CHAPTER 3

OVERLAPPING TECHNIQUE

APPROACHING THE COEFFICIENT OF TRANSFORMATION

3.1 Schematic of Typical Model Relation

Typically, characterization is a process to define relationship between input signal and output signal of an unknown model. Figure 3-1 shows a schematic of a typical relationship between *Input Signal (X_i) and Output Signal (Y_i)* of *Real Model (Ω)*. There are two approaches to characterize the model: *Forward Characterization* and *Backward Characterization*. The former usually uses for an input device in Color Management System and the latter for an output device.

For the output device, an *Input Signal (X_i)* is *Device Color Space* or *Dependent Color Space*, and an *Output Signal (Y_i)* is *System Color Space* or *Independent Color Space*.

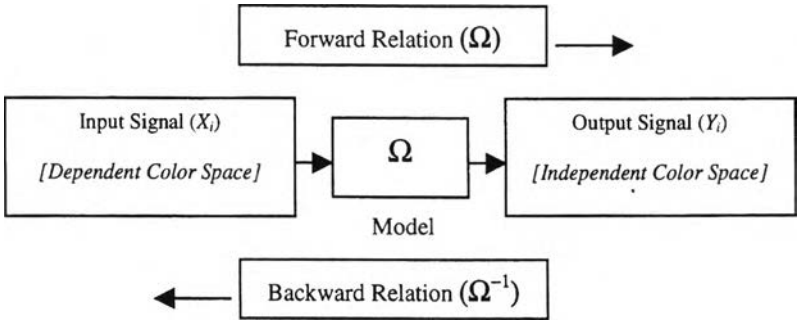

Figure 3-1 The schematic diagram of typical relationship of Input and Output Signal.

3.2 Method of Characterization with Least Square Matrices

Basically, there are three steps to characterize the unknown model by Least Square Method.

3.2.1 Look-up Table (LUT)

Look-up table or LUT is a table that consists of samples of input signals and output signals. Usually, this table is obtained from a physical pass of some input signals through a *Real Model* and a measurement of corresponding output signals. Thus, it always be a *Forward Characterization* to create the LUT.

If the LUT containes all of the possible signals, it could exactly represent the *Real Model.* However, it usually has only a few samples of signals because its large number is impossible to obtain, 16.7 million color signals in case of printer model. The samples obtained from sampling of total signals become *LUT data.*

The *LUT data* can be written in mathematic expression as follow:

$$[Y]_i = \varphi.[X]_i \qquad (3\text{-}1)$$

where; φ *is the Forward Function represents the relationship of LUT data of the Real Model.*

X *is the input signal and independent variable of* φ.

Y *is the output signal that corresponds to the input signal, It is a function of* φ *and dependent variable.*

i = *1,2,...,n, represents n sample of input signal that uses to characterize the Real Model* (Ω).

3.2.2 Inverse LUT

Coefficients of an approximated model are obtained from *LUT data* by least square method. These coefficients use to predict the input signal for the model to retrieve the expected output signals and vice versa. In this case, the *Backward Characterization* is interested, then the *Forward LUT data* will be converted to be *Backward LUT data.*

The relation of the *Backward LUT data* can be written in mathematic expression as follow

$$[X]_i = \varphi^{-1}.[Y]_i \qquad (3\text{-}2)$$

where; φ^{-1} *is Backward Function represents the relationship of the Backward LUT data of the Real Model.*

Then for this *Backward LUT data*, the *Output Signal* $[Y]_i$ become input signal and the *Input Signal* $[X]_i$ become output signal for the *Backward Characterization*. If each pair of input signal and output signal composes of three parameters, then it can be written in mathematic expression as follow:

$$[X_1, X_2, X_3]_i = \varphi^{-1}.[Y_1, Y_2, Y_3]_i \qquad (3\text{-}3)$$

3.2.3 The Multiple Linear Regression Model

The regression method uses *LUT data* to calculate the coefficients that approach the *Real Model (Ω).* In this case, the *Backward Characterization* (φ^{-1}) links the *Input Signal* $[Y]_i$ to the *Output Signal* $[X]_i$, each has three parameters. Thus, the model can be assumed as follow:

$$\begin{aligned} X_1 &= a_0 + a_1 R_1 + a_2 R_2 + a_3 R_3 + \ldots + a_i R_i \\ X_2 &= b_0 + b_1 R_1 + b_2 R_2 + b_3 R_3 + \ldots + b_i R_i \\ X_3 &= c_0 + c_1 R_1 + c_2 R_2 + c_3 R_3 + \ldots + c_i R_i \end{aligned} \qquad (3\text{-}4)$$

