

COL783: Digital Image Processing

## Assignment 1: Artistic Image Enhancement, and Style Transfer

Deadline: 25/08/2023, 10:59 PM

### Introduction

In this assignment, you will perform the rendering of real photos into non-photorealistic painting-like images. Real paintings often demonstrate strong light-shadow contrast, important line emphasis and lively colour scheme. i.e, the artists often follow the following protocol to paint an image.

- First, they simplify the relationship between light and shadow. The shadow area is usually drawn on a separate layer to enhance the contrast between light and dark, especially for cartoon works.
- The line draft is employed not only to describe the overall composition of an artwork but also to emphasize some important shadow lines.
- They tend to keep the colour scheme exaggerated and contrasted to produce stronger visual impact

In this assignment, we take inspiration from the above-mentioned protocol and propose to mimic the artistic observation to develop our image enhancement technique. Later we quantize the output and compare the results. Lastly, we transfer this enhancement from one image to another using the method of style transfer. The assignment is divided into 3 parts.

### Part 1: Simple Image Enhancement

In this part of the assignment, we follow a standard image enhancement method, as shown in Figure 1. This method consists of two main steps, viz. The artistic Enhancement step and the colour adjustment step. In the artistic enhancement step, we extract a simplified shadow layer from the image and use it to enhance the light-shadow contrast. In the colour adjustment step, we adopt the real toning process with a modulating scheme to create lively colours with sharp contrast. Lastly, the results of both steps are combined to generate different rendered outputs.

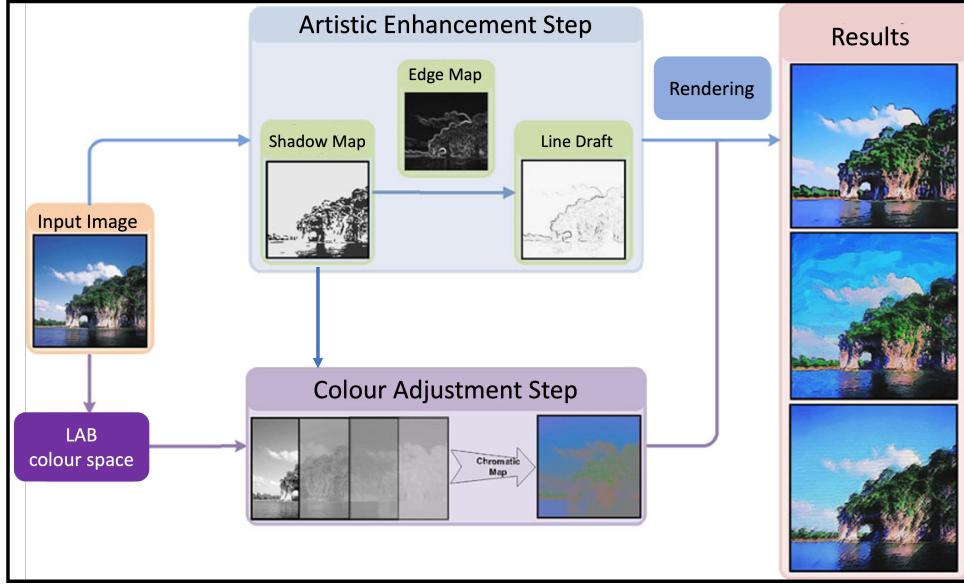


Figure 1: Proposed Artistic Image Enhancement Framework

### 1.1. Artistic Enhancement Step:

The main aim of this step is to treat the light-shadow contrast. It consists of three sub-stages, namely shadow map generation, edge map generation and line draft generation.

#### 1.1.1. Shadow-Map generation

In the shadow map generation, each pixel is assigned a flag to denote whether it belongs to a light area or a shadow area (known as the shadow map).

The Input image is to be converted from RGB colour space to HSI colour space using the following colour conversion formula (1-3)

$$\begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ -\sqrt{6}/6 & -\sqrt{6}/6 & \sqrt{6}/3 \\ 1/\sqrt{6} & -2/\sqrt{6} & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

$$S = \sqrt{(V_1^2 + V_2^2)} \quad (2)$$

$$H = \begin{cases} \tan^{-1} \frac{V_2}{V_1}, & \text{if } V_1 \neq 1 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Normalize the values of  $H$  and  $I$ , so that they come in the range of  $[0, 1]$ .

