# Iteratively Clustered Interpolation (ICI) Algorithm for Geometrical Interpolation of an Irregularly Spaced Multidimensional Color Space

*L.K. Mestha, Y.R. Wang, S.A. Dianat and D.E. Viassolo*
*Xerox Corporation, Wilson Center for Research and Technology*
*Webster, New York*

## Abstract

A printer inverse map is usually represented as a multivariable lookup table, associating points in the printer's output color space with points in the printer's input color space. This lookup table is an essential component in many print quality enhancement algorithms. It is often desirable to have a printer inverse table with input nodes regularly spaced on a sequential plane. Thus, the computation of this lookup table requires the interpolation of irregularly sampled multidimensional data, coming from experiments used to determine the printer forward map. Existing computational techniques do not provide an accurate printer inverse map from irregularly sampled data. In this paper we introduce a new Iteratively Clustered Interpolation (ICI) algorithm to compute an accurate inverse table from irregularly sampled color data. This algorithm is based on a gradient optimization method with initial points generated through a novel iterative technique. Experimental results are included to show the effectiveness of this algorithm in comparison with other techniques.

## 1. Introduction

The printer inverse map is often given as a lookup table mapping points from the printer's output color space into points in the printer's input color space; i.e. L*a*b*→CMYK, XYZ→CMYK, etc. For networked printers, with no knowledge of the internals of the driver, the inverse could also be thought of as the lookup table with the map L*a*b*→L*a*b*. Measuring the printer forward transfer function between the inputs and the outputs generates the lookup table. For example, in a PostScript print path, the PostScript interpreter with colors in XYZ/L*a*b* becomes the input, and the corresponding colors as measured by a spectrophotometer (color sensor) becomes the printer output. A table with input and output colors becomes the forward transfer function. The forward transfer function is used to create the printer inverse and the associated rendering intents; e.g., colorimetric, pictorial/perceptual, saturation, pure, etc.

While measuring the forward transfer function, it is possible to structure the input data by selecting equally spaced input grid points. Although just by swapping the data of the forward transfer function we can generate the inverse, the data of the input grids for such an inverse becomes unstructured because the output grids of the forward transfer function are unstructured. Especially for colors at the boundary, this type of inverse is not well defined (resulting into multi-valued outputs). Given the unstructured lookup table with the input-output (I/O) values, our goal is to find an efficient algorithm to compute a "good" inverse printer table with structured input nodes. Multidimensional interpolation is a key to obtain such a structured table.

In our proposal we will consider the colorimetric rendering intent; i.e., to match the output L*a*b* with the input L*a*b* for colors inside the printer gamut. Also, the problem is further compounded by the requirement of keeping the colors consistent across not only one printer, but also across the whole enterprise (host of printers, Xerographic, thermal ink jet, etc., host of monitors and scanners). To do so, we need to be able to describe the color in some device independent color space. Hence we use L*a*b* as our color space standard. Also, we stress that we require the printer inverse to be a structured lookup table; i.e., uniformly sampled with grid points equally separated in the input space.

There exist different multidimensional interpolation techniques; e.g., trilinear, tetrahedral,[1] Sequential Linear Interpolation (SLI),[2] that qualify for use in generating the printer inverse lookup table. Multidimensional interpolation is also required for those colors that are not in the lookup table (i.e., not in the nodes of the lookup table). So, there are basically two kinds of applications for the interpolation algorithms: (a) for generating the basic printer inverse table (i.e., the structured table nodes) and (b) for interpolating colors that are not nodes of the printer inverse table. In this paper, we concentrate on the application (a).

Notice that the SLI technique can handle non-uniformly sampled input data whereas others require uniformly spaced mapping table. Still SLI requires the input data to be on a sequential grid; i.e., not totally unstructured.

Many techniques have been proposed for interpolating multidimensional, unstructured lookup tables. Examples are the Shepard's interpolation algorithm[3] and the Master Color Controls (MCC) algorithm.[4] The Shepard's algorithm is very time consuming and is not accurate when compared to other methods. The MCC algorithm uses a control-based technique with a multi-input multi-output linear controller. It can give zero interpolation errors (numerically) when the printer I/O pairs are used as the model, and can lead to minimal errors in the printer inverse when used directly on the printer. It suffers, however, from stability problems for input-output pairs near the boundary of the printer color gamut. The accuracy of the proposed ICI algorithm is comparable to the accuracy of the MCC, while the ICI algorithm can capture more points of the invertible region of the color space than the MCC algorithm. From the experimental evidence gathered on Xerox printers, the ICI algorithm gives excellent results when compared with the existing algorithms.

