# Efficient Implementation of Nonlinear Color Transformations

*Jan P. Allebach, James Z. Chang and Charles A. Bouman*
*School of Electrical Engineering, Purdue University*
*West Lafayette, Indiana*

## Abstract

We introduce a new approach which we call sequential linear interpolation (SLI) for interpolating multidimensional nonlinear functions. SLI grid points can be nonuniformly allocated to minimize the interpolation error. Thus, we use the grid points more efficiently, there- by reducing the number of required measurements. An iterative procedure is also proposed to solve the color printer calibration problem where the function to be interpolated must be measured from its inverse point by point.

## 1. Introduction

Due to the different color spaces native to different color devices,[1, 2] nonlinear color transforms are needed to convert between color spaces. In this paper, we propose an efficient way to implement these nonlinear transforms. We specifically use the color printer calibration problem to illustrate our new method, but our general technique can also be applied to other types of nonlinear color transforms, such as those described in [3].

For color printers, the colors are specified in a CMY color space to control the amount of the colorants to be put on paper. But for other devices such as CRT color monitors, colors are specified in an RGB space to control the excitation of the three primary phosphors. In order to make different color devices communicate efficiently, Gentile[4] described a scheme to specify color in device independent space. When the image is sent to a specific color printer, a series of transformations are then performed by the print-er driver to render colors in the printer colorant (CMY) space. The last stage of this rendering process is a printer specific multidimensional nonlinear transform implemented by interpolative lookup-tables (LUTs).

In this paper, we will study an efficient way to implement the interpolative LUTs for nonlinear transforms. Other aspects of the color printer calibration problem, such as the gamut matching problem[5] and the relation between the CMY and CMYK colorant spaces,[6, 7] are not discussed. The work of implementing this efficient interpolation algorithm to actually calibrate our color printers is currently underway. We also plan to incorporate this method into the Apple ColorSync color management system to generate the printer profiles as described in[8] more efficiently.

## 2. Motivation and Overview

The computational model we will study in this paper is depicted in Fig. 1. Here we assume that the input image is in the CIE L*a*b* space. The 3-D vector $\mathbf{x}$ represents the L*a*b* values for this image. They are transformed to CMY values $\mathbf{y}$ by the function $g(\mathbf{x})$ with domain $\Omega$, which is the printer output gamut. Finally, the function $f(\mathbf{y})$ represents the transformation from printer CMY to L*a*b* values $\mathbf{z}$ measured from the printer output. Our objective is to find an efficient interpolative LUT scheme to implement the nonlinear function $g(\mathbf{x})$ so that the mean

squared error (MSE) between vectors **x** and **z** is minimized given the number of interpolation grid points. That is, we need to minimize

$$\mathcal{E} = \int_{\Omega} \|\mathrm{x} - \mathrm{z}\|^2 d\mathrm{x}. \quad (1)$$

If the color transform $g(\mathbf{x})$ that is implemented were the exact inverse of the printer transfer function $f(\mathbf{y})$, there would be no error between the input **x** and the measured output **z** for all input colors **x** that are in the gamut of the printer. Therefore, $g(\mathbf{x})$ is an approximation to the inverse printer transfer function $f^{-1}(\mathbf{x})$ which is implemented by interpolative LUTs.

The conventional method[7] for implementing 3-D nonlinear color transforms is trilinear interpolation on regular grid points as depicted in Fig. 2. However, in the color printer calibration problem, we can only control the selection of input points in the CMY colorant space; therefore the placement of the measured output points in CIE L*a*b* space is irregular. To implement the regular grid trilinear interpolation, we must first generate a regular grid in the measured CIE L*a*b* space. A method called tetrahe- dral linear interpolation[7, 9] is used to interpolate points in the CMY space corresponding to regular grid points in the L*a*b* space. These regular grid points may then be used for trilinear interpolation of the inverse printer transfer function.

There are two disadvantages to this method. First, it does not consider the optimal placement of the grid points. Some regions might require more grid points than others due to the characteristics of the function being interpolated. Second, this method does not select the measured points to minimize the distance to the desired regular grid locations. The overall result is the inefficient use of the measured information.

