Shaped Local Regression and Its Application to Color Transforms

Vishal Monga and Raja Bala, Xerox Research Center Webster, NY, USA

Abstract

Local linear regression is widely used in describing input-output relationships and has been applied with reasonable success to computational problems in color imaging such as approximating printer-models and device color characterization transforms. A popular flavor of local regression is one where locality is achieved by using a weight function which decays as a function of the distance from the regression data point. This paper proposes an improved method for local regression by introducing the notion of ``shaping" in the localizing weight function. We make two novel contributions: 1) a parameterization of the regression weight function via a shaping matrix, and 2) a method to optimize shape by explicitly introducing the shaping matrix parameters in the regression error measure. Experiments reveal dramatic improvements in approximating printer color transforms by using shaped local linear regression. A particularly pronounced benefit is gained in the case of sparse training sets, which are fairly common in color characterization applications due to the effort and/or cost associated with acquiring color measurements.

1. Introduction

Local regression has attracted considerable attention in both statistical learning literature [1] as well as a variety of applications where a complex non-linear multi-dimensional transform needs to be approximated with the availability of a training set of input-output data samples.

An important class of applied problems which benefits from local regression is the derivation of color device transforms. In all generality, the problem of color transform derivation is one of estimating the multi-dimensional color mapping $\mathbf{y} = f(\mathbf{x})$ where $\mathbf{x}$ and $\mathbf{y}$ represent color variables in distinct input and output multi-dimensional color spaces. Examples include CMYK → CIELAB color transforms, RGB → CMYK transforms etc. Local regression is well suited to these problems for two reasons: 1.) physically based models that describe these relationships are often inadequate, and 2.) color transforms exhibit varying curvature as a function of location in color space, engendering the need for locally based approximations.

In particular, local *linear* regression is widely employed because it provides the desirable trade-off between achieved accuracy and ease of formulation and computation. *Local* linear regression exploits the fact that, over a small enough subset of the input domain, any reasonable function *can* be well approximated by a linear function.

As with any local learning problem, the user must define what is to be considered local to a test point. Two standard methods are employed for defining locality: (i) specifying the number of neighbors k also known as a neighborhood definition, or (ii) specifying the volume of a symmetric distance-decaying weight function which attaches more weight to nearby points. Commonly, the neighborhood size or the volume of the weight function is chosen by cross validation over training samples [2].

Much previous work in local regression has focused on adaptive neighborhood definitions. This includes k-nearest neighbors (NN), k-surrounding neighbors, space partitioning techniques and input density based neighborhood selection. Recently, the idea of enclosing neighborhoods was put forward by Gupta et al in [3], which also provides a comprehensive review of neighborhood selection techniques.

Like most statistical smoothing approaches, local regression suffers from the so-called "curse of dimensionality", the well known fact that the proportion of training data that lie in a fixed radius neighborhood of a point decreases to zero at an exponential rate with increasing dimension of the input space. The neighborhood based approaches for local regression therefore fare well for low-dimensional data (<=2) and a reasonably dense sampling of the input space to generate the training set. Color data can easily be of high dimensionality, e.g. CMYK or more colorants, and a dense sampling of the colorant space would incur significant printing and measuring. Many neighborhood definitions, e.g. the convex hull based neighborhoods in [3] will actually lead to degenerate neighborhoods for high-dimensional data and hence yield high model bias or regression error.

In this paper, we propose the use of weight function *shaping* in local linear regression to address this problem. The fundamental premise of our scheme is to recognize that it is not just the volume of the decaying weight function but how that volume is spread out directionally, i.e. the shape, which plays an important part in estimating the true regression transform f($\mathbf{x}$). Analogous to neighborhood based approaches; our proposal decides which training data points to pick and how much weight to attach to them. For sparse and high-dimensional data (>=3) this is a crucial concern because making the best use of a given *volume* is often of paramount importance to the quality of estimates.

