Algorithms of Information Security: Steganography II

Faculty of Information Technology
Czech Technical University in Prague

December 14, 2022

Image compression

- Image processing refers to processing an image into digital image.
- Image compression is a type of data compression applied to digital images, to reduce their cost for storage or transmission.
- Algorithms may take advantage of visual perception and the statistical properties of image data to provide superior results compared with generic data compression methods which are used for other digital data.
- Image compression may be lossy or lossless.

Image compression

- Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics.
- Lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossy methods are especially suitable for natural images, such as photographs, in applications where the minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. Lossy compression that produces negligible differences may be called visually lossless.

Transform-domain formats

- One of the lossy compression methods is transform coding.
- It is a mathematical procedure done in data (usually digital images or music) that converts it from one domain (time, for example) to another (frequency), usually doing Fourier's or Laplace's Transforms.
- In the new domain the data could be more easily handled, for lossy compression, de-noising, sharpening, etc. After edited, data is transformed back to its original domain.
- In images, transform coding is mostly used for noise filters (focusing, blurring, etc.) and watermarks.
- An image in a digital form is basically a collection of pixels.
- The edge pixels are known to have high frequency whereas nonedge pixels are known to be low frequency pixels.

Transform-domain formats (JPEG)

- The two most commonly used transforms today are the Discrete Cosine Transform (DCT) and the Discrete Wavelet Transform (DWT).
- The DCT is a type of Fourier-related transform, and was originally developed by Nasir Ahmed, T. Natarajan and K. R. Rao in 1974. It is generally the most efficient form of image compression. The DCT is used in JPEG, the most popular lossy format, and the more recent HEIF.
- The DWT is also used extensively, followed by quantization and entropy coding. The DWT is used in JPEG2000.
- The term "JPEG" is an initialism/acronym for the Joint Photographic Experts Group, which created the standard in 1992.

YC_rC_b and YUV models.

- The YC_rC_b color space is derived from the RGB color space and has the following three components: Y is Luminance or Luma component, C_r and C_b are the red and blue chroma components.
- Digital image formats that use the YC_rC_b model include IIF, TIFF, JFIF, JPEG, and MPEG (motion JPEG).
- YC_rC_b is often confused with the YUV color model, which was originally developed for transmission of color TV signals.
- The YUV model defines a color space in terms of one luma component (Y) and two chrominance components, called U (blue projection) and V (red projection) respectively.
- The luminance Y is defined as weighted linear combination of the RGB channels with weights determinated by the sensitivity fo the human eye to the three RGB colors,

$$Y = 0.299R + 0.587G + 0.114B. \tag{1}$$

YC_rC_b and YUV models.

- The chrominance components are the differences

$$U = R - Y, \tag{2}$$

$$V = B - Y \tag{3}$$

conveying the color information.

- If the RGB colors are represented using 8-bit integers in the range $\{0, \ldots, 255\}$, the luminance Y shares the same range, while U and V fall into the range $\{-179, \ldots, 179\}$.
- To adjust all three components to the same range representable using 8 bits, the chrominance components are further linearly transformed to C_r and C_b, obtaining thus the YC_rC_b color model

$$\begin{pmatrix} Y \\ C_r \\ C_b \end{pmatrix} = \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} + \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.419 & -0.081 \\ -0.169 & -0.331 & 0.5 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{4}$$

Transform-domain formats (JPEG).

JPEG compression consists of the following five basic steps.

1. **Color transformation.** The color is transformated from the RGB model to the YC_rC_b model. Although this step is not necessary (JPEG can work directly with the RGB representation) it is typically used because it enables higher compression ratios at the same fidelity.
2. **Division into blocks and subsampling.** The luminance signal Y is divided into 8×8 blocks. The chrominance signals C_r and C_b may be subsampled before dividing into blocks.
3. **DCT transform.** The YC_rC_b signals from each block are transformed from the spatial domain to the frequency domain with the DCT. The DCT can be tought of as a change of basis representing 8×8 matrices.

Transform-domain formats (JPEG).

4. **Quantization.** The resulting transform coefficients are quantized by dividing them by integer value (quantization step) and rounded to the nearest integer. The luminance and chrominance signals may use different quantization tables. Larger values of the quantization steps produce a higher compression ratio but introduce more perceptual distortion.
5. **Encoding and lossless compression.** The quantized DCT coefficients are arranged in a zig-zag order, encoded using bits, and then losslessly compresed using Huffman or arithmetic coding. The resulting bit stream is prepended with a header and stored with extension '.jpg' or '.jpeg'.