In this paper, an efficient multidimensional interpolation scheme which we call sequential linear interpolation (SLI) is proposed. This scheme optimally allocates grid points according to the characteristics of the function. Since the function's characteristics are not known in advance, we propose a design procedure which iteratively estimates the function's characteristics and constructs the SLI structure with increasing accuracy. This procedure is depicted in Fig. 3. An initial set of printer measurements is made first. We then optimally construct an initial SLI structure so that the distance between the measured output points and the grid locations is minimized. We use the sequential scalar quantization (SSQ) method[10, 11] to initialize this structure. Next, we estimate the necessary characteristics of the inverse printer transfer function. These estimates are then used to guide us in selecting new printer measurement points so that the MSE in (1) is minimized. Since the new measurement points are approximately estimated by the initial SLI structure, the actual measured output points in the CIE L*a*b* space will not exactly have the desired SLI structure. Therefore, we need to perform SSQ on the output data again to obtain a new SLI structure based on the new measurement points. This process can be continued iteratively with an increasing number of measurement points to obtain more accurate SLI structures.

In order to obtain the optimal allocation of grid points in SLI, we apply results from an asymptotic analysis. The assumption for the asymptotic analysis is that the number of grid points is large and the function to be interpolated is smooth. In previous applications,[12] we have found that the results of an asymptotic analysis of this type are generally valid, even when the underlying assumptions are not well satisfied.

In the next section, we introduce the details of the SLI structure. In Sec. 4, we analyze the error criterion for the printer calibration problem. This is important to the optimal allocation of interpolation grid points. In Sec. 5, we briefly describe the asymptotic analysis used to obtain the optimal grid point allocation. The details of the asymptotic analysis and the iterative design procedure are presented in.[13]

## 3. Sequential Linear Interpolation

We first describe the SLI structure for a 2-D scalar-valued function. If the function is vector-valued, we may treat it as several scalar-valued functions. These functions can be independently interpolated by SLIs. Let $y = f(\mathbf{x})$ be a scalar-valued 2-D function, where $\mathbf{x} = (x_1, x_2)$ is a vector in $\Re^2$. We further assume that the domain of $f(\mathbf{x})$ is bounded. To sequentially allocate grid points to the domain of $f(\mathbf{x})$, we first place $n$ grid points, $x_{11}, x_{12}, \ldots, x_{1n}$, onto the $x_1$ axis as shown in Fig. 4 for $n = 5$. These grid points can be placed nonuniformly according to the characteristics of the function being interpolated. The grid points on the $x_1$ axis correspond to $n$ grid lines in the 2-D domain. As also shown in Fig. 4, we further place $n_i$ grid points on the $i$-th grid line for $i = 1, 2, \ldots, n$, according to the characteristics of the function. Suppose $N$ is the total number of grid points, then the $n_i$ must satisfy the constraint

$$\sum_{i=1}^{n} n_i = N. \quad (2)$$

The output $y$'s at these grid points are stored for interpolation.

To find the approximation $y \approx f(\mathbf{x})$ at an arbitrary point $\mathbf{x} = (x_1, x_2)$, **x** is first projected onto the two adjacent grid lines at $x_{1l}$ and $x_{1r}$, respectively, as depicted in Fig. 5a. These projection points are $\mathbf{x}_l$ and $\mathbf{x}_r$. Linear interpolations are performed along the grid lines at $x_{1l}$ and $x_{1r}$, to obtain function outputs $y_l$ and $y_r$ at $\mathbf{x}_l$ and $\mathbf{x}_r$, respectively. The values $y_l$ and $y_r$ are then used to linearly interpolate the value $y$ at **x**. If as depicted in Fig. 5b, **x** does not lie bet- ween two grid lines, the value $y$ is extrapolated from the projection points on the two nearest grid lines. The implementation of the sequential linear interpolation process is depicted in Fig. 6a. The linear interpolations on the grid lines can be implemented by a pair of interpolative LUTs, which are the "ILUT" blocks in Fig. 6a. The projection step can be implemented by passing the first component $x_1$ into a LUT to obtain the indices $l$ and $r$ which identify the grid lines at $x_{1l}$ and $x_{1r}$. These indices and $x_2$ are the inputs to the interpolative LUTs which compute the interpolated values $y_l$ and $y_r$. These outputs are then linearly combined to obtain $y$. That is

$$y = w_l y_l + w_r y_r, \quad (3)$$

where the coefficients

$$w_r = \frac{x_1 - x_{1l}}{x_{1r} - x_{1l}} \tag{4}$$

and

$$w_l = 1 - w_r \tag{5}$$

are also obtained from the first LUT. If the 2-D SLI structure is used as a module in a larger structure (e.g. a 3-D SLI structure), then an optional index input $i$ is also required.