To achieve this in an automated fashion, we parameterize the weight function using a "shaping matrix". We then select optimal shapes that provide the best estimates of the true transform by minimizing error over the training set. In other words, the regression cost function is re-formulated to include a joint minimization over the regression parameters as well as the shaping matrix.

We apply the proposed shaped local linear regression (SLLR) to the problem of modeling CMYK → Lab printer color transforms with good success. The benefits over local linear regression (LLR), which employs a fixed symmetric shape, are particularly pronounced for sparse training data that is representative of real world imaging applications.

The rest of the paper is organized as follows. Section 2 reviews traditional linear regression, and local linear regression where locality is achieved by a weight function. Section 3 introduces and motivates shaping in the weight function via a synthetic example, and formulates SLLR as an optimization problem. Section 4 presents experimental results in approximating a real-world printer color transform. Section 5 concludes the paper with directions for future work.

2. Multi-dimensional local linear regression

Regression is a common technique for estimating a functional relationship between input and output data. Linear regression is a specific case where the functional relationship is approximated by a linear transform. When the input and output data belong to multidimensional vector spaces, the linear transform is a matrix. Specifically, consider the problem where **y** in R^m is to be estimated as a function of an input variable **x** in R^n. Let $\Gamma = \{ (\mathbf{x}_i, \mathbf{y}_i), i = 1, 2, \ldots, T\}$ denote the set of training data over which this response in known. The linear approximation is given by:

$$\mathbf{y} = \mathbf{A}.\mathbf{x}, \quad \mathbf{x} \in R^n, \mathbf{y} \in R^m, \mathbf{A} \in R^{m\times n} \qquad (1)$$

The best regression parameter **A** is determined by minimizing the regression cost function that describes an aggregate error between $\mathbf{y}_i$ and $\mathbf{A}.\mathbf{x}_i$ over the training set.

Local linear regression [1] is a generalization wherein the matrix **A** varies as a function of location **x** in input space. Thus we have

$$\mathbf{y} = \mathbf{A}_X.\mathbf{x}, \quad \mathbf{x} \in R^n, \mathbf{y} \in R^m, \mathbf{A} \in R^{m\times n} \qquad (2)$$

One way to achieve this locality is to obtain the regression matrix $\mathbf{A}_x$ for each **x** by simply carrying out the regression in a neighborhood of **x** which is a subset of the training set. Such neighborhood definitions have indeed been the focus of much research in local regression and are reviewed in [2, 3].

This paper focuses on the alternate formulation, where for each input data point **x**, the "best" regression parameter $\mathbf{A}_x$ is determined by minimizing the regression cost function:

$$C(\mathbf{A}_x) = \frac{1}{T}\sum_{i=1}^{T} \left\| \mathbf{y}_i - \mathbf{A}_x.\mathbf{x}_i \right\|^2 .w(\mathbf{x}_i, \mathbf{x}) \qquad (3)$$

Here $w(\mathbf{x}_i, \mathbf{x})$ is a non-negative weighting function that assigns more weight to training data points *closer* to **x** than to those distant from **x.** A popular instantiation is:

$$w(\mathbf{x}, \mathbf{x}_i) = e^{-\alpha\left(\|x - x_i\|^2\right)} \qquad (4)$$

The above weight function is plotted in Fig. 1 for a 2-D input variable **x**. Clearly, there is decay as a function of distance from the center point, $\mathbf{x}_0$ which means that in the cost function of Eqn (3), more weight is attached to regression sets for which $d(\mathbf{x}_i, \mathbf{x}_0)$ is small. Qualitatively, the subset of $\{\mathbf{x}_i\}$'s for which $w(\mathbf{x}, \mathbf{x}_0)$ is greater than a threshold (so as to have a significant impact on the cost function) constitutes a local neighborhood of $\mathbf{x}_0$.

