problems and their solutions after the day the assignment is due and we can discuss.

A.1 LSB embedding in two LSBs

Assume that the cover is an 8-bit grayscale image with pixel values in the set $\{0, \ldots, 255\}$.
Consider two different embedding schemes: LSBR2 and LSBM2.

LSBR2 embeds two bits into each pixel by replacing the two least LSBs (the 8th and the 7th bit) with message bits:

> *Emb*: Two bits are embedded into each pixel by replacing its two least significant bits with two message bits. For example, if the cover pixel value is $c_i = 14 = (00001110)_2$ and we want to embed message bit pairs 00, 01, 10, 11, we change c_i to 12, 13, 14, 15, respectively. If $c_i = 32 = (00100000)_2$, we embed the same bit pairs by changing c_i to 32, 33, 34, and 35, etc.
>
> *Ext*: Two bits are extracted as the least two LSBs from the pixels to form the message.

Assuming that the message is a random bit-stream of maximal length (2× number of pixels) and the pixel values are uniformly distributed in $\{0, \ldots, 255\}$, compute the expected L1 distortion per pixel.

Hint: Realize that you have essentially four types of pixels depending on their last two LSBs or, equivalently, the remainder after dividing by 4 ($c_i = 4k$, $4k+1$, $4k+2$, and $4k+3$, $k = 0, 1, \ldots, 63$). The distortion will depend on the class into which the pixel belongs. Assuming that the pixel values are uniformly distributed means that the expected distortion per pixel will be the arithmetic average of the expected distortion per each pixel type.

LSBM2 is a version of LSBM also capable of embedding two bits per pixel:

> *Emb*: Two bits are embedded at each pixel by always modifying the pixel value to the *closest* value with the required least two LSBs. Example: if $c_i = 14 = (00001110)_2$ and we

want to embed message bit pairs 00, 01, 10, 11, we change c_i to 12, 13, 14, 15, respectively. If $c_i = 32 = (00100000)_2$, we embed the same bit pairs by changing c_i to 32, 33, 34, and 31, etc. Note that the last change from 32 to 31 will lead to modification of six LSBs!

Ext: Again, two bits are extracted from the least two LSBs of pixels to form the message.

Assuming that the message is a random bit-stream of maximal length (2× number of pixels) and the pixel values are uniformly distributed in {0, …, 255}, compute the expected L1 embedding distortion per pixel. For simplicity, ignore the boundary issues. Which scheme has a lower expected distortion and why?

A.2 Impact of LSBR2 and LSBM2 on the histogram

Derive the formulas expressing the impact of embedding using LSBR2 and LSBM2 on the histogram $h_\alpha[i]$, $i = 0, \dots, 255$. Assume that the cover is an 8-bit grayscale image with pixel values in the set {0, …, 255} and that the secret message is a random bit-stream of relative length α. Following the derivations from the lecture, find the expected value of $h_\alpha[i]$ as a function of the cover image histogram $h_0[i]$ and the relative message length α. For LSBM2, work out also the correct formulas for the boundary bins.

Note: In LSBM2, there can be ambiguous cases when there are *two* closest values with the same two LSBs, e.g., if the cover pixel value is 7 (the last two LSBs are 11) and if one needs to embed message bits 01, one could change 7 to either 9 or 5 with the same distortion (modifying by 2). Resolve this ambiguity by always selecting the option that disturbs fewer bits. In this case, it means changing 7 to 5 rather than 9.

A.3 LSBR of a biased bit-stream

Assume that the cover is an 8-bit grayscale image with pixel values in the set {0, …, 255}. Suppose that the secret message is random *biased* bit-stream, i.e., the bits are i.i.d. (**i**ndependent and **i**dentically **d**istributed) realizations of a binary random variable B with probability mass function

$$\Pr\{B = 0\} = p_0$$
$$\Pr\{B = 1\} = p_1$$

with $p_0 + p_1 = 1$. Let $h_\alpha[i]$ be the image histogram after embedding with LSBR a secret biased message of relative length α in the cover image ($h_\alpha[i]$ is the number of pixels with grayscale i, $i = 0, 1, \dots, 255$ in the stego image). Following the derivations from the lecture, find the expected value of $h_\alpha[i]$ as a function of the cover image histogram $h_0[i]$ and the parameters α, p_0, and p_1. Additionally, show that for a fully embedded image ($\alpha = 1$):

$$h_1[2i] \,/\, h_1[2i+1] = p_0/p_1 \quad \text{for all } i = 0, \dots, 127.$$

Can you think of a way to use this result for generalizing the histogram attack using the chi-square test?