The SLI discussed above may be generalized to 3-D functions. For a scalar-valued 3-D function $y = f(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, x_3)$ is in a bounded domain in $\Re^3$, we first place $n$ grid points onto the $x_1$ axis. These grid points will generate $n$ grid planes in the $x_2$ - $x_3$ space. On each of these grid planes, we construct a 2-D SLI structure as previously described. To obtain the output value for a certain point **x,** we first project it onto the two adjacent grid planes. The projection process is implemented by a LUT, as depicted in Fig. 6b. The function values at the projected points on those grid planes can be obtained by the 2-D SLI modules implemented on those grid planes. These values are then linearly combined to obtain the desired output using the weights generated by the first LUT. An optional index input $i$ is also shown for use in a higher dimensional SLI structure. From the discussion here, we see that this meth-od can be readily generalized to higher dimensional functions.

Comparing the sequential interpolation grid in Fig. 4 with the regular interpolation grid in Fig. 2, we see that the regular interpolation grid is a special case of the sequen- tial interpolation grid when the grid lines and the grid points are distributed uniformly and the domain is regular. As is in the case with the regular interpolation grid, the sequential interpolation scheme will produce a continuous function, since all the l-D interpolations are continuous. Figure 4 shows how the sequential interpolation grid automatically tracks the domain of the function *(i.e.* the printer gamut) by placing all the grid points inside the domain. If the domain of the function is not rectangular, then a regular grid will waste grid points outside the domain, as indicated in Fig. 2. More importantly, when the number of grid points is limited, the sequential interpolation grid allows us to arbitrarily allocate the grid lines and points to minimize the interpolation error.

## 4. Error Analysis for Color Printer Calibration

In this section, we first discuss the error criterion for the printer calibration problem. Then we show how this error may be decomposed on the SLI grid. For the purposes of the asymptotic analysis,[13] we assume here that the number of grid points is large and the functions are smooth enough to permit a first order approximation.

### 4.1 Error Criterion for Color Printer Calibration

Our model for the color transforms involved in printing is illustrated in Fig. 1. The vector-valued color transformation function $g(\boldsymbol{x})$ and printer transfer function $f(\mathbf{y})$ can be decomposed into three scalar-valued functions,

$$\mathbf{y} = g(\mathrm{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), g_3(\mathbf{x})), \tag{6}$$

and

$$\mathbf{z} = f(\mathrm{y}) = (f_1(\mathrm{y}), f_2(\mathrm{y}), f_3(\mathrm{y})). \tag{7}$$

If $g(\mathrm{x}) = f^{-1}(\mathbf{x})$, then $\mathbf{z} = f(g(\mathrm{x})) = \mathbf{x}$ and there is no error in the printed image. Therefore, the output of the approximate transformation $g(\mathbf{x})$ can be represented by

$$\mathrm{y} = f^{-1}(\mathrm{x}) + \Delta\,\mathrm{y}\,(\mathrm{x}), \tag{8}$$

where

$$\Delta\mathrm{y}\,(\mathrm{x}) = g\,(\mathrm{x}) - f^{-1}(\mathrm{x}), \tag{9}$$

is the error vector introduced by the implementation of $g(\mathrm{x})$. Writing (7) and (8) componentwise and applying a first order approximation, we have

$$\begin{aligned} z_i &= f_i(\mathrm{y}) \\ &= f_i(f^{-1}(\mathrm{x}) + \Delta\mathrm{y}(\mathrm{x})) \\ &\approx x_i + \sum_{j=1}^{3} \frac{\partial f_i(\mathrm{y})}{\partial y_j} \Delta y_j(\mathrm{x}). \end{aligned} \tag{10}$$

Then, the error in the $i$-th component may be expressed as

$$\begin{aligned} &(x_i - z_i)^2 \\ &= \left( \sum_{j=1}^{3} \frac{\partial f_i(\mathrm{y})}{\partial y_i} \Delta y_j(\mathrm{x}) \right)^2 \\ &= \sum_{j=1}^{3} \sum_{k=1}^{3} \frac{\partial f_i(\mathrm{y})}{\partial y_j} \frac{\partial f_i(\mathrm{y})}{\partial y_k} \Delta y_j(\mathrm{x}) \Delta y_k(\mathrm{x}), \end{aligned} \tag{11}$$

which is composed of self-multiplication and cross-multiplication terms. Since the cross-multiplication terms would complicate the optimization procedure, they will be dropped. This is justified under a statistical model where $\Delta y_j(\mathrm{x})$, $j$ = 1, 2, 3, are uncorrelated, since in this case the cross-multiplication terms will not contribute to the expected value of the error in (1).