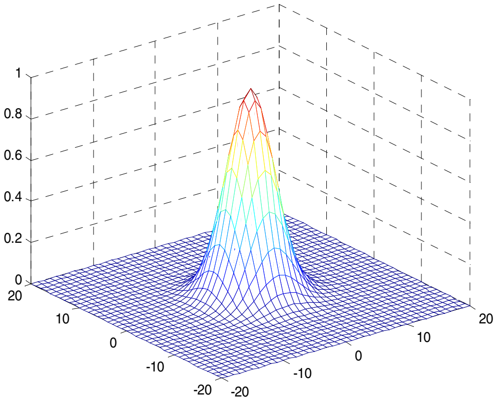


Fig. 1: Exponential weighting function given by Eqn (4)

The more general requirement is for $w(\mathbf{x_i}, \mathbf{x})$ to decay as a function of the distance $d(\mathbf{x}, \mathbf{x_i})$**.** Hence, $w(\mathbf{x_i}, \mathbf{x})$ may be expressed as:

$$w(\mathbf{x}, \mathbf{x}_i) = \varphi(d(\mathbf{x_i}, \mathbf{x})) \quad , \qquad (5)$$

where φ is a univariate non-negative function that is decreasing in its argument. For the instantiation of the weight function in Eqn (4), $\varphi(z) = \exp(-\alpha.z^2)$ and $d(\mathbf{x}, \mathbf{x_i}) = \|\, \mathbf{x} - \mathbf{x_i} \|$**.**

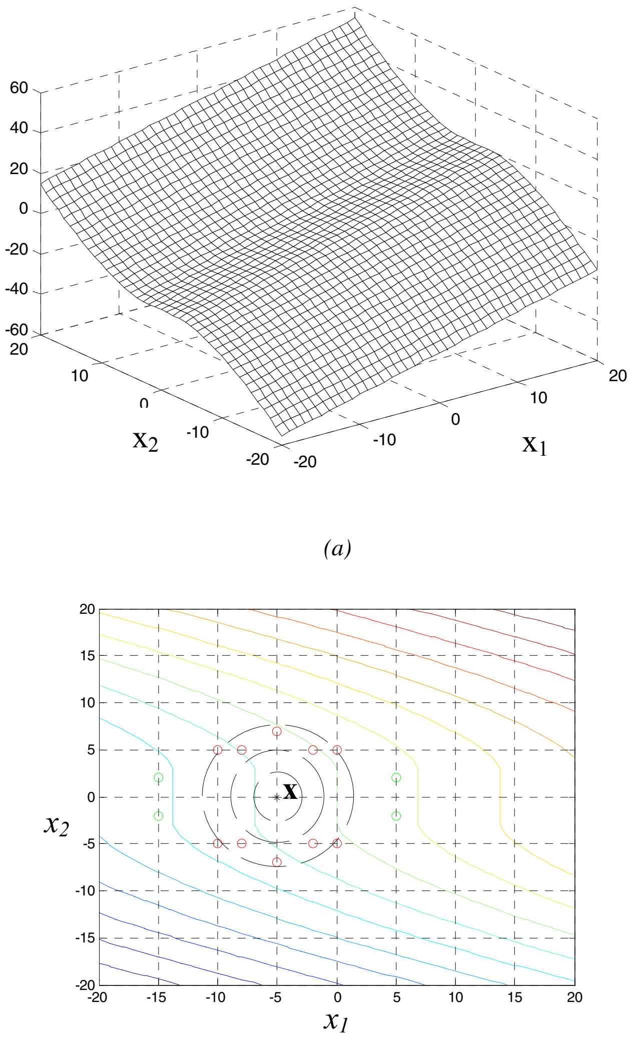
3. Shaped Local Linear Regression (SLLR)

We begin by observing that the particular choice of $w(\mathbf{x}, \mathbf{x}_i)$ has a significant impact on the solution to the regression error minimization problem. Further, this depends in general both on as well as the choice of the distance function $d(\mathbf{x}, \mathbf{x}_i)$**.**