Now, Compute the ratio map ( $r - map$ ) for each pixel  $(x, y)$  of the image using the equation 4

$$r(x, y) = \frac{H(x, y) + 1}{I(x, y) + 1} \quad (4)$$

The shadow map can now be evaluated as shown in equation 5

$$s(x, y) = \begin{cases} 1, & \text{if } r(x, y) > T \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

The value of the threshold  $T$  is computed as shown in equation 6

$$T = \arg \min_T \left[ \sum_{i=0}^T P(i)(i - \mu_1)^2 + \sum_{i=T+1}^{255} P(i)(i - \mu_2)^2 \right] \quad (6)$$

Where,

$$\mu_1 = \sum_{i=0}^T \frac{iP(i)}{W_1} \quad \mu_2 = \sum_{i=T+1}^{255} \frac{iP(i)}{W_2}$$

$$W_1 = \sum_{i=0}^T P(i) \quad W_2 = \sum_{i=T+1}^{255} P(i)$$

and  $P(i)$  is the probability of occurrence of  $i$  in the  $r - map$  [as seen in histogram]

(**Hint:** The optimal value of  $T$  can be estimated by simply assuming the value of  $T$  from the all possible values of  $T \in [0, 255]$  in iteration. Thus the best value of  $T$  is the one for which the value of Equation 6 is minimum.)

Note that you may have to perform an inversion of your shadow map to make it resemble the one shown in Figure 1. Once the final shadow map is obtained, you need to merge it with the original Image using the formula,

$$\text{Shadowed Image}, \quad SI(x, y) = \begin{cases} \lambda \text{Input}(x, y) + (1 - \lambda)s(x, y), & \text{if } s(x, y) = 0 \\ \text{Input}(x, y), & \text{otherwise} \end{cases}$$

Where  $\text{Input}$  is the input image and  $s$  is the shadow map.  $\lambda$  is the hyper-parameter influencing the effect of the shadow on the original image. You can perform various ablations before coming up with the final value for  $\lambda$ . Also, it must be noted that since  $\text{Input}$  is a colour image, the value of  $SI(x, y)$  will also be a tuple of 3 numbers.

### 1.1.2. Line Draft generation

We follow the below-mentioned procedure to obtain the line-map of the given image.

- **Grayscale Conversion:** Convert the original image to grayscale by using the standard opencv package.
- **Bilateral Filtering:** Use the algorithm explained in the class [1] and perform bilateral filtering on the image. It preserves the edges and performs smoothing of the image. This helps us to get rid of noise and unwanted lines in the image. Experiment with various hyper-parameters of the bilateral filter and come up with the best possible smoothed image.

- **Edge Detection:** Convolve the bilateral filtered image with the  $Sobel_X$  and  $Sobel_Y$  filters separately to obtain the gradients ( $I_X$  and  $I_Y$ ) of the Image. The edges of the image are obtained by

$$\text{Edge Map, } E = \sqrt{I_X^2 + I_Y^2}$$

- **Thresholding:** Apply a thresholding operation to convert the grayscale edge map to a binary image with only black and white pixels. This step helps emphasize the lines and remove unnecessary details. The output of this stage forms the Line Draft of the Image

$$\text{Line Draft, } LD(x, y) = \begin{cases} 1, & \text{if } E(x, y) \geq T \\ 0, & \text{otherwise} \end{cases}$$

Here,  $T$  is the threshold whose value can range between [0,255]. You can perform various ablations before coming up with the final value for  $T$ .

It must be noted that the quality of the line drawing heavily depends on the choice of parameters for bilateral filtering, edge detection and thresholding. Adjusting these parameters according to the specific characteristics of your input image is crucial to achieving the desired results. Also, note that you may have to perform an inversion of your Line Draft to make it resemble the one shown in Figure 1

## 1.2. Colour Adjustment Step:

Often, the paintings appear intense and emotional because they feature vivid colours with strong contrast. We propose to achieve such an effect in this assignment. Towards this, we propose a colour adjustment scheme.

To achieve this, we create a chromatic map by removing the influence of lightness. Specifically, after decomposing the input image in the  $LAB$  colour space, its chromatic map can be obtained by merging the original  $A$  and  $B$  channels with a neutral light channel set as the mid-value of the whole range present in that image. Using this map, the R, G, and B colour components of each pixel are enhanced, as shown in Equation 7

$$SI'(x, y) = SI(x, y) \cdot \frac{1 + \tanh(\rho \cdot (C_M(x, y) - 128))}{2} \quad (7)$$

Where  $SI(x, y)$  is the intensity of the Shadowed Image (obtained after section 1.1.1) at pixel  $(x, y)$ .  $C_M(x, y)$  is the colour of the corresponding pixel in the chromatic map (Convert the chromatic map into RGB and use the corresponding colour).  $\rho$  is a hyper-parameter controlling the enhancement level (typical values are in the range of [0.005, 0.2]). Also, it must be noted that  $SI(x, y)$ ,  $C_M(x, y)$  and  $SI'(x, y)$  are all tuples of three numbers each.

