A method for color test target generation and extraction of color values

Shankhya Debnath

August 25, 2024

1 Introduction

A method is presented for generating a color test target based on LAB values and extracting RGB values from a scanned image of the target. These processes involve color space transformations, perspective geometry, and statistical methods. The test target has been made in accordance to the Gretag ColorChecker target. It contains 24 patches.

2 Generating the Test Target Image

The first step is to generate a test target image using LAB color values. The process involves converting LAB values to the RGB color space, arranging color patches on a grid, and adding fiducial markers.

2.1 LAB to RGB Conversion

The LAB color space is device-independent and is designed to approximate human vision. The conversion from LAB to RGB involves several steps:

1. Convert LAB to XYZ color space:

$$\begin{aligned} Y &= \frac{L^* + 16}{116} \\ X &= a^* \cdot 0.002 + Y \\ Z &= Y - b^* \cdot 0.005 \end{aligned}$$

   where L^*, a^*, and b^* are the LAB color values, and X, Y, and Z are the XYZ color values.

2. Convert XYZ to linear RGB:

$$\begin{bmatrix} R_{\text{linear}} \\ G_{\text{linear}} \\ B_{\text{linear}} \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

3. Apply gamma correction to convert to sRGB:

$$R = \begin{cases} 12.92 \cdot R_{\text{linear}}, & R_{\text{linear}} \leq 0.0031308 \\ 1.055 \cdot R_{\text{linear}}^{1/2.4} - 0.055, & R_{\text{linear}} > 0.0031308 \end{cases}$$

$$G = \begin{cases} 12.92 \cdot G_{\text{linear}}, & G_{\text{linear}} \leq 0.0031308 \\ 1.055 \cdot G_{\text{linear}}^{1/2.4} - 0.055, & G_{\text{linear}} > 0.0031308 \end{cases}$$

$$B = \begin{cases} 12.92 \cdot B_{\text{linear}}, & B_{\text{linear}} \leq 0.0031308 \\ 1.055 \cdot B_{\text{linear}}^{1/2.4} - 0.055, & B_{\text{linear}} > 0.0031308 \end{cases}$$

4. Convert the resulting R, G, and B values to 8-bit integers (0-255):

$$R_{\text{8bit}} = \text{round}(R \times 255), \quad G_{\text{8bit}} = \text{round}(G \times 255), \quad B_{\text{8bit}} = \text{round}(B \times 255)$$

2.2 Arranging Color Patches

The color patches are arranged on a grid. The position of each patch is determined by its row r and column c in the grid:

$$\text{Top\_Left\_X} = c \times (\text{Patch\_Width} + \text{Padding}) + \text{Padding} + \text{Marker\_Size}$$

$$\text{Top\_Left\_Y} = r \times (\text{Patch\_Height} + \text{Padding}) + \text{Padding} + \text{Marker\_Size}$$

$$\text{Bottom\_Right\_X} = \text{Top\_Left\_X} + \text{Patch\_Width}$$

$$\text{Bottom\_Right\_Y} = \text{Top\_Left\_Y} + \text{Patch\_Height}$$

2.3 Adding Fiducial Markers

Fiducial markers are added to the corners of the image. These markers are used later for perspective correction. The positions of the markers are:

$$\text{Marker\_Positions} = \{(x, y) | x, y \in \{\text{Marker\_Size}, \text{Image\_Width} - \text{Marker\_Size}, \text{Image\_Height} - \text{Marker\_Size}\}\}$$

3 Reading the Scanned Image and Extracting RGB Values

The second script processes the scanned image, detects the fiducial markers, corrects the perspective, and extracts the RGB values from the patches.

3.1 Fiducial Marker Detection

Fiducial markers are detected by converting the image to grayscale and applying a threshold to create a binary image. Contours are then detected, and those matching the expected size and shape of the fiducial markers are selected:

$$\text{Contour\_Area} = \pi \times (\text{Radius})^2$$

$$\text{if Contour\_Area} > \pi \times (\text{Marker\_Size} \times 0.5)^2\text{, accept the contour as a marker.}$$

3.2 Perspective Transformation

A perspective transformation is applied to align the image so that the patches are rectangular and correctly aligned. The transformation matrix M is calculated based on the positions of the fiducial markers:

$$\mathbf{M} = \text{cv2.getPerspectiveTransform}(\text{src\_points}, \text{dst\_points})$$

$$\text{Warped\_Image} = \text{cv2.warpPerspective}(\text{Image}, \mathbf{M}, \text{Target\_Size})$$

3.3 Segmenting Patches and Extracting RGB Values

After the image is warped, the patches are segmented, and the mean RGB values are extracted for each patch. The segmentation is based on the calculated positions of the patches:

$$\text{Top\_Left\_X} = c \times (\text{Patch\_Width} + \text{Padding}) + \text{Marker\_Size} + \text{Margin}$$

$$\text{Top\_Left\_Y} = r \times (\text{Patch\_Height} + \text{Padding}) + \text{Marker\_Size} + \text{Margin}$$

$$\text{Bottom\_Right\_X} = \text{Top\_Left\_X} + \text{Patch\_Width} - 2 \times \text{Margin}$$

$$\text{Bottom\_Right\_Y} = \text{Top\_Left\_Y} + \text{Patch\_Height} - 2 \times \text{Margin}$$

The mean RGB values for each patch are calculated as follows:

$$R_{\text{mean}} = \frac{1}{n}\sum_{i=1}^{n} R_i, \quad G_{\text{mean}} = \frac{1}{n}\sum_{i=1}^{n} G_i, \quad B_{\text{mean}} = \frac{1}{n}\sum_{i=1}^{n} B_i$$

where n is the number of pixels in the patch.

4 Conclusion

A method for generating a color test target from LAB values and extracting RGB values from a scanned image has been presented. The processes involve color space transformations, perspective corrections, and statistical averaging to ensure accurate color representation and analysis.