To appreciate this, consider the following synthetic function of two variables $\boldsymbol{x} = (x_1, x_2)$:

$$f(x_1,x_2) = x_1 + g(x_2)$$
$$where \quad g(x_2) = \begin{cases} 0 & if \quad |x_2| \le \alpha \\ \beta|x_2-\alpha|^p & if \quad x_2 > \alpha \\ -\beta|x_2+\alpha|^p & if \quad x_2 < -\alpha \end{cases} \tag{6}$$

Note that *f( )* is a linear function in the regime $|x_2| \le \alpha$, and assumes a nonlinear power-law function outside this regime. Fig. 2(a) shows a plot of the function for the case where $\alpha = 4$, $\beta = 0.5$, and $p = 1.5$.



(a)

(b)

Fig. 2: (a) The simulated 2-D function of Eqn (6), (b) Contour plot of simulated function with training data overlaid. Green circles lie within the linear region of f() while red circles are located in the nonlinear regime. The circles are the contours of the distance function induced by LLR.

Fig. 2 (b) shows contours of the same function with training data overlaid. It is desired to compute the regression output for a test data point **x**. This figure illustrates the impact of the weight function *w*(**x, x**$_i$) on the regression error. Note that the circles centered around regression point **x** are contours of the standard Euclidean distance function d(**x, x')** = || **x – x'||.** The different circular contours correspond to varying choice of in Eqn. (4) – which is inversely related to the volume (spread) of the weighting function.

From a visual inspection of the plot in Fig. 2(a) it is clear that the function is linear in the vicinity of the regression point and can hence be perfectly approximated with local linear regression. But for that to happen, the regression must attach significant weights only to training data in the "linear region" (shown in green) and essentially ignore any training points in the non-linear region (shown in red). However, with the circular spreads induced by the standard Euclidean metric in Fig. 2(b), the red training points are closer to **x** and will in fact get more weight than the green points. This holds regardless of the radius of the circular contours or equivalently weight function volume. Thus even though the function is locally linear, LLR will introduce error via its inferior choice of how to weigh the training samples.

We now present a generalization of the Euclidean distance metric which effectively addresses the aforementioned problem. For finite-dimensional input spaces, the square of the distance function d(**x**, **x**$_i$) = || **x – x**$_i$|| can alternatively be written as

$$\left\|\mathbf{x}-\mathbf{x}_i\right\|^2 = (\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}-\mathbf{x}_i) \tag{7}$$

As shown in Fig. 1 (b), contours of such a distance function result in hyper-spheres in R^n (special case circle in R^2). A generalization of this distance is:

$$\left\|\mathbf{x}-\mathbf{x}_i\right\|_\Lambda = (\mathbf{x}-\mathbf{x}_i)^T\Lambda(\mathbf{x}-\mathbf{x}_i) \tag{8}$$

where **Λ** is defined as a positive definite matrix to ensure non-negativity of the distance for all **x, x**$_\mathbf{i}$**.** It is easy to see now that the contours of this new distance are elliptical. A diagonal **Λ** with positive unequal diagonal entries results in a hyper-ellipse with different ellipse radii in different dimensions, while non-diagonal choices of Λ allow the control of orientation.

Optimizing the choice of Λ: It is not difficult to see that the local linear estimates resulting from different choices of **Λ** can vary considerably. A reasonable strategy would be to choose **Λ** proportional to the sample covariance matrix.

In locality induced by weight functions, the focus has been on carefully choosing volume or spread of the weight function. We argue that optimizing shape of the weight function is at least as important as optimizing the volume. More specifically, for a *fixed* volume of the weighting function, the bias of the estimate can be reduced drastically by shrinking the weight function in directions of large non-linear variation of the true transform f(**x**), and stretching in directions of small non-linear variation.

