

CS 443/543 Homework 2—Chroma Subsampling

Due: Mar. 27 Midnight

CS443 students: Up to two students can collaborate on this homework. (Leave a comment on the canvas indicating who you collaborated with.)

CS543 students: You must complete this homework independently.

Chroma subsampling is a technique used in image encoding that involves encoding chroma information (Cb and Cr) at a lower resolution than luma information (Y). This method exploits the human visual system's reduced sensitivity to color differences compared to luminance differences. For this assignment, you are required to implement Chroma Subsampling in the 4:2:0 and 4:2:2 formats.

Dr. Chung has provided two sample images for this assignment, available in the "HW2.zip" file from the Modules. In your report and output images, you are required to apply the subsampling algorithms to these sample images.

Part 1: Subsampling function

- 1) Transform the (normalized) RGB color values of your input images into YCbCr components by utilizing the conversion formula provided below. Avoid using any existing functions or codes, such as Matlab's `rgb2ycbcr()`, for this color space conversion.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16874 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix}$$

- 2) Perform chroma subsampling (4:2:0 and 4:2:2) independently on the Cb and Cr color components. Apply both subsampling methods to each 2×4 pixel block in the input image. After subsampling, fill the vacant pixels within each 2×4 block with the chosen (subsampled) colors.
- 3) Convert the subsampled YCbCr color data back to the RGB color space. Then, generate and save your output images for both 4:2:2 and 4:2:0 configurations. Ensure that you do not utilize any built-in functions for the conversion process; instead, employ matrix calculations. All output images should be saved in the .png format.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.77200 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb - 0.5 \\ Cr - 0.5 \end{bmatrix}$$

Part 2: Subsampling GUI

Design and develop a GUI application using MATLAB's app designer. Ensure that the app and its image panels are adequately sized. Your GUI should incorporate the following elements:

The GUI must feature the following widgets (but you can add any widgets if you want):

- Read and Input – Loads and displays the original input (4:4:4) RGB image, similar to HW1's GUI.
- Output 1 – Shows the subsampled result in RGB based on the selected subsampling option.
- Output 2 – Displays a subsampled output showing the grayscale version of either the Cr or Cb component (only one of the two).
- Subsampling Selection Options – Provide the user with the ability to choose from different subsampling configurations using radio buttons. These widgets will enable users to select either 4:2:2 or 4:2:0.
- A button for performing the selected subsampling.

Part 3: Report

In your report, provide a comparison and discussion of the observed outcomes from subsampling the images in 4:2:2, 4:2:0, and 4:4:4 formats. Highlight the visual differences among these images. Additionally, include a 2x4 block from each output, presenting the pixel values directly for detailed analysis.

To summarize, your report should contain:

- List the names of all team members if you collaborated on this. (Leave a comment on the canvas indicating who you collaborated with.)
- For the two sample inputs:
 - Output images – RGB images generated using 4:4:4, 4:2:2, and 4:2:0 chroma subsampling.
 - Subsampled Cb and Cr images – Grayscale representations of the Cb and Cr channels from 4:4:4, 4:2:2, and 4:2:0 subsampling.
 - Block-wise output comparison – Extracted 2x4 blocks from the same position in each image for 4:4:4, 4:2:2, and 4:2:0, including their corresponding pixel values.
- Observations & Comparisons:

- Discuss the visual differences among the subsampling methods.
- Identify any patterns or noticeable artifacts resulting from subsampling.
- Etc.

(d) GUI Screenshots:

- Provide screenshots demonstrating the layout and functionality of your graphical user interface (GUI).

(e) MATLAB Source Code:

- Include the MATLAB code used for GUI creation, subsampling, and conversion processes.

Submission:

Compress the following files into a single ZIP archive:

1. Output PNG images with subsampled Cb and Cr in grayscale.
2. MATLAB code.
3. Report (in pdf).

Do not use generative AI or download any code from the internet. For this assignment, you must not use any pre-existing MATLAB functions that provide the same functionality (e.g., `rgb2ycbcr()`). Instead, implement them from scratch using basic MATLAB functions.