The error $\tilde{\varepsilon}$ in (1) that remains can be written as a sum of the errors $E_i$ along each coordinate axis of **y**; so

$$\tilde{\varepsilon} = E_1 + E_2 + E_3, \tag{12}$$

where

$$E_i = \int_{\Omega} \left[ \sum_{j=1}^{3} \left( \frac{\partial f_i(\mathrm{y})}{\partial y_i} \right)^2 \right] \left[ \Delta y_i(\mathrm{x}) \right]^2 d\mathrm{x}. \tag{13}$$

Here $E_i$ is the error introduced by the approximation of the $i$-th scalar-valued component $f_i^{-1}(\mathrm{x})$. These terms can be minimized independently since the three component functions are implemented by three independent SLIs. However, we note that the error $E_i$ is not simply the inte-gral of the squared error $\left[\Delta y_i(\mathrm{x})\right]^2$ at the output of the $i$-th SLI. Instead, it is weighted by the factor

$$K_i(\mathrm{x}) = \sum_{j=1}^{3} \left( \frac{\partial f_i(\mathrm{y})}{\partial y_i} \right)^2, \tag{14}$$

which is dependent on **x**, since **y** is a function of **x**. It represents the sensitivity of the printer transfer function to the errors introduced in the **y** vector.

Now the total mean squared error has been decomposed into three components related to $\Delta y_i(\mathrm{x})$, $i = 1, 2, 3$, respectively. Due to the sequential nature of the grid points on **x**, each of these components can in turn be decomposed into three components related to the three axes in the **x** vector. We will describe this decomposition next.

### 4.2 Decomposition of the Error on the Sequential Interpolation Grid

Since we minimize each $E_i$, $i = 1, 2, 3$ independently, we will drop the subscript $i$ throughout the remainder of this section, in order to simplify the notation as much as possible. That is, we will minimize

$$\boldsymbol{E} = \int_{\Omega} \boldsymbol{K}(\mathrm{x})\left[\Delta \boldsymbol{y}(\mathrm{x})\right]^2 d\mathrm{x}. \tag{15}$$

where

$$\Delta \boldsymbol{y}(\mathrm{x}) = \boldsymbol{g}(\mathrm{x}) - \boldsymbol{f}^{-1}(\mathrm{x}). \tag{16}$$

From Fig. 6b, we see that the output $y = g(\mathbf{x})$ is interpolated from the values $y_l = g(\mathbf{x}_l)$ and $y_r = g(\mathbf{x}_r)$ of the two projection points $\mathbf{x}_l$ and $\mathbf{x}_r$ as shown in Fig. 5. These values are obtained by the two 2-D SLI modules. Their errors are denoted by

$$\Delta_{23}(\mathrm{x}_l) = y_l - f^{-1}(\mathrm{x}_l) \tag{17}$$

and

$$\Delta_{23}(\mathrm{x}_r) = \boldsymbol{y}_r - \boldsymbol{f}^{-1}(\mathrm{x}_r), \tag{18}$$

respectively. Here the subscript "23" denotes the fact that these errors result from interpolation of $f^{-1}(\mathrm{x})$ in the $x_2$-$x_3$ plane. Under the condition for asymptotic analysis, we assume that the distribution of the grid points is dense enough so that the interpolation errors of the two 2-D SLIs in Fig. 6b are approximately the same. This allow us to reference these errors to the coordinate **x**; so we can write

$$\Delta_{23}(\mathrm{x}) \approx \Delta_{23}(\mathrm{x}_l) \approx \Delta_{23}(\mathrm{x}_r). \tag{19}$$

Applying this to (3), we have

$$\begin{aligned} \boldsymbol{y} &= (1-\boldsymbol{w}_r)\boldsymbol{y}_l + \boldsymbol{w}_r \boldsymbol{y}_r \\ &= (1-\boldsymbol{w}_r)\boldsymbol{f}^{-1}(\mathrm{x}_l) + \boldsymbol{w}_r \boldsymbol{f}^{-1}(\mathrm{x}_r) \\ &\quad + \Delta_{23}(\mathrm{x}). \end{aligned} \tag{20}$$