This can be written in matrix form as

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & \ldots & a_n \\ b_0 & b_1 & b_2 & \ldots & b_n \\ c_0 & c_1 & c_2 & \ldots & c_n \end{bmatrix} . \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \ldots \\ R_n \end{bmatrix} \qquad (3\text{-}5)$$

where; *a, b, c* are the coefficients of each parameter.

i = *0,1,2 ,...,k, represents k different independent parameter.*

n = *0,1,2 ,...,m, represents m polynomial terms that less than the number of data in LUT.*

R is a function of $[Y]_i$ that can be a combination of three parameters as follow:

$$R_1=Y_1$$

$$R_2=Y_2$$

$$R_3=Y_3$$

$$R_4=Y_1 .Y_2$$

$$R_5=Y_2 .Y_3$$

$$R_6=Y_3 .Y_1$$

$$R_7=(Y_1)^2$$

$$R_8=(Y_2)^2$$

$$R_9=(Y_3)^2$$

$$R_{10}=Y_1 .Y_2 .Y_3$$

$$R_{11}=(Y_1)^3$$

$$R_{12}=(Y_2)^3$$

$$R_{13}=(Y_3)^3$$

$$R_{14}=(Y_1)^2 .Y_2$$

$$R_{15}=(Y_2)^2 .Y_3$$

$$R_{16}=(Y_3)^2 .Y_1$$

$$R_{17}= Y_1 .(Y_2)^2$$

$$R_{18}= Y_2 .(Y_3)^2$$

$$R_{19}= Y_3 .(Y_1)^2$$

Finally, the coefficient metrics are derived. And it can be written as follow:

$$[X_1, X_2, X_3]_i = \theta.[Y_1, Y_2, Y_3]_i \qquad (3\text{-}6)$$

or

$$[X]_i = \theta.[Y]_i \qquad (3\text{-}7)$$

where; θ is *Regression Model* obtained by least square method with *Backward LUT data.*

It is *Backward Relationship* that approximates the *Real Model* (Ω^{-1}). And can be written by mathematic expression as:

$$\theta = \begin{bmatrix} a_0 & a_1 & a_2 & \dots & a_n \\ b_0 & b_1 & b_2 & \dots & b_n \\ c_0 & c_1 & c_2 & \dots & c_n \end{bmatrix} \qquad (3\text{-}8)$$

3.3 Tetrahedral Partition Method

Another constituent is number of data, or *LUT data size*, that uses to derive the coefficient matrix. There is some difference of accuracy, to approximate the result, between the use of coefficient matrix obtained from the whole data and the portion of data around the interested point. In geometric viewpoint, this is the method of dividing data into *Sub-divided color space*. The coefficient matrices can be derived

from each sub-space because the data can arrange itself as axis to form a space. Hence, there are several alternatives to partition the data space.

Tetrahedral method is a technique to partition a cube into six tetrahedrons; each has a triangle base as shown in Figure 3-2.

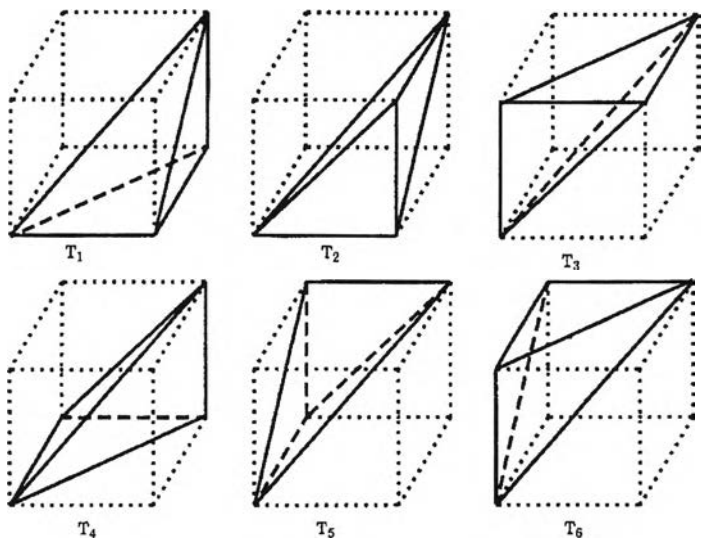

Figure 3-2 Tetrahedral partition method divide a cube into six sub-spaces.

