

The goal of this assignment is to explore the process of gamut mapping. There are many different approaches that can be taken to gamut mapping, as described, for example, in the textbook on gamut mapping by Jan Morovic. For this assignment, you will explore one particular approach that has been developed by a series of graduate students in my laboratory over a period of approximately 30 years. Jiayin Liu and Baekdu Choi have graciously provided the codes that you will execute to study each step of our gamut mapping process. They are all written in Matlab.

Gamut mapping algorithms can be quite complex to implement. To allow you to more easily understand the gamut mapping process, we have done the bulk of the coding for you. You just need to run the codes as directed in the two README files, put the results in your homework paper, and comment on what they mean. To better understand how our gamut mapping procedure works, it may be helpful to read the lecture notes Module 37.0 New Developments in Gamut Mapping. Modules 37.1 and 37.2 may also be of interest.

The codes can be divided into two parts: (1) executing the actual gamut mapping (codes provided by Jiayin), and (2) visualizing the results of the gamut mapping (codes provided by Baekdu). I have uploaded all the codes and a few data files to the course website under HW No. 8, and placed everything in the folder GamutMapping. In that folder, there is a folder called input, which contains the input image musicians.tif that you will use. The instructions for running the gamut mapping codes can be found in the file README_1_GamutMapping in the README folder. The instructions for running the visualization codes can be found in the file README_2_mask_diff.

The source gamut that you will be using is the actual gamut of the input image. Thus, the gamut mapping algorithm is image-dependent. The destination gamut is an idealized polyhedral approximation of the gamut for the Indigo press, when printing is done using only C, M, and Y colorants (called Enhanced Productivity Mode or EPM). The vertices of this gamut are given by the 8 Neugebauer primaries.

1. Following the instructions in README_1_GamutMapping, perform Commands 1, 2, and 3 (as designated in the README file). Examine the plots showing the superposition of the source and destination gamuts, using the 3D meshplot view in Matlab. To answer the following questions, you will need to rotate the plot to many different views:
 - a. What are the most saturated colors in the input image (source gamut), corresponding to the largest chroma values $C = \sqrt{C_x^2 + C_z^2}$?
 - b. What is the lightest color in the image, corresponding to the largest Y_y value?
 - c. What is the darkest color in the image, corresponding to the smallest Y_y value?
 - d. What are the most saturated colors in the destination gamut, corresponding to the largest chroma values $C = \sqrt{C_x^2 + C_z^2}$?
 - e. What is the lightest color in the destination gamut, corresponding to the largest Y_y value?

- f. What is the darkest color in the destination gamut, corresponding to the smallest Y_y value?
 - g. For what colors in the input image is the mismatch between the source and output gamuts the largest?
 - h. Turn in a 3D view of the overlapped source and destination gamuts that illustrates the mismatch between these two gamuts, and clearly shows where source colors are outside the destination gamut.
2. Following the instructions in README_1_GamutMapping, perform Commands 4 and 5 (as designated in the README file, these are softCompressLightness and gamutMappingStep1).
 - a. Examine the Step 1 hue slices that will be in the folder hueSlicePlots, which is in the folder output, and compare with the Step 0 hue slices, which show the hue slices before any gamut mapping has been performed. Comment on the effect of Step 1 on the Source gamut.
 - b. Turn in a plot of the hue slices (100,105] (the units are degrees) for Step 0 and Step 1. The name of the files is Step 0 Before any compression Hue slice view (100,105].png and Step 1 Hue slice view (100,105].png. Explain what has happened to the source gamut within this hue slice.
 3. Following the instructions in README_1_GamutMapping, perform Command 6 (as designated in the README file, this is gamutMappingStep2).
 - a. Examine the Step 2 hue slices that will be in the folder hueSlicePlots, which is in the folder output, and compare with the Step 1 hue slices. Comment on the effect of Step 1 on the Source gamut.
 - b. Turn in a plot of the hue slices (100,105] (the units are degrees) for Step 1 and Step 2. The name of the files is Step 1 Hue slice view (100,105].png and Step 2 Hue slice view (100,105].png. Explain what has happened to the source gamut within this hue slice in going from Step 1 to Step 2.
 4. Following the instructions in README_1_GamutMapping, perform Command 7 (as designated in the README file, this is gamutMappingStep3).
 - a. Examine the Step 3 hue slices that will be in the folder hueSlicePlots, which is in the folder output, and compare with the Step 2 hue slices. Comment on the effect of Step 1 on the Source gamut.
 - b. Turn in a plot of the hue slices (100,105] (the units are degrees) for Step 2 and Step 3. The name of the files is Step 3 Hue slice view (100,105].png and Step 2 Hue slice view (100,105].png. Explain what has happened to the source gamut within this hue slice in going from Step 2 to Step 3.
 5. Following the instructions in README_1_GamutMapping, perform Commands 8 and 9 (as designated in the README file, these are gamutMappingStep4 and afterProcessGamutPlot).
 - a. Examine the files gamut_src_dest_afterGamutMapping to confirm that the gamut-mapped image colors are now all within the destination gamut.
 - b. Turn in a 3D plot of gamut_src_dest_afterGamutMapping.
 6. Following the instructions in README_1_GamutMapping, perform Command 10 (as designated in the README file, this is yxcz2rgb(musiciansGM.tif).

Visually compare the two images `musicians.tif` (before gamut mapping) and `musiciansGM.tif` (after gamut mapping). Comment on which colors have been significantly modified, and which colors have not been significantly modified. How would you describe the changes that have occurred to the colors that were modified?

For the remaining problem on this assignment, you will use the gamut mapping visualization software that was developed by Baekdu Choi. How to use the software is described in the file `README_2_mask_diff`. As described in that file, this software has two primary outputs: (1) a mask image that is white for pixels that are in gamut, and which is black for pixels that are out of gamut, (2) a color difference image that shows the change in color at each pixel between the original image and the gamut-mapped image. To make the changes more visible, the color difference can be scaled by a factor that is greater than 1. The default value is 3. To prevent the occurrence of non-displayable colors with negative RGB values, an offset of 50 is added to the Yy component.

7. Following the instructions in `README_2_mask_diff`, generate the mask image and difference image for `musicians.tif` and `musiciansGM.tif`.
 - a. Comparing the image `musicians.tif` and the mask image, comment on which colors in the original image are out of gamut. Turn in a print-out of the mask image.
 - b. Comparing the image `musicians.tif` and the difference image, comment on how the gamut mapping has changed the colors in the original image. Note that even colors that are within the destination gamut will be changed. Turn in a print-out of the difference image.