## 2. The ICI Algorithm

Before describing the actual technique, let us discuss the criteria for judging the accuracy of the algorithm. In Figure 1 we show schematically a smooth (forward) printer function *P,* with input *Laby* and output *Labz* (notations are listed at the end of the paper). A smooth inverse printer function is denoted by $P^{-1}$ with input *Labx* and output *Laby*. Once the inverse *Laby* is obtained, it is passed through the printer model to obtain *Labz*. If the inverse of the printer model is accurate, *Labz* will be "very similar" (in theory, equivalent) to *Labx* for all invertible in-gamut colors. To quantify this similarity we propose to use the quantity $\Delta E$, which is the Euclidean distance in the *Lab*-space between the input *Labx* and the output *Labz*. Thus, this quantity is used to judge the accuracy of the algorithm for all the colors selected on the input grid, which may or may not be on the nodes of the $P^{-1}$ lookup table. The larger the value of $\Delta E$, the more inaccurate the algorithm is.

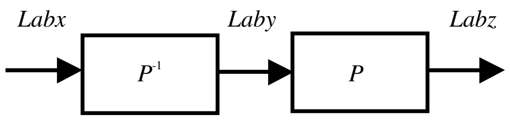

*Figure 1: Representation of the input-output data for the* $P^{-1}$ *model and the P model.*

The algorithm consists of three parts.

1. In the *first part* we obtain an initial estimate of the inverse for a given set of *Lab* values using the clustering interpolation method. These estimates are later refined to get the best accuracy (i.e., the lowest $\Delta E$ from *Labx* and *Labz*) using a gradient search method.
2. If the inversion of a particular color does not yield a good inverse (a low $\Delta E$), then the *second part* is invoked, in which the initial estimate of *Laby* is changed to *Labx* and the gradient search method is applied again for this new initial point.
3. If this part fails, then the *third part* is called, in which we change the initial *Lab* estimate of *Laby* by a method "similar" to the simulated annealing algor-ithm, and use the gradient search again. These three parts, outlined above and described in detail next, will give the best inverse for any color if that color is within the printer gamut (i.e., if the inverse exists).

### 2.1 Initial Estimation Using Clustering Interpolation

Let the forward printer model be specified by the lookup table mapping uniformly spaced points in the *Labi* space into non-uniformly spaced points in the *Labo* space. Points in the *Labi* space are chosen to form a perfect cube, and thus for the forward interpolation process from *Labi* to *Labo* various linear interpolation techniques on structured input can be used. To compute the initial estimate of the inverse of an arbitrary point $p_x$ in the output space, follow the steps described next (see Figure 2):

- Find the point $p_i$ that is the closest point to $p_x$ over all the points in the output of the lookup table.
- Find its pre-image point $Z_i$ in the *Labi* space ($Z_i$ is a grid point in the input space of the forward printer lookup table). This point is taken as a coarse estimate of the inverse to $p_x$.
- Improve this coarse estimate by moving along the L*, a* and b* axis around $Z_i$ and generating a cluster of *N* points (in our simulations we take *N=125*). Use a linear interpolation technique to find the *N* points in the *Labo* space corresponding to these *N* clustered points. (Recall that for the forward interpolation from *Labi* to *Labo* the input is structured and thus existing linear interpolation techniques give accurate results.)
- Select the nearest point to $p_x$ among all *N* points in the *Labo* space, and take its pre-image point in the *Labi* space as the initial estimate of the inverse of $p_x$.

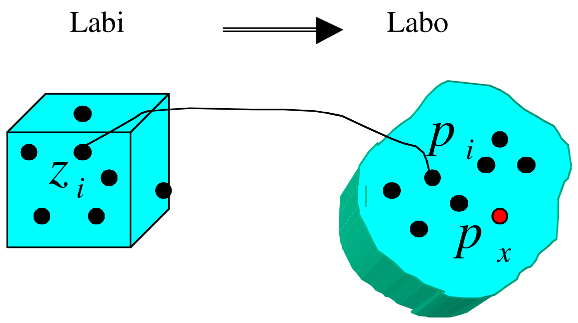

*Figure 2: Schematic representation of the input-output data for the forward printer model.*

If this estimate of the inverse of $p_x$ is not accurate enough ($\Delta E$ is large), it is then used as the starting point for a gradient search algorithm. This algorithm is described next.