For the three parameter cube color space, such as *sRGB* color space, It could be easily defined a boundary partition of an interesting point. Just compare each parameter value itself as show below:

Section T_1 R>G>B

Section T_2 R>B>G

Section T_3 G>R>B

Section T_4 G>B>R

Section T_5 B>R>G

Section T_6 B>G>R

3.4 The propose method

The proposed method is a combination of *Regression model* and *Tetrahedral partition* technique. There are two major alternatives to make partition: 1) to create a partition based on the knots point, 2) to apply any geometric form into the color space regardless of the knot point.

The first alternative can be a simplex*[13]* if the partition is base on 4 knots in three dimension space which is the smallest partition it could be. This method usually gives the outstanding approximated result for the maximum error entire the space for typical algorithm. However, the search engine to find the adjacent knots of each interested point is expensive.

The second alternative is just apply any geometric forms into a color space. This will geometrically divide the *LUT data* into sub-spaces. Thus, It is important to realize that each partition might consist of uncertain number of *LUT data* and, moreover, the position of each *LUT data* in the partition.

The tetrahedral partition technique chosen here is the latter approach. It applies to the *sRGB* color space followed by the plane of $R+G+B=383$, whereby are obtained 12 partitions. Finally, each sub-divided color space is applied by regression model to retrieve the coefficient of transformation.

Before going further to the experiment, it is necessary to categorize some kinds of partition used in this work. The following are some terminologies that use in the propose thecnique.

3.4.1 Coefficient Partition

Coefficient Partition is a portion of *LUT data*, which is graphically grouped by a partition method. This portion of *LUT data* uses for deriving its coefficient. Figure 3-3 shows a graphically tetrahedral partition in *sRGB* color space.

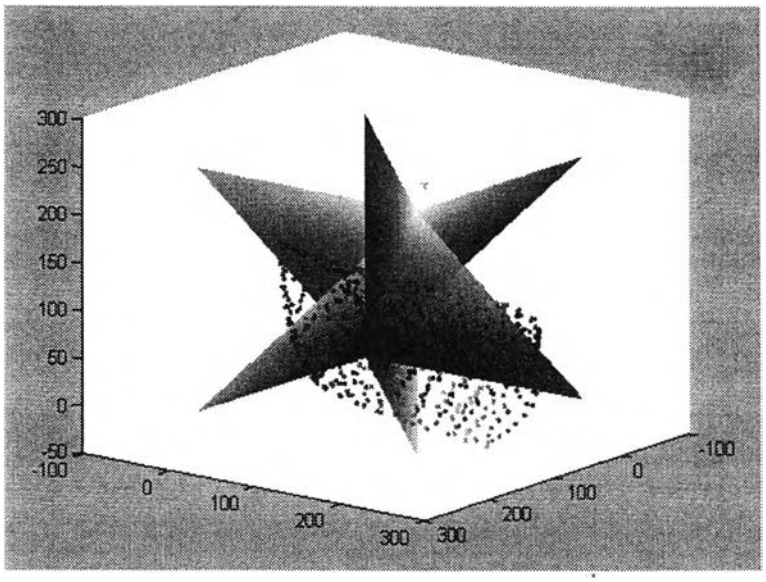

Figure 3-3 Tetrahedral partition in *sRGB* color space.

3.4.2 Target Partition

Target Partition is a boundary which locates each input signal to approximate the result by corresponding coefficient partition. Typically, this partition has the same boundary as the *Coefficient Partition*.

3.4.3 Overlapping Partition

Overlapping Partition is a method of partitioning color space to be *Coefficient Partition*, which allow each partition overlap each other. Basically, each *Coefficient Partition* has the same boundary as each corresponding *Target Partition*. However, It doesn't practical to approach by this method. Because some of input signal might be out side the boundary of *LUT data*.

