CCM

CCM is usually used as an approximation for the transformation from the linearized input color space to the linear absolute color space during color correction.

The shape of CCM is usually 3×3 or 4×3. The former performs linear transformation on color values, while the latter performs affine transformation. In other words. The color space keeps the origin unchanged after the linear transformation, while the latter can be translated. It can be seen that the transform set of 3×3 CCM is the proper subset of 4×3, which means that the solution set fitted by 4×3 CCM is larger. However, the latest papers prefer to use 3×3 CCM.

3×3 CCM

Fitting and Inference

The input may be a list of colors when fitting while an image when inference. Take input as a two-dimensional matrix at first. Assume that the linearized input colors is:

$$S_l = \begin{bmatrix} R_{l1} & G_{l1} & B_{l1} \\ R_{l2} & G_{l2} & B_{l2} \\ \ldots & \ldots & \ldots \end{bmatrix}$$

The output colors of the linear absolute color space is:

$$D_l = \begin{bmatrix} R'_{l1} & G'_{l1} & B'_{l1} \\ R'_{l2} & G'_{l2} & B'_{l2} \\ \ldots & \ldots & \ldots \end{bmatrix}$$

Then we have:

$$D_l = S_l \times M_{CCM}$$

In the above formula, the multiplication symbol is purposely not omitted. This multiplication is the same as the matrix multiplication, and could be implemented by using the numpy.dot, numpy.matmul or the symbol @ in numpy. In addition, the CCM can be expressed as:

$$M_{CCM} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

If the input is a multi-order tensor, it could be expressed as:

$$T_D = T_S \times M_{CCM}$$

Here, the multiplication is extended matrix multiplication, and may still be implemented by using the numpy.dot, numpy.matmul or the symbol @. In this case, the results of numpy.dot and numpy.matmul are consistent.

Initial Value

The program provides two methods.

One is white balance method[1]. The initial value is:

$$M_{CCM} = \begin{bmatrix} k_R & 0 & 0 \\ 0 & k_G & 0 \\ 0 & 0 & k_B \end{bmatrix}$$

where

$$\begin{aligned} k_R &= mean(R'_{li})/mean(R_{li}) \\ k_R &= mean(G'_{li})/mean(G_{li}) \\ k_R &= mean(B'_{li})/mean(B_{li}) \end{aligned}$$

The second is the least square method. That is an optimal solution under the linear RGB distance function, or expressed as:

$$M_{CCM} = (S_l^T S_l)^{-1} S_l^T D_l$$

This could be implemented by numpy.linalg.lstsq in numpy, expressed here as:

$$M_{CCM} = ls(S_l, D_l)$$

If there are weights:

$$w = [w_1, w_2, \ldots]$$

Remark:

$$W = \begin{bmatrix} \sqrt{w_1} & 0 & \ldots \\ 0 & \sqrt{w_2} & \ldots \\ \ldots & \ldots & \ldots \end{bmatrix}$$

Then we have:

$$M_{CCM} = ls(W \times S_l, W \times D_l)$$

Inverse

When evaluating the model, the original image of the color before the CCM is required. For a 3x3 CCM, the inverse is expected to exist. We can calculate the original image by the inverse of CCM.

$$S_l = D_l M_{CCM}^{-1}$$

4×3 CCM matrix

Fitting and Inference

4×3 CCM can be expressed as:

$$M_{CCM} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix}$$

Remark:

$$S_l^+ = [\, S_l \quad 1_{col} \,] = \begin{bmatrix} R_{l1} & G_{l1} & B_{l1} & 1 \\ R_{l2} & G_{l2} & B_{l2} & 1 \\ \ldots & \ldots & \ldots & \ldots \end{bmatrix}$$

Then we have:

$$D_l = S_l^+ \times M_{CCM}$$

For multi-order tensors:

$$T_D = T_S^+ \times M_{CCM}$$

Initial Value

The program provides two methods.

One is white balance method[1]. The initial value is:

$$M_{CCM} = \begin{bmatrix} k_R & 0 & 0 \\ 0 & k_G & 0 \\ 0 & 0 & k_B \\ 0 & 0 & 0 \end{bmatrix}$$

where

$$\begin{aligned} k_R &= mean(R'_{li})/mean(R_{li}) \\ k_R &= mean(G'_{li})/mean(G_{li}) \\ k_R &= mean(B'_{li})/mean(B_{li}) \end{aligned}$$

The second is the least square method. That is an optimal solution under the linear RGB distance function, or expressed as:

$$M_{CCM} = (S_l^{+T} S_l^+)^{-1} S_l^{+T} D_l$$

This could be implemented by numpy.linalg.lstsq in numpy, expressed here as:

$$M_{CCM} = ls(S_l^+, D_l)$$

If there are weights:

$$w = [w_1, w_2, \dots]$$

Remark:

$$W = \begin{bmatrix} \sqrt{w_1} & 0 & \dots \\ 0 & \sqrt{w_2} & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

Then we have:

$$M_{CCM} = ls(W \times S_l^+, W \times D_l)$$

Inverse

When evaluating the model, the original image of the color before the CCM is required.

Since:

$$D_l = S_l^+ M_{CCM} = [\, S_l \quad 1_{col} \,] \begin{bmatrix} Up_{(3,3)} \\ Down_{(1,3)} \end{bmatrix} = S_l Up_{(3,3)} + 1_{col} Down_{(1,3)}$$

Therefore:

$$S_l = (D_l - 1_{col} Down_{(1,3)}) Up_{(3,3)}^{-1}$$

It can also be equivalently expressed as:

$$S_l = \begin{bmatrix} D_l & 1_{col} \end{bmatrix} \begin{bmatrix} Up^{-1}_{(3,3)} \\ -Down_{(1,3)} Up^{-1}_{(3,3)} \end{bmatrix}$$

References

1. https://www.imatest.com/docs/colormatrix/
2. http://www.its.bldrdoc.gov/pub/ntia-rpt/04-406/