Illumination estimation via thin-plate spline interpolation

Lilong Shi,[1,*] Weihua Xiong,[2] and Brian Funt[1]

[1]*Simon Fraser University, Burnaby V5A-1S6, Canada*
[2]*Omnivision Corporation, Santa Clara, California 95054, USA*
**Corresponding author: lshia@cs.sfu.ca*

Received October 6, 2010; revised January 29, 2011; accepted March 16, 2011;
posted March 28, 2011 (Doc. ID 136186); published April 29, 2011

Thin-plate spline interpolation is used to interpolate the chromaticity of the color of the incident scene illumination across a training set of images. Given the image of a scene under unknown illumination, the chromaticity of the scene illumination can be found from the interpolated function. The resulting illumination-estimation method can be used to provide color constancy under changing illumination conditions and automatic white balancing for digital cameras. A thin-plate spline interpolates over a nonuniformly sampled input space, which in this case is a training set of image thumbnails and associated illumination chromaticities. To reduce the size of the training set, incremental k medians are applied. Tests on real images demonstrate that the thin-plate spline method can estimate the color of the incident illumination quite accurately, and the proposed training set pruning significantly decreases the computation.

OCIS codes: 330.0330, 330.1690, 330.1710, 330.1720, 100.2000.

1. INTRODUCTION

A new approach to illumination estimation for color constancy and automatic white balancing is developed based on the technique of *thin-plate spline* (TPS) interpolation. Assuming there is a single, dominant scene illuminant, we describe the overall scene illumination in terms of its chromaticity components r and g [$r = R/(R+G+B)$, $g = G/(R+G+B)$]. These chromaticity components can be viewed as functions of the chromaticity image (each input RGB pixel value converted to rg chromaticity space) $\boldsymbol{I}$ of the scene; namely, $r = f_r(\boldsymbol{I})$ and $g = f_g(\boldsymbol{I})$. The problem of estimating the rg chromaticity of the illuminant then becomes that of estimating the two functions $r = f_r(\boldsymbol{I})$ and $g = f_g(\boldsymbol{I})$. The proposed method estimates these functions by interpolating over the ground-truth r and g chromaticity values measured from a training set of images.

Interpolation is a common problem, and there are many well-established interpolation methods [1]. The majority of these methods, such as bilinear or bicubic interpolation, are based on interpolation over training data sampled on a uniform grid. However, since there is no clear way to sample uniformly the space of images, interpolation over a nonuniformly sampled space is required. TPS is an effective interpolation method under these conditions and has been widely used in the context of deforming one image into registration with another [2]. In the case of illumination estimation, TPS maps an input rg-chromaticity image to the r-chromaticity and g-chromaticity values of the scene illumination.

The intuition underlying the use of TPS for illumination estimation is similar to that of other training methods such as the neural network approach [3] or the use of Support Vector Regression [4]. Each "learns" a function from the training data. The advantage of one of these techniques over the other will eventually be measured in terms of their relative performance as well as memory and processing requirements. However, one specific advantage of TPS over neural nets and Support Vector Regression is that it is easy to envision the nature of what TPS is learning since it straightforwardly creates a simple function that smoothly interpolates the training data points. A second advantage of TPS is that it is much more straightforward to apply than either neural nets or Support Vector Regression. It could be that some neural net or Support Vector Regression implementation would work as well as TPS, but there is the problem of determining what that implementation might be. Neural nets have many more variables—number of nodes, hidden layers, interconnections, and so on—than TPS and so are more difficult to tune. It could be that *some* neural net would work as well, but how to design it? Similarly, there are many choices in the way Support Vector Regression is applied in terms of the kernel functions, preprocessing, and so on, so once again it is possible that *some* Support Vector Regression implementation could work as well as TPS, but again, which implementation?

In terms of the algorithm's efficiency, the most important factor is to reduce the training set's size since both the computation and memory requirements grow polynomially with the number of training images. Therefore, when the training set becomes large, both the training and the testing become infeasible. We address this problem by finding representative training images and pruning the remaining ones.

2. TPS METHOD

As is typical of interpolation methods in general, TPS constructs a function that matches a given set of data values $\{y_i\}$, corresponding to a given set of data vectors $\{x_i\}$, in the sense that $y_i = f(\boldsymbol{x}_i)$. TPS interpolation was originally designed for two-dimensional image registration [2,5–7]. In the color context, it has been extended to three dimensions and successfully applied to the problem of camera and color display calibration [8]. Compared with other methods, TPS

has been found to be quite stable and accurate in terms of finding a unique solution without having to tune a lot of parameters. Here we extend TPS to D dimensions and apply it to the problem of estimating the chromaticity of a scene's overall incident illumination from an image of that scene.