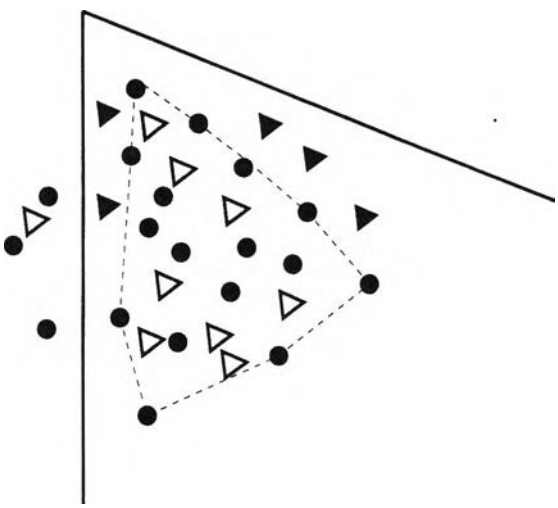

Figure 3-4 The *Input Signals* $[Y]_i$ which are out side of the *LUT data boundary*.

In Figure 3-4, the black thick line is the *Coefficient Partition* and dash line is *LUT data boundary*, the black circle points represent *LUT data*, the white triangle is the *Input Signal* $[Y]_i$, the black triangle is *Input Signal* $[Y]_i$ which are out of *LUT data boundary*.

In this case, if the *Target Partition* has the same boundary as *Coefficient Partition*, some of the *Input Signal* $[Y]_i$ might be out of *the LUT data boundary*. The approximated value may be huge of error outside *LUT data boundary* due to the characteristic of regression model that it is not suitable for predicting an

outer range value. So, it's better to extract the *Coefficient Partition*, at least, to cover the *Target Partition*. Figure 3-5 shows an extracted *Coefficient Partition*.

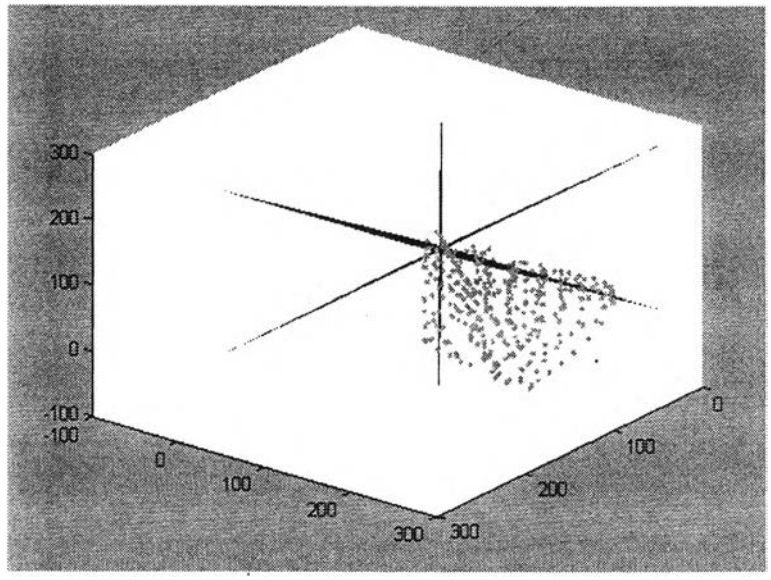

Figure 3-5 A partition, which is extracted to cover corresponding target partition.

3.5 The categories of the Testing Data

An evaluation method uses the amount of input signal to be *a Testing data*$[T]_i$. The testing signal can be divided into two categories; 1). Uniformly sampling of signal testing data defined as T_{uni}. Figure 3-6 shows an example of T_{uni} signal of printer *Canon* model *BJC8500*, which only exists in its *Gamut boundary*. 2). Domain of signal data in *Backward LUT data*, as shown in Figure 3-7, defined as T_{lut}.

The Latter can be used as a *Testing data*, because the nature of least square algorithm that each *LUT data* has an error from the regression line. And the combination of these is global signal testing data defined as T_{glo}.