Hint: You may use the following "sanity checks" to verify that your calculations are correct (they do not *guarantee* correctness, though):

a) Verify that $h_\alpha[2i] + h_\alpha[2i+1]$ is independent of α, p_0, p_1, for all $i = 0, 1, \ldots, 127$.
b) Verify that for $p_0 = p_1 = ½$, you get the formula from the lecture for random unbiased message.

A.4 Bit-plane analysis tool

Write a Matlab function that can be used for visual steganalysis of LSB embedding. It takes on the input:

- A path to an image X (the program should be able to properly work with ***both*** grayscale and color images)
- An integer L from the set $\{1, 2, \ldots, 8\}$

and outputs the following:

- A figure with a histogram of pixel values of X. If X is a color image, three histograms will be displayed in one figure – one for each color channel. Use your esthetic feeling to make the output look "nice." You may choose, for example, to render the histogram as a curve using the Matlab 'plot' function and color it based on the color channel (make it black for a grayscale image).
- Another figure(s) showing the LSB bitplane(s). For a grayscale image, the figure will be a black-and-white rendering of the Lth bit plane (L=8 for LSB and L=1 for the MSB). If the image is a color image, show three figures, one for each color (make the figures red-and-white, green-and-white, and blue-and-white so that the user immediately and easily recognizes to which color channel each plot belongs).

Use your program to analyze whether the image 'stego_white_pocket_test.bmp' contains a secret message and what the message is. Report on your findings, include the outputs of your code, and interpret the output of your program. Enclose the code as an m-file to your submission so that I can test it myself.

A.5 Repetitive embedding using LSBR

Let X be an 8-bit grayscale cover image and Y be the stego image after embedding with LSBR in X a random unbiased message of relative length α_1 (along a pseudo-random path). Continue by embedding in Y using LSBR another random unbiased message of relative length α_2, obtaining a doubly-embedded stego image Z. Assume that the paths for both embeddings are *randomly* chosen and *independently* of each other. The stego image Z will appear to have been embedded with a random unbiased message of relative length $l(\alpha_1, \alpha_2)$. Show that

$$l(\alpha_1, \alpha_2) = \alpha_1 + \alpha_2 - \alpha_1\alpha_2 . \qquad [1]$$

Furthermore, consider the case when you embed messages of relative length α repetitively into the same image k-times. Show that the relative length $l_k(\alpha)$ of the message that appears to have been embedded after k repetitive embeddings is

$$l_k(\alpha) = 1 - (1 - \alpha)^k.$$

Note that $l_k(\alpha) \to 1$ with $k \to \infty$ exponentially fast, for any α positive.

Hint: This is really just a high-school level exercise on probability and Venn diagrams. Do not "overthink" this problem. **Good advice: do not work with relative payloads.** Instead, think in terms of **change rates** (= probabilities of making embedding changes). What is the fraction of pixels flipped during the first embedding? How many of them get unflipped (flipped back) during the second embedding? Then count how many will end up flipped after the second embedding.

The expression for $l_k(\alpha)$ is easiest shown using mathematical induction with respect to k: Show that the expression is true for $k = 1$. Then, assuming it is valid for k, show that it is valid for k+1 (embedding k+1 times is the same as embedding k times and then again one more time – this is where you use the induction step [1] on message lengths $l_k(\alpha)$ and α).

You can e-mail your assignment to me (fridrich@binghamton.edu), slide a hard copy under my door, or give it to me in person at any time (e.g., after a lecture). If e-mailing the answers, please, put everything into a single document (pdf or MS Word). If an assignment requires Matlab code, provide the m-function in a separate m file and e-mail it to me. Please, make sure that all copied / photographed / scanned handwritten notes are legible. If I am unable to read your handwriting or if the copy quality is poor, it cannot help you!

Optional problem

O.1 Bound on maximal message length hidden using LSBR

Let $h_\alpha[i]$ be the histogram of an 8-bit grayscale image embedded using LSBR with a random unbiased message of relative length α. Prove the following upper bound

$$\alpha \le \min_i 2\frac{\min\{h_\alpha[2i], h_\alpha[2i+1]\}}{h_\alpha[2i]+h_\alpha[2i+1]}.$$