TPS for illumination estimation requires a "training" set, $\mathbb{T}$, of N chromaticity images along with their corresponding illumination chromaticities. For the ith RGB image containing n pixels, its chromaticities are formed into a $D=2n$ vector, $\boldsymbol{I}_i$. $\mathbb{T}$ contains the pairs $(\boldsymbol{I}_i,(r_i,g_i))$. Solving TPS means determining the parameters (ws and as below) that control the two mapping functions f_r and f_g, such that

$$\begin{cases} r_i = f_r(\boldsymbol{I}_i) \\ g_i = f_g(\boldsymbol{I}_i) \end{cases}. \tag{1}$$

Given a chromaticity image $\boldsymbol{I}_s=(I_{s,1},I_{s,2},\ldots,I_{s,d},\ldots,I_{s,D})$, where $I_{s,d}$ stands for the dth element of image s, the mapping function f_r is defined as (function f_g is defined similarly)

$$f_r(\boldsymbol{I}_s)=\sum_{i=1}^{N} w_i U(\|\boldsymbol{I}_s-\boldsymbol{I}_i\|)+a_0+\sum_{j=1}^{D} a_j I_{s,j}, \tag{2}$$

where $U(x)=x^2\log x$ and $\|\cdot\|$ is the operator for Euclidean distance. There are N weights w_i that control the nonlinear terms and $D+1$ coefficients a_j that control the additional linear terms.

Each element of the training set (an image plus its illumination chromaticity) provides two equations, as shown in Eq. (1). In addition, a smoothness constraint is usually imposed that minimizes the bending energy. In the original TPS formulation [1], the bending energy function $J(f)$ of the function $f(x_1,x_2)$ is defined over $\mathcal{R}^2$ as

$$J(f)=\iint_{\Re^2}(f_{x_1x_1}^2+2f_{x_1x_2}^2+f_{x_2x_2}^2)\mathrm{d}x_1x_2. \tag{3}$$

The bending energy models the energy required to bend a thin sheet of metal to pass through the control points and is measured by the second derivatives of the surface. The bending energy of a plane is zero. Here we use the generalized function J in D-dimensional domain $\mathcal{R}^D$ with order D of derivatives. Hence, Eq. (3) can be generalized over $\boldsymbol{x}=(x_1,x_2,\ldots,x_D)$ yielding

$$J(f_r)=\sum_{\theta_1+\theta_2+\ldots+\theta_D=D}\left[\frac{D!}{\theta_1!\theta_2!\ldots\theta_D!}\int_{\Re^D}\left(\frac{\partial^D f_r}{\partial x_1^{\theta_1}\partial x_2^{\theta_2}\ldots\partial x_D^{\theta_D}}\right)^2 \times \mathrm{d}x_1\mathrm{d}x_2\ldots\mathrm{d}x_D\right], \tag{4}$$

where $\{\theta_i\in N|1\le\theta_i\le D\}$ and $J(f_r)$ is the total bending energy described in terms of the curvature of f_r by higher order of derivatives (the bending energy for f_g is defined similarly).

Following [9–11], the energy $J(f_r)$ in Eq. (4) over all images $\{\boldsymbol{I}_i|i=1,\ldots,N\}$ in the training set $\mathbb{T}$ will be minimized when the following constraint holds:

$$\sum_{i=1}^{N} w_i=\sum_{i=1}^{N} I_{i,1}w_i=\sum_{i=1}^{N} I_{2,1}w_i=\ldots=\sum_{i=1}^{N} I_{i,D}w_i=0, \tag{5}$$

In order to solve TPS in estimating the parameter set, both Eqs. (1) and (5) need to be satisfied at the same time. Given a training set $\mathbb{T}$ containing N images of size D, there are $(N+D+1)$ linear equations and unknowns that can be uniquely solved by matrix operations. This is true for each of f_r and f_g. In particular, we can define $\boldsymbol{U}$, $\boldsymbol{Q}$, $\boldsymbol{w}$, $\boldsymbol{a}$, and $\boldsymbol{c}$, such that

$$\boldsymbol{U}\boldsymbol{w}+\boldsymbol{Q}\boldsymbol{a}=\boldsymbol{c} \qquad \boldsymbol{Q}^T\boldsymbol{w}=\boldsymbol{0}, \tag{6}$$