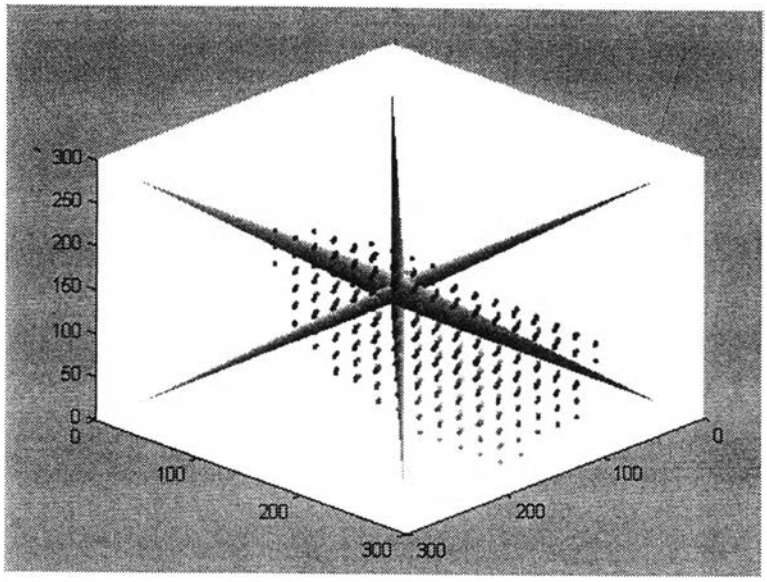

Figure 3-6 Testing data T_{uni} uniformly distribute around printer gamut.

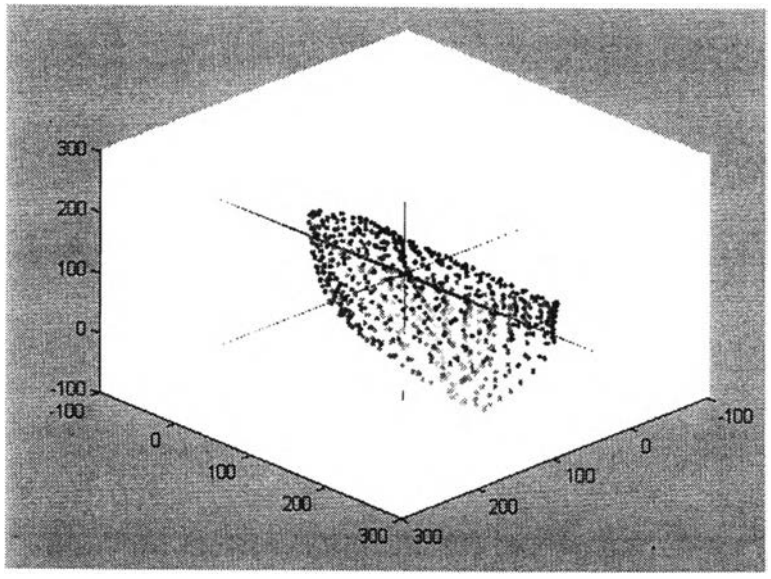

Figure 3-7 Testing data T_{lut} un-uniformly distribute around printer gamut.