Transform-domain formats (JPEG) - Decompression.

- To view a JPEG image, we first need to obtain the spatial-domain representation from the JPEG file, which is achieved essentially by reversing the five steps above.
- The JPEG bit stream is first parsed, then decompressed, and then the two-dimensional array of quantized DCT coefficients is formed.
- The coefficients in each block are then multiplied by the quantization steps and the inverse DCT is applied to produce raw pixel values.
- Finally, the values are rounded to integers from a certain dynamic range (usually the set $\{0, \ldots, 255\}$).
- While lossless compression and the DCT are reversible processes, the quantization in Step 4 is irreversible and, in general, the decompressed image will not be identical to the original image before compression.

Transform-domain formats (JPEG).

- The chrominance signals, C_r, C_b, are typically downsampled before applying the DCT to achieve a higher compression ratio. This is executed by initially dividing the image into macroblocks of 16×16 pixels.
- Each macroblock produces four 8×8 luminance blocks and 1, 2, or 4 blocks for each chrominance, depending on how the chrominance in the macroblock is subsampled.
- *Example.* If it is subsampled by a factor of 2 in each direction, each macroblock will have 4 luminance blocks and only one 8×8 chrominance C_r block and one chrominance C_b block. This is usually written in an abbreviated from 4:1:1. If neither chrominance signal is subsampled, we have the 4:4:4 representation.

Transform-domain formats (JPEG).

- If the image dimensions, $M \times N$, are not multiplies of 8, the image is padded to the nearest larger multiplies, $8\lceil \frac{M}{8} \rceil \times 8\lceil \frac{N}{8} \rceil$. During decompression, the padded part are not displayed.
- Before applying the DCT, all pixel values are shifted by subtracting 128 from them.
- For an 8×8 block of luminance (or chrominance) values $B[i,j], i,j = 0,\ldots,7$, the 8×8 block of DCT coefficients $d[k,l], k,l = 0,\ldots,7$, is computed as a linear combinations of luminance values,

$$d[k,l] = \sum_{i,j=0}^{7} f[i,j;k,l]B[i,j] \tag{5}$$

Transform-domain formats (JPEG).

- $$d[k,l] = \sum_{i,j=0}^{7} \frac{w[k]w[l]}{4} \cos\frac{\pi}{16}k(2i+1)\cos\frac{\pi}{16}l(2j+1)B[i,j], \quad (6)$$

  where

  $f[i,j;k,l] = \left(\frac{w[k]w[l]}{4}\cos\frac{\pi}{16}k(2i+1)\cos\frac{\pi}{16}l(2j+1)\right)$ and

  $w[0] = \frac{1}{\sqrt{2}}, w[k>0] = 1.$

- The coefficient $d[0,0]$ is called the DC coefficient (or the DC term), while the remaining coefficients with $k+l>0$ are called AC coefficients.
- The DCT is invertible and the inverse transform (IDCT) is

$$B[i,j] = \sum_{k,l=0}^{7} \frac{w[k]w[l]}{4} \cos\frac{\pi}{16}k(2i+1)\cos\frac{\pi}{16}l(2j+1)d[k,l]. \quad (7)$$

LSB embedding

- LSB embedding belongs to the class of steganographic algorithms that embed each message bit at one cover element. In other words, each bit is located at a certain element.
- The embedding proceeds by visiting individual cover elements and applying the embedding operation (flipping the LSB) if necessary, to match the LSB with the message bit.
- The embedding operation of flipping the LSB can be written mathematically in many different ways and one of them is:

$$LSBflip(x) = \begin{cases} x+1 & \text{when } x \text{ even} \\ x-1 & \text{when } x \text{ odd} \end{cases} \tag{8}$$

LSB embedding

- LSBflip is an idempotent operation satisfying $LSBflip(LSBflip(x)) = x$.
- This means that repetitive LSB embedding will partially cancel itself out and thus there is a limit on the maximal expected distortion that can be introduced by repetitive LSB embedding.
- Many other embedding operations do not have this property.
- Partial cancellation of embedding changes can be used to attack schemes that use LSB embedding.

