

Image Watermarking

cs4: Computer Science Bootcamp

Çetin Kaya Koç

<http://koclab.cs.ucsb.edu/teaching/cs4>

cetinkoc@ucsb.edu

Image Size and Pixel Color

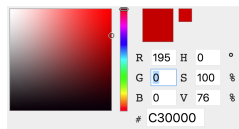
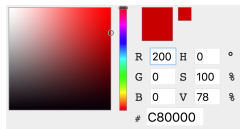
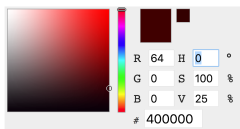
- Consider our sample image `leo1.gif`, which is of size 500×500 , represented using RGB mode



- In the RGB model, we have 3 bytes for each pixel, and therefore, $2^8 \times 2^8 \times 2^8 = 2^{24} = 16,777,216$ different colors
- Each pixel has 3 color densities: red, green, blue
- Each color value is an integer between 0 and 255, e.g., $(r, g, b) = (175, 89, 67) = (AF, 59, 3E)$

Pixel Colors

- However, small differences between color values are not visible to human eye and most display technologies cannot display them
- For example, let's keep the green and blue values constant (say, at 0), and change the red color value
- As we change r value from 00 to FF, the red color intensity changes
- While there is a visibly big difference between C8 and 64, there is little (almost invisible) difference between C8 and C3

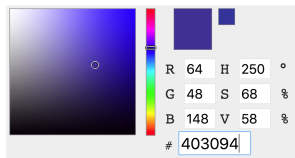
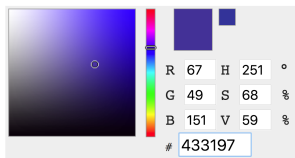


- Lower bits of each color value is almost irrelevant, in the sense that small changes in them will make no difference to human eye

Pixel Colors

- In the previous experiment, we changed only the red color, but we can change all three colors simultaneously
- Provided that each change is small (in the lower bits), the total pixel color difference is still invisible
- Consider the purplish color: 433197, and change the lower 2 bits of each color to zero; the new color becomes 403094, as

43 : 0100 0011 → 0100 0000 : 40
 31 : 0011 0001 → 0011 0000 : 30
 97 : 1001 0111 → 1001 0100 : 94



Pixel Colors

- For example, if we write zeros in the last 2, 4, and 6 bits of each pixel color in `leo.gif`, we obtain the following images



2 bits



4 bits



6 bits

Watermarking Basics

- This property gives us an idea: Hide an image inside another one
- We will lose some resolution, obviously, however, this may not matter if the purpose is to mark the original image
- Since each color value is 8 bits, we can take every pixel (r, g, b) from the original image and use lower d bits of it to embed the higher d bits of the second (watermark) image, where $d = 1, 2, 3, \dots$
- This process is done for every color intensity and for every pixel in both images
- It is expected that the original and watermarked images be the same size, if not, the watermark image is clipped or expanded (with white space) to make them the same size

Watermarking Basics

- For example, for $d = 3$, the watermarking process is depicted as

Original Image

| | | | | | | | | | |
|-------|-------|-------|-------|-------|--|-------|-------|-------|---|
| r_7 | r_6 | r_5 | r_4 | r_3 | <table border="1"><tr><td>r_2</td><td>r_1</td><td>r_0</td></tr></table> | r_2 | r_1 | r_0 | ← |
| r_2 | r_1 | r_0 | | | | | | | |
| g_7 | g_6 | g_5 | g_4 | g_3 | <table border="1"><tr><td>g_2</td><td>g_1</td><td>g_0</td></tr></table> | g_2 | g_1 | g_0 | ← |
| g_2 | g_1 | g_0 | | | | | | | |
| b_7 | b_6 | b_5 | b_4 | b_3 | <table border="1"><tr><td>b_2</td><td>b_1</td><td>b_0</td></tr></table> | b_2 | b_1 | b_0 | ← |
| b_2 | b_1 | b_0 | | | | | | | |

Watermark Image

| | | | | | | | | |
|---|--------|--------|--------|--------|--------|--------|--------|--------|
| <table border="1"><tr><td>r'_7</td><td>r'_6</td><td>r'_5</td></tr></table> | r'_7 | r'_6 | r'_5 | r'_4 | r'_3 | r'_2 | r'_1 | r'_0 |
| r'_7 | r'_6 | r'_5 | | | | | | |
| <table border="1"><tr><td>g'_7</td><td>g'_6</td><td>g'_5</td></tr></table> | g'_7 | g'_6 | g'_5 | g'_4 | g'_3 | g'_2 | g'_1 | g'_0 |
| g'_7 | g'_6 | g'_5 | | | | | | |
| <table border="1"><tr><td>b'_7</td><td>b'_6</td><td>b'_5</td></tr></table> | b'_7 | b'_6 | b'_5 | b'_4 | b'_3 | b'_2 | b'_1 | b'_0 |
| b'_7 | b'_6 | b'_5 | | | | | | |

Watermarked Image



| | | | | | | | | |
|--------|--------|--------|-------|-------|---|--------|--------|--------|
| r_7 | r_6 | r_5 | r_4 | r_3 | <table border="1"><tr><td>r'_7</td><td>r'_6</td><td>r'_5</td></tr></table> | r'_7 | r'_6 | r'_5 |
| r'_7 | r'_6 | r'_5 | | | | | | |
| g_7 | g_6 | g_5 | g_4 | g_3 | <table border="1"><tr><td>g'_7</td><td>g'_6</td><td>g'_5</td></tr></table> | g'_7 | g'_6 | g'_5 |
| g'_7 | g'_6 | g'_5 | | | | | | |
| b_7 | b_6 | b_5 | b_4 | b_3 | <table border="1"><tr><td>b'_7</td><td>b'_6</td><td>b'_5</td></tr></table> | b'_7 | b'_6 | b'_5 |
| b'_7 | b'_6 | b'_5 | | | | | | |