To formally distinguish shape from size, we re-write Λ as follows:

$$\Lambda = \alpha \mathbf{S}^T \mathbf{S}; \quad \Lambda, \mathbf{S} \in R^{n \times n} \tag{9}$$

where **S** denotes the "shape matrix" and *α* quantifies the (inverse of the) volume of the localizing weight function. Clearly, for such an approach to be meaningful we need to restrict the volume of the weight function; otherwise the bias of the estimate could be minimized trivially by choosing zero volume. One way to accomplish this is to set a measure of the spread of Λ to a constant value. More elaborate methods for controlling volume such as those based on number of neighboring observations are known and the reader is referred to [2] for a review.

For this work, because we focus on demonstrating benefits of shaping we choose to set det ( ) as a constant. This in turn may be done by setting   to be a constant while imposing det (S) = 1.

Given this separation of volume and shape, the shaping matrix may be solved for optimally by generalizing Eqn (3) to define a new regression error measure:

$$\begin{aligned} &C(\mathbf{A}_x, \Lambda) = \frac{1}{T}\sum_{i=1}^{T} \left\| \mathbf{y}_i - \mathbf{A}_x . \mathbf{x}_i \right\|^2 . w_\Lambda(\mathbf{x}, \mathbf{x}_i) \\ &\text{where } w_\Lambda(\mathbf{x}, \mathbf{x}_i) = e^{-(\mathbf{x}-\mathbf{x}_i)^T \Lambda (\mathbf{x}-\mathbf{x}_i)}, \Lambda = \alpha \mathbf{S}^T \mathbf{S} \\ &\text{subject to } \det(\mathrm{S}) = 1, \alpha = \mathrm{c} \end{aligned} \tag{10}$$

For each regression test point, the optimal regression as well as shape parameters are obtained by minimizing the above error measure. We refer to this framework for regression as shaped-local linear regression (SLLR).

Comparing the optimization problem above against Eqn. (3), we note that the shaping matrix has been explicitly introduced in the optimization. Hence, the problem may be understood as one of determining the shape in conjunction with the regression matrix $\mathbf{A}_x$ that provides the best estimate of the true transform f(**x**). In other words, ideally the shaping would determine the ideal directional "shrinkage" or "spread" of the weight function depending on transform curvature, and the optimal matrix is then derived for this weight function.

Reverting to the synthetic function in Fig. 2, we recall that it is indeed the lack of spread in the linear region that leads to high regression error in Fig. 2 (b). Figure 3 shows the same plot of *f*( ) along with a contour plot of the shaped distance function obtained by solving the SLLR problem of Eqn (10) (black ellipse). We clearly see that the shaping function elongates to favor those (green) training samples that lie within the linear region of *f*( ). Linear regression using these samples will indeed result in a perfect approximation. As a point of comparison, the blue circle is the contour of the distance function induced by LLR for an equivalent weight function volume.

Note that although the concept of shaping been described for linear regression, in principle the same technique readily extends for nonlinear regression. The elements of the matrix **A** would be simply replaced by the parameters of the nonlinear approximation function. Likewise, a generalization of shaping is possible by alternate choices of d(**x**, $\mathbf{x}_i$) and (z).

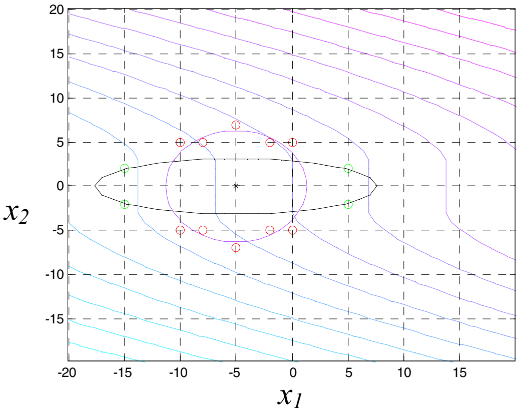


Fig. 3: When shaping is introduced into the regression formulation, the distance function and associated definition of neighborhood are automatically adapted according to the local characteristics of the function.

4. Experimental Results

We compared both LLR and SLLR in approximating a printer characterization transform mapping CMYK to CIELAB. A set of 336 CMYK samples spanning the entire device gamut, and their corresponding CIELAB measurements, were used as the training set for the regression. A standard IT8 target of 928 CMYK patches was used as the test set.