LSB embedding

- LSB embedding also induces 2^{n_c-1} disjoint LSB pairs on the set of all possible element values $\{0, 1, \ldots, 2^{n_c} - 1\}$,

$$\{0, 1\}, \{2, 3\}, \ldots, \{2^{n_c} - 2, 2^{n_c} - 1\}. \tag{9}$$

- Note that if $x[i]$ is in LSB pair $\{2k, 2k + 1\}$, it must stay there after embedding because the pair elements differ only in their LSBs $(2k \leftrightarrow 2k + 1)$.
- For any steganographic method, it is often valuable to mathematically express the impact of embedding on the image histogram.
- Many steganographic techniques introduce characteristic artifacts into the histogram and these artifacts can be used to detect the presence of secret messages.

LSB embedding

- Let $h[j], j = 0, \ldots, 2^{n_c} - 1$ be the histogram of elements from the cover image

$$h[j] = \sum_{i=1}^{n} \delta(x[i] - j), \tag{10}$$

where δ is the Kronecker delta,

$$\delta(x) = \begin{cases} 1 & \text{when } x = 0 \\ 0 & \text{when } x \neq 0. \end{cases} \tag{11}$$

- Let's assume that Alice is embedding a stream of m random bits. The assumption of randomness is reasonable because Alice naturally wants to minimize the impact of embedding and thus compresses the message and probably also encrypts to further improve the security of communication.

LSB embedding

- Denote by $\alpha = m/n$ the relative payload Alice communicates.
- Assuming Alice embeds the bits along a pseudo-random path through the image, the probability that a pixel is not changed is equal to the probability that it is not selected for embedding, $1 - \alpha$, plus the probability that it is selected, α, multiplied by the probability that no change will be necessary, which happens with probability $\frac{1}{2}$ because we are embedding a random bit stream. Thus, for any j,

$$\Pr\{y[i] = j \mid x[i] = j\} = 1 - \alpha + \frac{\alpha}{2} = 1 - \frac{\alpha}{2} \tag{12}$$

$$\Pr\{y[i] \neq j \mid x[i] = j\} = \frac{\alpha}{2}. \tag{13}$$

LSB embedding

- During LSB embedding the pixel values within one LSB pair $\{2k, 2k+1\}, k = 0, \ldots, 2^{n_c-1} - 1$, are changed into each other but never to any other value, that's why, the sum $h_\alpha[2k] + h_\alpha[2k+1]$ stays unchanged for any α and thus forms an invariant under LSB embedding.
- h_α is the histogram of the stego image.
- Thus, the expected value of $h_\alpha[2k]$ is equal to the number of pixels with values $2k$ that stay unchanged plus the number of pixels with values $2k+1$ that were flipped to $2k$:

$$\mathsf{E}[h_\alpha[2k]] = \left(1 - \frac{\alpha}{2}\right) h[2k] + \frac{\alpha}{2} h[2k+1], \tag{14}$$

$$\mathsf{E}[h_\alpha[2k+1]] = \frac{\alpha}{2} h[2k] + \left(1 - \frac{\alpha}{2}\right) h[2k+1]. \tag{15}$$

LSB embedding

- *Note.* If Alice fully embeds her cover image with n bits $(\alpha = 1)$, then

$$\mathsf{E}[h_1[2k]] = \mathsf{E}[h_1[2k+1]] = \frac{h[2k] + h[2k+1]}{2}, k = 0, \ldots, 2^{n_c - 1} - 1. \tag{16}$$

- LSB embedding has tendency to even out the histogram within each bin. This leads to a characteristic staircase artifact in the histogram of the stego image, which can be used as an identifying feature for fully embedded with LSB embedding. This observation is quantified in the so-called histogram attack.

Histogram attack

- In fully embedded stego image $(\alpha = 1)$, we expect

$$h_\alpha[2k] \approx \overline{h}[2k] = \frac{h_\alpha[2k] + h_\alpha[2k+1]}{2}, k = 0, \ldots, 2^{n_c - 1} - 1. \quad (17)$$

- The histogram attack includes to the following composite hypothesis - testing problem:

$$H_0 : h_\alpha \sim \overline{h}, \quad (18)$$

$$H_1 : h_\alpha \nsim \overline{h}, \quad (19)$$

which we approach using Pearson's chi-square test.