where

$$\boldsymbol{U}=\begin{bmatrix} 0 & U_{1,2} & \cdots & U_{1,N} \\ U_{2,1} & 0 & & U_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ U_{N,1} & U_{N,2} & \cdots & 0 \end{bmatrix}$$

$$\boldsymbol{Q}=\begin{bmatrix} 1 & I_{1,1} & I_{1,2} & \cdots & I_{1,D} \\ 1 & I_{2,1} & I_{2,2} & \cdots & I_{2,D} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & I_{N,1} & I_{N,2} & \cdots & I_{N,D} \end{bmatrix},$$

$$\boldsymbol{w}=(w_1,w_2,\ldots,w_N)^T \quad \text{and} \quad \boldsymbol{a}=(a_0,a_1,\ldots,a_D)^T,$$
$$\boldsymbol{c}=(r_1,r_2,\ldots,r_N)^T \quad \text{or} \quad \boldsymbol{c}=(g_1,g_2,\ldots,g_N)^T,$$
$$U_{i,j}=U(\|\boldsymbol{I}_i-\boldsymbol{I}_j\|).$$

The vector $\boldsymbol{c}$ represents the chromaticities of the corresponding N illumination chromaticities $\{(r_i,g_i)\}$ in the training set $\mathbb{T}$. Symbol $\mathbf{0}$ is a vector of $D+1$ zeros. The linear system described in Eq. (6) can be rewritten in an even more compact form as

$$\begin{bmatrix} \boldsymbol{U} & \boldsymbol{Q} \\ \mathbf{Q}^T & \bar{\boldsymbol{0}} \end{bmatrix}\times\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{a} \end{bmatrix}=\begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{0} \end{bmatrix} \quad \text{or } \boldsymbol{L}\times\boldsymbol{W}=\boldsymbol{C}. \tag{7}$$

Here, the matrix $\bar{\boldsymbol{0}}$ contains $(N+1)\times(N+1)$ zeros.

Equation (7) forms a linear system of $N+D+1$ equations, which are then easily solved by $\boldsymbol{W}=\boldsymbol{L}^{-1}\times\boldsymbol{C}$. Once the parameter set $\boldsymbol{W}=(\boldsymbol{w},\boldsymbol{a})$ is calculated, the illumination (r_s,g_s) of an input image $\boldsymbol{I}_s$ can be estimated by applying Eq. (1) for r and g separately.

The complexity of TPS-based interpolation is primarily related to the matrix operations. For example, the training phase of TPS requires a matrix inverse operation, which has a complexity of $O((N+D)^3)$ in general. The testing phase, on the other hand, only requires $O(\log ND)$ operations.

3. REDUCING THE TRAINING DATA SET

TPS interpolates data between points defined by the training set. This implies we have to have the entire training set available during testing. Since the training stage of TPS requires a matrix inversion operation of an $(N+D+1)$-by-$(N+D+1)$ matrix, the matrix inversion step becomes infeasible when the training-set size or image dimension is large, in the sense of both CPU computation and memory usage.

To reduce the training-set size, we need to select good representatives from the initial training set. For example, if there are two similar images in the training set, then likely

one can be removed without much loss in overall accuracy. The problem is how to decide which training images can be discarded and which should be retained. As one possible solution, we propose using the k-median algorithm to cluster the training data $\mathbb{T}$ and then retain only the subset $\mathbb{T}'$ consisting of the k cluster centroids of $\mathbb{T}$. The k-median algorithm is similar to the standard k-mean clustering algorithm except that, as its name implies, it is based on the median rather than the mean. The advantage of k-medians for TPS is that the median image defining each cluster's center is necessarily one of the input images. In contrast, with k-means a cluster's center is defined by the mean of the cluster, which is not necessarily a member of the original data set.

To find a large number of clusters from a large data set, the k-median computation can be quite intensive, especially when the dimensionality of the data is high as well, so we have implemented an incremental k-median approach [12]. This approach requires much less memory and only sublinear computational time. The incremental k-median algorithm is an approximate clustering method that incrementally updates the existing clusters based on a divide-and-conquer strategy. The whole data set is divided into trunks that are clustered individually, and their centers are merged to generate the final result [12]. In general, the complexity of a batch k-median algorithm is non-deterministic polynomial-time hard (NP-hard). The complexity of incremental k medians reduces this to $O(mn + nk\log(nk))$, where n is the size of data set, k is the number of clusters, and m is a constant factor related to the size of available memory [12]. Figure 1 shows a plot of the 220 clusters (out of 11,346 images [13]) found by k medians denoted by red circles superimposed onto a two-dimensional projection of a principal component analysis (PCA) basis of the training images.