The CIELAB approximation error incurred by a given regression technique (LLR or SLLR) was computed in two different ways. First the standard ΔE_{ab} color difference metric was computed. Second, a signal-to-noise (SNR) metric was computed between the squared error obtained from SLLR vs SLR:

$$\begin{aligned} E_{LLR} &= \sum_{i=1}^{M} \left\| F(x_i) - \hat{F}_{LLR}(x_i) \right\|^2 \\ E_{SLLR} &= \sum_{i=1}^{M} \left\| F(x_i) - \hat{F}_{SLLR}(x_i) \right\|^2 \\ J &= 20 \log_{10} \frac{E_{LLR}}{E_{SLLR}} \end{aligned} \tag{11}$$

where $\hat{F}_{LLR}(x_i)$ and $\hat{F}_{SLLR}(x_i)$ are CIELAB approximations computed at the i-th CMYK sample (1 • *i* • *M*) via LLR and SLLR respectively. For example, a gain of 6 dB means that the error from the SLLR approximation is about one-half of that from LLR.

We quantify these error measures for two different cases. The results for these two experiments are reported next in Sections 4.1

and 4.2. Additionally, 4.3 reports some statistics of the shape matrix in SLLR over CMYK color space in the form of deviations from an identity matrix which would be the case if LLR was used.

4.1 A purely computational approximation to a 4-color printer model

In the first case, the non-linear CMYK → LAB transform is approximated by a purely computational technique – i.e. simply by using LLR and SLLR directly in the CMYK color space. The results for this experiment are shown in Table 1 for the IT8 test set and labeled as "direct approximation".

Clearly SLLR significantly outperforms LLR. In particular, the J (dB) values indicate that the squared error in Eqn (11) is cut by more than half by optimizing shape in local regression.

The reader may note that all the ΔE values are relatively high for this case. This is because we are attempting to fit a highly complex nonlinear 4-dimensional printer response with a relatively sparse training set of 336 samples spanning all of CMYK space.

4.2 Refinement of a cellular spectral Neugebauer printer model

A large body of literature exists in the derivation of physically based printer models [4]. For halftone printers, a popular choice is the classical Neugebauer printer model [5] where a few canonical color patches of solid overprints (primaries) and single-colorant ramps are printed and reflectance spectra are measured for each patch. Approximations of the reflectance for an arbitrary CMYK patch are derived by a mixing model in conjunction with dot area functions. The second technique we propose uses local regression to refine estimates obtained via a state-of-the-art cellular spectral Neugebauer printer model [6]. That is, a CMYK → Lab printer model (really a spectral printer model and subsequent conversion to CIELAB) is first developed using a cellular Neugebauer model. Then either of LLR or SLLR are used as a post-processing refinement transform to provide an improved model prediction. Specifically, the regression is derived from LABin → LABout training data pairs, where LABin represents estimates from the cellular Neugebauer model, and LABout represents the "actual" LAB values as determined via measurement.

Results for this experiment are also in Table 2. Again, the benefits of SLLR are readily apparent. As expected, the gain is less pronounced over what is observed in a direct approximation because the cellular Neugebauer printer model captures some of the non-linearity in the transform. Overall, the average as well as 95th percentile ΔE errors are much smaller than in the direct approximation case and more representative of the state of the art in printer model approximations. Indeed the performance of SLLR for refinement is remarkable given the relatively small training set.

4.3 Shape matrix variation in CMYK color space

In this Section we provide statistics of the shaping matix to show that real-world printer transforms can significantly benefit from shaping when approximated via local regression.