Assuming that the data used for interpolation along the $x_1$ axis are accurate, the error resulting from this interpolation step may be written as

$$\begin{aligned} &\Delta_1(\mathrm{x}) = \\ &\quad \boldsymbol{f}^{-1}(\mathrm{x}) - \left[(1-\boldsymbol{w}_r)\boldsymbol{f}^{-1}(\mathrm{x}_l) + \boldsymbol{w}_r \boldsymbol{f}^{-1}(\mathrm{x}_r)\right]. \end{aligned} \tag{21}$$

Thus, we have decomposed the total error $\Delta y$ (x) as

$$\Delta y(\mathrm{x}) = \Delta_1(\mathrm{x}) + \Delta_{23}(\mathrm{x}). \tag{22}$$

We can similarly decompose $\Delta_{23}(\mathbf{x})$ into components along the $x_2$ and $x_3$ axes. Therefore, we have

$$\Delta y_i(\mathrm{x}) = \Delta_1(\mathrm{x}) + \Delta_2(\mathrm{x}) + \Delta_3(\mathrm{x}), \tag{23}$$

where $\Delta_i(\mathbf{x})$ is the error resulting from linear interpolation along the $x_i$ axis, assuming that the data used for the interpolation are accurate.

With this decomposition, (15) becomes

$$E = \int_{\Omega} K(\mathrm{x})[\Delta_1(\mathrm{x}) + \Delta_2(\mathrm{x}) + \Delta_3(\mathrm{x})]^2 d\mathrm{x}. \tag{24}$$

Following the same kind of reasoning that was applied to (11) in Sec. 4.1, we exclude the effect of the cross-multiplication terms in the expansion of the squared term above. Therefore, we seek an optimal SLI structure that minimizes

$$D = D_1 + D_2 + D_3, \tag{25}$$

where

$$D_i = \int_{\Omega} K(\mathrm{x})\left[\Delta_i(\mathrm{x})\right]^2 d\mathrm{x}. \tag{26}$$

## 5. Optimal Sequential Linear Interpolation

Using the distortion metrics of Sec. 4, it is possible to derive the optimal SLI structure as the grid points become finely spaced. However, this asymptotic analysis goes beyond the scope of this paper and is presented in [13]. The analysis in[13] is similar to that used in [11]. We analyze the three components in (25) and the results are combined appropriately to obtain the conditions for optimal grid point allocation. These conditions are expressed in terms of the asymptotic grid point density function.

These conditions depend on the first derivatives of $f$ (**y**) and the second derivatives of $f^{-1}$(**x**). However, these derivatives are not possible to obtain directly since the printer transfer function $f$(**x**) must be measured point by point. Therefore, we develop an iterative design procedure in.[13] In this procedure, we estimate the derivatives based on the existing interpolation structure, and then use these derivatives to guide us in constructing more accurate interpolation structures.

## 6. Conclusion and Discussion

In this paper, we developed a flexible linear interpolation grid structure for implementing multidimensional functions. In this structure, we can allocate the grid points nonuniformly to minimize the error in (25). The detailed analysis of the optimal grid point allocation is presented in the technical report.[13]

We plan to implement this procedure to calibrate our color monitors and printers. We will also study the effect of the number of grid points and the order of allocation on the three axes, and do experiments on using color spaces other than the CIE L*a*b* space as the input color space.

## 7. Acknowledgment

*This research was supported by Apple Computer, Inc.

We would like to acknowledge helpful discussions with John Dalton and Gary Starkweather of Apple Computer during the course of this research project.