## 2.2 Gradient Search Algorithm

Denote the estimate obtained from the clustering interpolation method by *Laby*(0). Now minimize the mean square error between the input to the inverse printer model *Labx* and the output from the forward printer model *Labz*. By computing the Jacobian of *Labz* with respect to *Laby,* we obtain the rule for updating *Laby* according to the following least mean square (LMS) recursive equation

$$Laby\,(k+1) = Laby\,(k) - \mu B\,(k)[Labz\,(k) - Labx],$$

where *Laby* is the 3 × 1 vector having L*, a*, and b* as its components at the input to the forward printer block *P*. Similarly, *Labz*(*k*) and *Labx* represent the output vector containing L*a*b* as its elements from the printer block *P* at the corresponding input vector in Figure 1. The 3x3 matrix *B*(*k*) is the Jacobian of *Labz*(*k*) with respect to *Laby*(*k*) which is evaluated from the printer model using linear interpolation and numerical differentiation. *k* is the iteration number localized to the gradient search algorithm. The quantity μ is selected to achieve a fast algorithm convergence and also to meet the accuracy requirement. Large values of μ will give faster convergence at the expense of increasing mean square error. In our simulations μ = 0.3 is found to be a good choice.

A large portion of the printer gamut can be "inverted" using this gradient search method together with the starting point provided by the clustering interpolation method. There are, however, invertible points in the printer gamut that are not captured by this approach. This is due to the fact that the mean square error function that we are minimizing has local minima, and the gradient search method with the above starting point will converge to one local minimum. To remedy this, we provide another method to compute the starting point for the gradient search.

## 2.3 Initial Estimation Using Alternative Methods

The method is as follows. First, choose the starting point *Laby(0)=Labx*. This starting point seems to be appropriate for colors near the center of the gamut. If this starting point still gives a poor inverse, use an approach "similar" to the simulated annealing method, i.e., we start with

$$Laby(k = 1) = (1 + \alpha)Laby(0),$$

where *Laby*(0) is the estimated point from the clustering interpolation method. A value of $\alpha$ = 0.25 seems appropriate for our simulations.

From our extensive experience in the laboratory and with simulations, the three methods for selecting starting points together with the gradient search method just described above worked well for all possible invertible points in the printer gamut. A flow chart, given in Figure 3, shows the data flow through various steps described above.