Experiment		Avg ΔE	95% ΔE	J
Direct approximation	LLR	4.53	6.94	8 dB
	SLLR	2.45	6.27	
Refinement on top of cellular Neugebauer	LLR	3.07	5.44	5.3 dB
	SLLR	2.12	5.00	

Table 1: Comparison of LLR and SLLR for approximating a forward printer characterization transform

We propose a simple measure labeled d_I which measures deviation of from an identity matrix of the same determinant:

$$d_I = \left\| \mathbf{\Lambda} - \beta^{1/n} . \mathbf{I} \right\|_F$$

where **I** is an n x n identity matrix of the same size as such that det( ) = det($\beta^{1/n}$ **I**) = **.**

Hence a value of $d_I \approx 0$ implies that the optimal shape was hyper-sphere, i.e. coinciding with LLR. But $d_I > 0$ indicates benefits were seen for alternate shapes. Further, note that this measure simultaneously captures notions of eccentricity as well as change in orientation which together comprises the shape. Fig 4 shows the histogram of d_I computed for the IT8 test set. Clearly considerable deviation from a hyperspherical shape is observed.

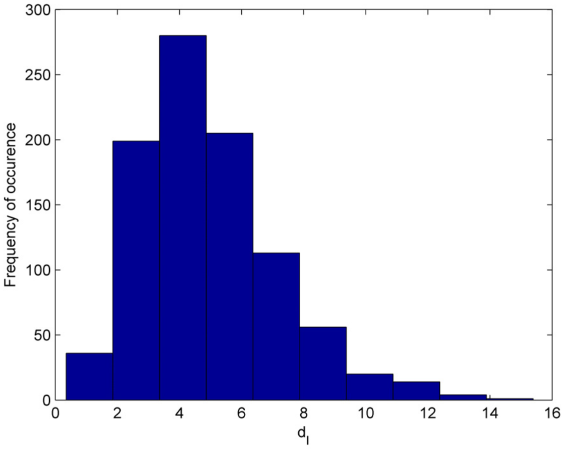


Fig. 4: Distribution of deviation measure d_I over CMYK color space. $d_I >> 0$ indicates optimized shape differs considerably from the isomorphic hyper-spherical neighborhoods induced by in standard local regression.

5. Conclusion

The concept of shaping has been introduced into local regression. The phenomenon of shaping is parameterized via a simple shaping matrix, which is then explicitly introduced into the regression cost function. In this manner, both the shape of the weighting function and the coefficients of the regression transform are jointly optimized. The benefits of shaped local linear regression over standard local linear regression are clearly demonstrated in a printer characterization application. The proposed technique is deemed to be particularly advantageous when the training data set is sparse and/or the dimensionality of the input space of the transformation is high. Future work will explore efficient algorithms for the optimization problem introduced in this paper and incorporating physical device characteristics into determining what is optimally local.

6. References

[1] C. G. Atkeson, A. W. Moore and S. Schaal, "Locally weighted learning", *Artificial Intelligence Review*, 1997

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer-Verlag, 2001.

[3] M. Gupta et al., "Adaptive Local Linear Regression With Application to Printer Color Management*", IEEE Transactions on Image Processing*, VOL. 17, NO. 6, JUNE 2008

[4] R. Bala, "Device Characterization", *Digital Color Imaging Handbook*, Chapter 5. CRC Press, 2003.

[5] W. Rhodes, "Fifty years of the neugebauer equations," *Proc. SPIE 1184*, pp. 7{18, 1989.

[6] R. Balasubramanian, "Optimization of the spectral neugebauer model for printer characterization," *Journal of Electronic Imaging*, vol. 8, pp. 156-166, Apr. 1999.

Author Biography

Vishal Monga received the BTech degree in Electrical Engineering from the Indian Institute of Technology (IIT), Guwahati in May 2001 and a PhDEE from the University of Texas, Austin in Aug 2005. Following a 4 year stint with Xerox Research Labs, Dr. Monga moved to the Pennsylvania State University in University Park, PA where he is currently an Assistant Professor. Dr. Monga's research interests lie broadly in statistical signal and image processing. In color image processing, he is particularly interested in computational problems that manifest in device modeling and in printer and digital camera image processing pipelines.