Histogram attack

- This test determines whether the even grayscale values in the stego image follow the known distribution $\overline{h}[2k], k = 0, \ldots, 2^{n_c-1} - 1$.
- The chi-square test first computes the test statistic S,

$$S = \sum_{k=0}^{d-1} \frac{\left(h_\alpha[2k] - \overline{h}[2k]\right)^2}{\overline{h}[2k]} \tag{20}$$

  where $d = 2^{n_c-1}$.
- Under the null hypothesis, the even grayscale values follow the probability mass function, $\overline{h}[2k]$, and the test statistic approximately follows the chi-square distribution, $S \sim \chi^2_{d-1}$, with $d - 1$ degrees of freedom. That is as long as all d bins, $h[2k], k = 0, \ldots, 2^{n_c-1} - 1$, are sufficiently populated. Any unpopulated bins must be merged so that $\overline{h}[2k] > 4$ for all k, to make S approximately chi-square distributed.

Histogram attack

- Note that, if the even grayscales follow the expected distribution, the value of S will be small, indicating the fact that the stego image is fully embedded with LSB embedding.
- Large values of S mean that the match is poor and notify us that the image under inspection is not fully embedded.
- Let's construct a detector of images fully embedded using LSB embedding by setting a threshold γ on S and decide "cover" when $S > \gamma$ and "stego" otherwise.
- The probability of failing to detect a fully embedded stego image (probability of missed detection) is the conditional probability that $S > \gamma$ for a stego image,

$$P_{MD}(\gamma) = \mathrm{Pr}\left\{S > \gamma \mid x \text{ is stego}\right\}. \tag{21}$$

We set the threshold γ so that the probability of a miss is at most P_{MD}.

Histogram attack

- Denoting the probability density function of χ^2_{d-1} as $f_{\chi^2_{d-1}}(x)$, the threshold is determined from (21),

$$P_{MD}(\gamma) = \int_{\gamma}^{\infty} f_{\chi^2_{d-1}}(x)dx = \int_{\gamma}^{\infty} \frac{e^{-\frac{x}{2}} x^{\frac{d-1}{2}-1}}{2^{\frac{d-1}{2}}\Gamma\left(\frac{d-1}{2}\right)}dx. \quad (22)$$

- The value $P_{MD}(\gamma)$ is called the $p-$value and it measures the statistical significance of γ.
- It is the probability that the a chi-square-distributed random variable with $d-1$ degrees of freedom would atain a value larger that or equal to γ.

Targeted steganalysis

- The features in targeted steganalysis are constructed from knowledge of the embedding algorithm and thus targeted to a specific embedding method (e.g., LSB embedding).
- Targeted steganalysis method can work very well with a single scalar feature.
- Targeted steganalysis can use multiple features and tools from machine-learning and pattern recognition.
- In targeted steganalysis the embedding mechanism of the stego system is known, it makes sense to choose as features the quantities that predictably change with embedding.

Targeted steganalysis

- Let's β is the change rate (ratio between the number of embedding changes and the number of all elements in the cover image) and $\mathbf{f}_\beta$ is the feature computed from a stego image obtained by changing the ratio of β of its corresponding cover elements.
- There are several strategies for constructing features for targeted steganography, for example:
  Testing for stego artifacts. Identify a feature that attains a specific known value, $\mathbf{f}_\beta$, on stego images and attains others, different values on cover images. Then formulate a composite hypothesis-testing problem as

$$H_0 : \mathbf{f} = \mathbf{f}_\beta, \tag{23}$$

$$H_1 : \mathbf{f} \neq \mathbf{f}_\beta, \tag{24}$$

- Note that here H_0 is the hypothesis that the image under investigation is stego, while H_1 stands for the hypothesis that is cover.

Targeted steganalysis

- This strategy was pursued when deriving the histogram attack. The attack was based on the observation that the histogram h of an 8-bit grayscale image fully embedded with LSB embedding must satisfy

$$h[2k] \approx h[2k+1], k = 0, \ldots, 127. \tag{25}$$

- Because the sum $h[2k] + h[2k+1]$ is invariant with respect to LSB embedding, we can take as a feature vector the histogram h and formulate the following composite hypothesis-testing problem:

$$H_0 : h[2k] = \frac{h[2k] + h[2k+1]}{2}, k = 0, \ldots, 127, \tag{26}$$

$$H_1 : h[2k] \neq \frac{h[2k] + h[2k+1]}{2}, k = 0, \ldots, 127, \tag{27}$$

which leads to the histogram attack, where this problem was approached using Pearson's chi square test.