Another way to reduce the training set is to reduce the dimensionality of the images. Many previous methods [3,4,14] have used a color histogram as the input data; however, for TPS we use image thumbnails as input. Initial tests indicated that TPS worked as well or better based on thumbnails than it did on histograms, so all the results reported here are based on thumbnails as input. The thumbnails are 8×8 images created by averaging the underlying pixels in the original input image. These thumbnails in chromaticity coordinates become input vectors of size $8 \times 8 \times 2 = 128$. Therefore, the dimensionality of the input data is $D = 128$. Our tests in Section 4 show that thumbnails with size 8×8 are sufficient for use in estimating the illumination.

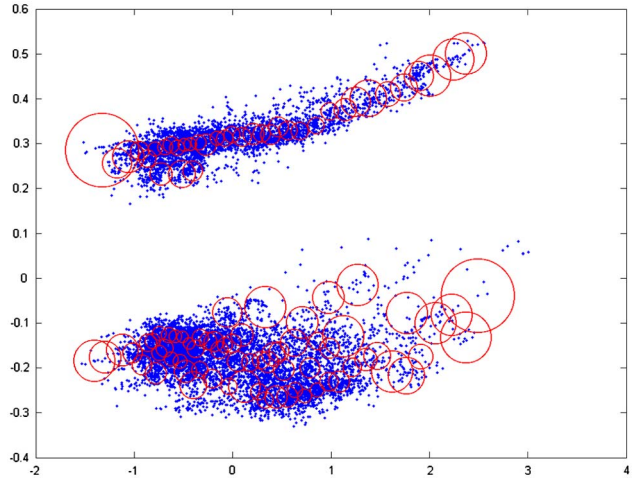

Fig. 1. (Color online) k-median (with $k = 220$) clustering of the 11,346 training real-world images [13]. The x axis and y axis stand for the first and second principal component vectors. Each red circle represents a cluster detected. Circle size is proportional to the standard derivation of the cluster. This shows graphically that the 220 colors cover the underlying data set quite well.

4. TESTS

We implemented the TPS illumination-estimation method and incremental k medians in Matlab and conducted tests to compare its performance to that of other illumination-estimation methods.

A. Evaluation Measures

Several different error measures are used to evaluate performance. For each image, the distance between the measured actual illumination chromaticity (r_a, g_a) and that estimated by an algorithm (r_e, g_e) is calculated as

$$E_{L2\text{-dist}} = [(r_a - r_e)^2 + (g_a - g_e)^2]^{1/2}, \tag{8}$$

For a test set of M images, we report the root mean square (RMS), median, and maximum (Max) distances [15]. Additionally, we report the mean $L2$ distance of the best 75% (labeled "B75" in the tables) and mean of the worst 25% (labeled "W25") estimates. RMS is defined in the standard way as

$$\text{RMS} = \frac{1}{M}\sqrt{\sum_{i=1}^{M} E_{L2\text{-dist}}^2} \tag{9}$$

Given illumination chromaticity components r and g, the third component can be obtained as $b = 1 - r - g$ from which the angular error in degrees between two three-dimensional (3D) chromaticity vectors is defined as

$$E_{\text{Ang}} = \cos^{-1}\left[\frac{(r_a, g_a, b_a)\cdot(r_e, g_e, b_e)}{\sqrt{r_a^2 + g_a^2 + g_a^2}\cdot\sqrt{r_e^2 + g_e^2 + g_e^2}}\right]\cdot\frac{2\pi}{360}. \tag{10}$$

In the case that one of the three components of our estimate is less than zero, it is truncated to zero. As with the distance measure, we also compute the median, RMS, and Max angular error over the test set of images.

To evaluate whether there is a significant difference in the performance of competing methods, the Wilcoxon signed rank is applied [15]. The threshold for rejecting the null hypothesis is set to the 5% significance level.