3.6 Evaluation of Algorithm

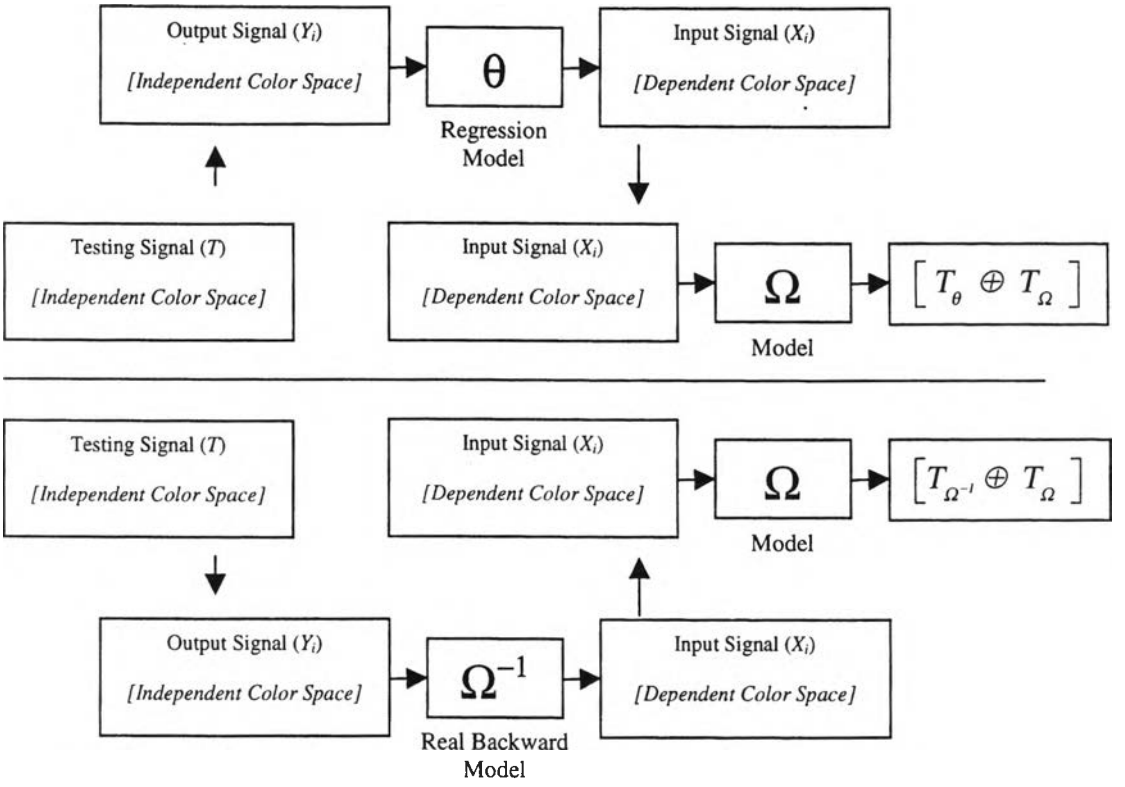

Figure 3-8 The schematic diagram of the testing algorithm process.

The schematic diagram of the testing algorithm process shown in Figure 3-8. First, the *Input Signal* $[X]_i$ obtained from many kinds of *Testing data* by backward simulated model *(θ)*. Then these *Input Signals* $[X]_i$ are passed through the *Real Model (Ω)* to get the *Output Signals* $[Y]_i$

Finally, The evaluating process is done by comparing the *Output Signal* $[Y]_i$ with *Testing data* $[T_i]$ by following relation.

$$\Delta E = \left[T_{\Omega^{-1}} \oplus T_\Omega\right] - \left[T_\Omega \oplus T_\theta\right] \qquad (3\text{-}9)$$

where; $\left[T_{\Omega^{-1}} \oplus T_{\Omega}\right]$

are the testing input signal that is passed through the inverse of *Real Model* (Ω^{-1}) and *Real Model* (Ω). Therefore, the output testing signal is exactly same as input signal.

$\left[T_{\theta} \oplus T_{\Omega}\right]$

are the testing input signal that is passed through the inverse of approximated model (θ) and real model (Ω). Therefore, the output testing signal is not exactly same as input signal.

ΔE *is the difference of both result signals in Euclidean Space.*

$\oplus$ *is the operand that stand for the real process of each model.*

The criterion to judge the algorithm is the difference of distance in Euclidean Space or color difference (ΔE) of each testing signal compared in *LAB* color space. The color difference (ΔE) can be obtained from the following equation:

$$\Delta E_i = || (\Delta L_i)^2 + (\Delta a_i)^2 + (\Delta b_i)^2 ||^{1/2} \qquad (3\text{-}10)$$

where; ΔE *is the difference of color in Euclidean Space.*

ΔL *is the difference of lightness in L*a*b* Color Space.*

Δa *is the difference of chromatic component a* in L*a*b* Color Space.*

Δb *is the difference of chromatic component b* in L*a*b* Color Space.*

i = *1,2,...,n, represents n sample of Testing Signal that uses to evaluate the algorithm performance.*

Then, the approximated accuracy of the algorithm is evaluated by statistical methods. They include Arithmetic Mean ($\overline{\Delta E}$), Root Mean Square (*RMS*) and Maximum of Color Different (L_∞).

The first one performs a comparison in a simple way. As shown in below:

$$\overline{\Delta E} = \frac{1}{n}\sum_{i=1}^{n}\|\Delta E_i\| \qquad (3\text{-}11)$$

The second one includes a meaning of Standard Deviation together with Arithmetic Mean, so typically the value of *RMS* is greater than the Arithmetic Mean.

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\|\Delta E_i\|^2} \qquad (3\text{-}12)$$

The last one judges the algorithm whether it's suitable for some kinds of work which special colors play an important role such as logo color etc.

$$L_\infty = \max_{i=1,\ldots,n}\|\Delta E_i\| \qquad (3\text{-}13)$$