## References


1. J. M. Kasson and W. Plouffe, "Requirements for Computer Interchange Color Spaces," *Image Communications and Workstations, Proc. SPIE*, Vol. **1258**, pp. 172-183, 1990.
2. J. M. Kasson and W. Plouffe, "Subsampled Device-independent Interchange Color Spaces," *Image Handling and Reproduction Systems Integration, Proc. SPIE*, Vol. **1460**, pp. 11-19, 1991.
3. J. Dalton, "Perceptually-based Adaptive Color Image Enhancement," *Conference Digest: IS&T's 46th Annual Conference*, pp. 146-147, Boston, MA, May 10-14, 1993.
4. R. S. Gentile, "Device Independent Color in PostScript," *Human Vision, Visual Processing, and Digital Display, Proc. SPIE*, Vol. **1913**, San Jose, CA, Jan. 3I - Feb. 4, 1993.
5. R. S. Gentile, E. Walowit, and J. P. Allebach, "A Comparison of Techniques for Color Gamut Mismatch Compensation," *Journal of Imaging Technology*, Vol. 6, No. **5**, pp. 176-181, Oct. 1990.
6. M. C. Stone, W. B. Cowan, and J. C. Beatty, "Color Gamut Mapping and the Printing of Digital Color Images," *ACM Trans. on Graphics*, Vol. **7**, No. 4, pp. 249-292, Oct. 1988.
7. S. I. Nin, J. M. Kasson, and W. Plouffe, "Printing CIELAB Images on a CMYK Printer Using Tri-linear Interpolation," *Color Hardcopy and Graphic Arts, Proc. SPIE*, Vol. **1670**, pp. 316-324, 1992.
8. J. Thornton, Y.-J. Lee, and M. Balonon-Rosen, "The Apple ColorSync Printer Profile Model and Its Optimization," *Conference Digest: IS&T's 46th Annual Conference*, pp. 147-150, Boston, MA, May 10-14, 1993.
9. P.-C. Hung, "Colorimetric Calibration in Electronic Imaging Devices Using a Look-up-table Model and Interpolations," *Journal of Electronic Imaging*, Vol. **2**(1), pp. 53-61, Jan. 1993.
10. R. Balasubramanian, C. A. Bouman, and J. P. Allebach, "Sequential Scalar Quantization of Color Images," *Conference Digest: IS&T's 46th Annual Conference*, pp. 97-101, Boston, MA, May 10-14, 1993.
11. R. Balasubramanian, C. A. Bouman, and J. P. Allebach, "Sequential Scalar Quantization of Vectors: An Analysis," submitted to the *IEEE Trans. on Image Processing.*
12. R. Balasubramanian, C. A. Bouman, and J. P. Allebach, "Sequential Scalar Quantization of Color Images," submit ted to the *Journal of Electronic Imaging.*
13. J. P. Allebach, J. Z. Chang, and C. A. Bouman, "Efficient Implementation of Nonlinear Color Transformations," *Technical Report*, School of Electrical Engineering, Purdue University, in preparation.