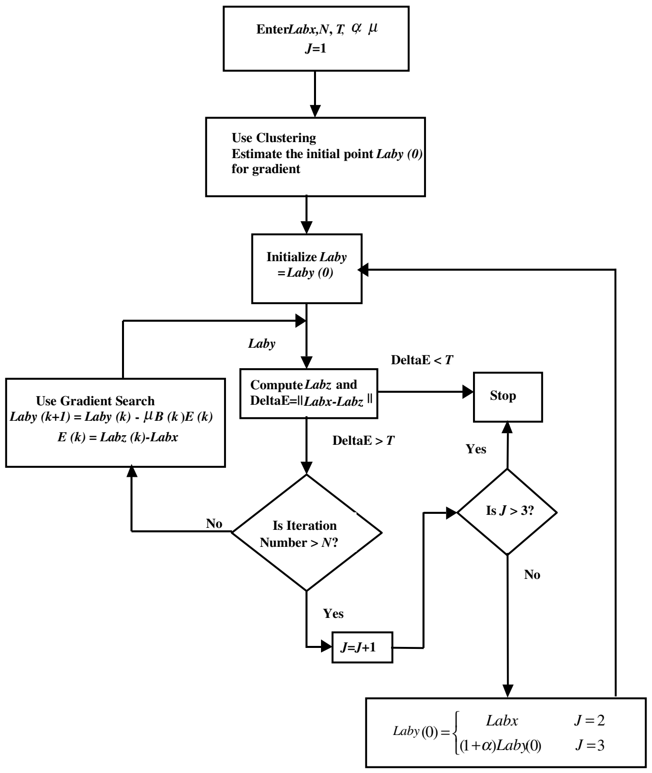

*Figure 3: Flow chart for the ICI algorithm*

## 3. Experimental Results

We tested the ICI algorithm to find the inverse lookup table for a Xerox printer. The input data to the printer was arranged as a 16 x 16 x 16 uniformly sampled cube (4096 points) covering the color points defined by L* from 0 to 100, a* and b* from –128 to 127. We built a forward printer model *P* using linear interpolation techniques. The printer inverse was constructed using the ICI algorithm. The input grid points *Labx* for the inverse were selected from a uniformly sampled input table, again a 16 x 16 x 16 uniformly sampled cube with the same bound as above. Some of the grid points inside this data set have an inverse, and some not. Through the ICI algorithm we were able to pick 266 nodes within the gamut that yield the inverse with a high mean accuracy and a small variance. The rest of the input points were either on the gamut boundary or outside the printer's reproducible space (i.e., they had no inverse). After this numerical exercise, the grid points containing inverse (i.e., the 266 colors) were then used in the printer to test the real accuracy between input to the inverse model and the real printer output. Accuracy results are shown in the table below, with and without the printer inverse, for the 266 colors. Mean deltaE is 22 for without inverse. That is the deltaE numbers obtained for 266 points for the measured forward printer model, *P*. Also, these results were compared with the inverse obtained from the master color control (MCC) algorithm. The results for the MCC algorithm are for 176 nodes only, because MCC algorithm did not work for all 266 colors obtained by the ICI algorithm. If we restrict to the common 176 nodes, the results are practically equal. In *summary*, the set of points in the printer gamut for which we can obtain an inverse using the ICI algorithm strictly contains the corresponding set for the MCC algorithm. Also, for these 176 "common points", we obtain a degree of accuracy of the same order for both the ICI and the MCC algorithms.

| | $\Delta E$ *mean* | $\Delta E$ *mean* + $2\sigma$ |
|---|---|---|
| Without Inverse (over 266 points) | 22.3 | 33.2 |
| With ICI Inverse (over 266 points) | 3.5 | 7.6 |
| With MCC Inverse (over only 176 pts) | 2.2 | 4.1 |

## 4. Conclusions

A new algorithm to compute a printer inverse lookup table with structured input nodes from unstructured data was described. This algorithm is based on a gradient optimization method, with starting points generated through a novel iterative technique. Experimental and numerical results show that the new algorithm gives a good inverse and outperforms other existing techniques.

## References

1. Henry R. Kang, "Color Technology for Electronic Imaging Devices", SPIE Press, 1997.
2. J.Z. Chang, J. Allebach, C. Bouman, "Sequential Linear Interpolation of Multidimensional Functions", IEEE Trans. Image Processing, vol. 6, pp. 1231-1245, September 1997.
3. Donald Shepard, "A two dimensional interpolation function for irregularly spaced data", ACM National Conference Proceedings, pp. 517 –524, 1968.
4. L.K. Mestha, "Dynamic Device Independent Image Correction Method and Apparatus", US Patent application number 09/083203 filed on May 22nd 1998.

## Glossary of Symbols and Notations

*Labi:* Uniformly sampled input to the lookup table of the forward printer.
*Labo:* Output color space corresponding to *Labi.*
*Labx:* Uniformly sampled input to the inverse printer model. This contains Lab values not in the lookup table.
*Laby:* Output of the inverse printer model. This contains output Lab values not in the lookup table.
*Laby*(0)*:* Initial estimate of *Laby*
*Laby(k):Laby* at k-th iteration of the gradient search algorithm.
*B(k): 3x3* Jacobian Matrix at *k*-th iteration.
*Labz:* Output of the printer model corresponding to *Laby*
*P:* Printer model defined through a uniformly sampled lookup table
***P***$^{-1}$: Inverse Printer model defined through a uniformly sampled look-up table
***N:*** **N**umber of points in cluster
*T :* Threshold for $\Delta E$
*J :* Method index
*k :* Iteration number
$\alpha$: Coefficient for initialization in the third method
$\mu$: Adaptation coefficient for gradient search

## Biography

L.K. Mestha has a Ph.D in control engineering, and currently he serves as a principal scientist in the Xerox Co.