The above-mentioned artistic enhancement and colour adjustment may cause undesired changes to the overall brightness of the input image. To mitigate this, we perform linear saturation correction with the appropriate saturation scale.

$$\hat{SI} = \text{saturation correction}(SI') \quad (8)$$

Now,  $\hat{SI}$  is the colour-enhanced image. This image is added with the line draft image ( $LD$  obtained in section 1.1.2) to obtain the final artistic rendered image.

$$\text{Artistic Rendered Image}, \quad \text{Final}(x, y) = \begin{cases} \beta \hat{SI}(x, y) + (1 - \beta) LD(x, y), & \text{if } LD(x, y) = 0 \\ \hat{SI}(x, y), & \text{otherwise} \end{cases} \quad (9)$$

Here,  $\beta$  is a hyper-parameter whose value can range between [0,1]. You can perform various ablations before coming up with the final value for  $\beta$ .

## Part 2: Quantized Rendering:

In this part, we analyze another mechanism to render the output. We perform image quantization of the result obtained in Part 1. We explore the “median-cut” method of colour quantization. We require results with and without Floyd-Steinberg’s method of dithering. The input image, in this case, will be a 24-bit image (8 bits in Red, Green and Blue components). For the sake of analysis, we assume that this is the original ground-truth image without quantization errors.

The median cut algorithm and Floyd Steinberg dithering are explained in [2]. Implement both of them **without using direct inbuilt functions**. You may have to follow the divide and conquer method to implement the median cut quantization. Compare the results of the quantization method with and without dithering, visually and using quantization error.

## Part 3: Artistic Style Transfer

In this part, we propose to transfer the properties (‘Style’) of artistic rendering from one image to another. Consider two Images, ‘ $In$ ’ and ‘ $Jn$ ’, of similar scenes. We perform the artistic rendering for image ‘ $In$ ’ using the method described in part 1, and we get the artistic rendered version of ‘ $In$ ’ (say ‘ $I_1$ ’). Now, we transfer the ‘style’ of image ‘ $I_1$ ’ to image ‘ $Jn$ ’ so that the image ‘ $Jn$ ’ would also look as though it is artistically rendered.

The task of style transfer can be implemented by following a similar approach as discussed in the task of colour transfer [3]. We convert the image ‘ $Jn$ ’ into grayscale (say ‘ $J_g$ ’). Assume ‘ $I_1$ ’ to be the reference colour image. Follow the protocol as explained in the class [3] and transfer the colour of ‘ $I_1$ ’ to ‘ $J_g$ ’ with and without the use of swatches (For the purpose of this assignment, you can manually annotate the swatches). The colour-transferred Image (say ‘ $J_1$ ’) can be considered as the style-transferred version of ‘ $Jn$ ’.

### Note:-

- Parts of the assignment are taken from [4], and [5]. You are encouraged to read these papers. However, you are informed to stick to the procedure mentioned in this document for implementation.
- The assignment is to be done individually or in a group with a maximum of two members.
- You can use the inbuilt opencv functions to read and write the images. You **CANNOT** use any other inbuilt image processing functions unless explicitly exempted in the description of the assignment.

- You are required to submit the code, data, and a detailed report for this assignment. The submission can be done using Moodle. Submission instructions will be shared later.
- We have a separate held-out test dataset. We shall test the performance of each of your algorithms on that dataset.
- Any sort of plagiarism will lead to serious punishments depending on the case.

## References

- [1] Sylvain Paris, Pierre Kornprobst, Jack Tumblin, Frédo Durand, et al. Bilateral filtering: Theory and applications. *Foundations and Trends® in Computer Graphics and Vision*, 4(1):1–73, 2009.
- [2] Paul Heckbert. Color image quantization for frame buffer display. *ACM Siggraph Computer Graphics*, 16(3):297–307, 1982.
- [3] Tomohisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 277–280, 2002.
- [4] Yu Zang, Hua Huang, and Chen-Feng Li. Artistic preprocessing for painterly rendering and image stylization. *The Visual Computer*, 30:969–979, 2014.
- [5] Bangyu Sun and Shutao Li. Moving cast shadow detection of vehicle using combined color models. In *2010 Chinese conference on pattern recognition (CCPR)*, pages 1–5. IEEE, 2010.