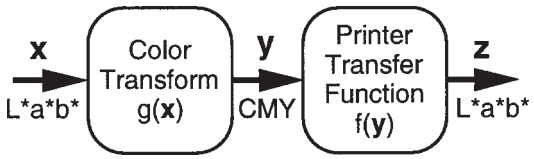


*Figure 1. Computational model*

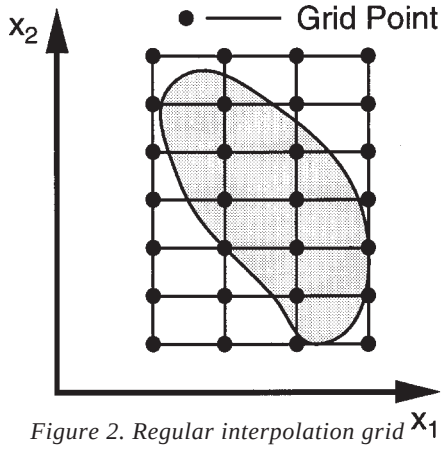


*Figure 2. Regular interpolation grid*

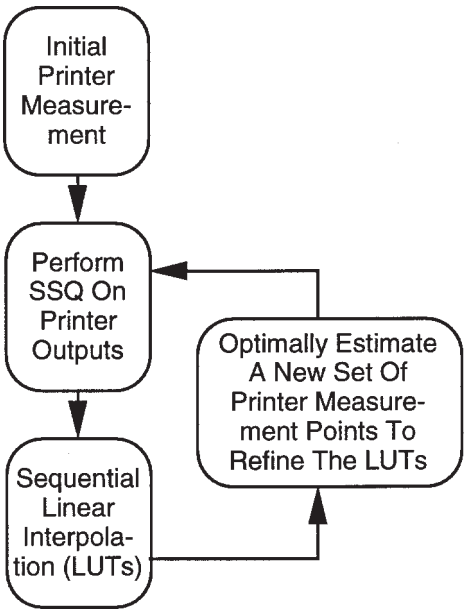


*Figure 3. Iterative design procedure*

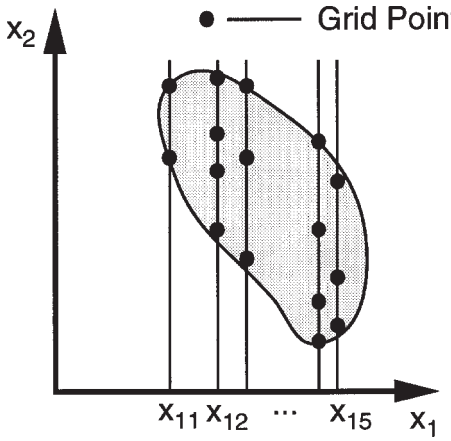


*Figure 4. Sequential interpolation grid*

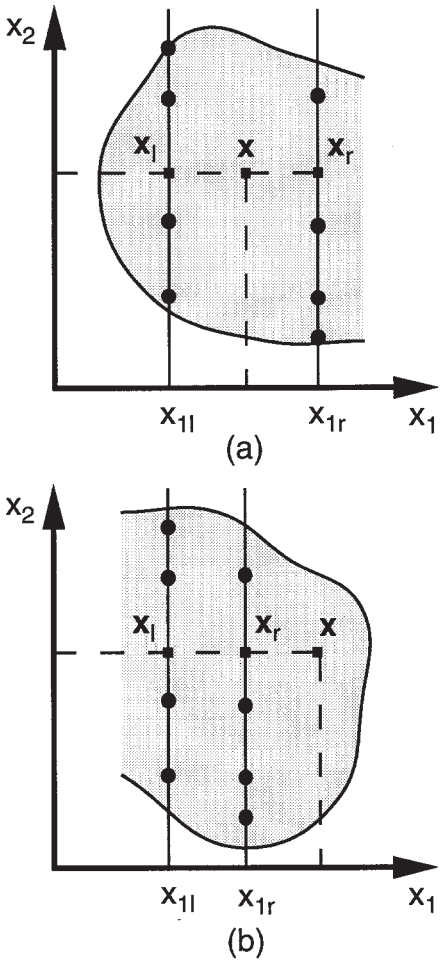


*Figure 5. Sequential interpolation scheme for (a) a point* **x** *that lies between two grid lines and (b) a point* **x** *that does not lie between two grid lines*

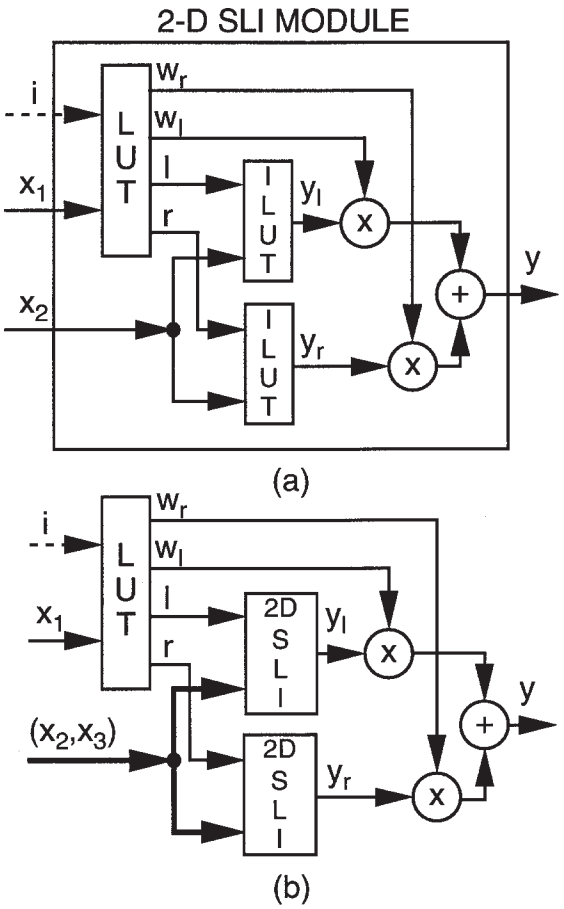


*Figure 6. LUT implementation of (a) 2-D and (b) 3-D sequential linear interpolation*