B. Tests of Training Based on Real Images

The first of the tests with real images used for training is based on Barnard's calibrated 321 linear (gamma = 1) SONY images [16]. The illumination error is evaluated using both leave-one-out cross-validation [16] and threefold cross-validation [17]. In the leave-one-out procedure, one image is selected for testing, and the remaining 320 images are used for training in order to determine the required parameters. This is repeated 321 times, with a different image left out each time. In the threefold procedure, one third of the images are randomly selected for testing, and the remaining two thirds are used for training. This is repeated three times, with a

Table 1. Performance Comparison of MaxRGB [18] and MaxRGB with Preprocessing of the Images (MaxRBG+) by Bicubic Resizing to 64 × 64 [19], GrayWorld (GW)[20], 3D Support Vector Regression (SVR) [4], Shades of Gray (SoG) [21], Edge-Based [22], Gray Surface Identification (GSI) [23], Color by Correlation (CbyC) [24], Gamut Mapping [25], *N*-Jet [26], and TPS[a]

Method	Angular Error			*L*-2 Distance ($\times 10^2$)				
	Median	RMS	Max	Median	B75	RMS	W25	Max
Do-nothing	15	19	11	11	11	13	24	24
MaxRGB	6.5	12	36	4.5	3.7	8.3	11	25
MaxRGB+ (Table 1 in [19])	3.2	8.6	27	2.3	1.7	5.8	10	17
GW	7.0	13	36	5.6	4.0	11	19	33
Database GW (Table 7 [24])	6.9	12	-	-	-	-	-	-
SoG ($p = 6$)	4.1	9.0	29	2.8	2.3	6.2	11	20
Edge-based (first order)	3.7	8.5	28	2.6	2.3	6.0	10	19
Edge-based (second order)	4.5	9.1	36	3.1	2.8	6.2	11	24
GSI	3.9	10	34	2.7	2.3	7.2	13	23
CbyC I (bright pixels only) Hordley and Finlayson (Table 7 in [24]) (310 out of 321)	3.2	10	-	-	-	-	-	-
CbyC Gijsenij *et al.* (Table 3 in [26]) (290 out of 321)	6.8	-	-	-	-	-	-	-
Gamut Mapping Gijsenij *et al.* (Table 3 in [26]) (290 out of 321)	3.1	-	-	-	-	-	-	-
N-jet (complete 1-jet) (Table 3 in [26]) (290 out of 321)	2.1	-	-	-	-	-	-	-
SVR (3D)	2.2	8.0	25	3.1	-	3.5	-	11
TPS (leave-one-out)	0.6	2.1	10	0.6	0.5	1.6	2.7	7.2
TPS (threefold cross-validation)	1.2	3.6	23	1.0	0.8	2.9	4.9	14

[a]Results involve real-data training and testing using the 321 SONY images. For TPS the images are converted to 8×8 thumbnails; the other algorithms use the original images. Errors for Color by Correlation and Support Vector Regression are based on leave-one-out cross-validation. Errors for TPS are based on both leave-one-out cross-validation and threefold cross-validation and are reported in terms of both the angular and distance error measures. See Subsection 4.B for error labels.

different subset selected for testing each time. In Tables 1 and 2, the TPS results along with corresponding results of the following methods are listed: MaxRGB [18], MaxRGB with preprocessing [19], GrayWorld [20], Shades of Gray [21] (Minkowski norm = 6), first- and second-order edge-based approaches [22] (Minkowski norm = 6, sigma = 2), Gray Surface Identification [23], Color by Correlation [24], Gamut Mapping [25], and *N*-jet [26] and Support Vector Regression [4].

Second, we test our method on the Gehler *et al.* [17] "ColorChecker" data set. The ColorChecker data set contains 568 images taken with two high-quality digital single-lens reflex cameras (Canon 5D and Canon1D) with all settings in auto mode. All images were saved in Canon RAW format. As well, the data set includes TIFF versions created from the RAW images using the automatic mode of the Canon Digital Photo Professional program to convert the images into TIFFs. Each image contains a Macbeth ColorChecker for reference. The image coordinates (measured by hand) of each ColorCheckers' squares is provided with the data set [17].

Because the TIFF images in the ColorChecker data set were produced automatically, they contain clipped pixels, are nonlinear (i.e., have gamma or tone curve correction applied), are demosaiced, and include the effect of the camera's white balancing. To avoid these problems, we chose to reprocess the raw data and created almost-raw 12 bit PNG format (lossless compression) images from the Canon RAW format data by decoding them using dcraw [27]. To preserve the original digital counts for each of the RGB channels, demosaicing was not enabled. The cameras both output 12 bit data per channel, so the range of possible digital counts is 0 to 4095. The raw images contain 4082×2718 (Canon 1D) and 4386×2920 (Canon 5D) 12 bit values in an RGGB pattern. To create a color image, the two *G* values were averaged, but no further demosaicing was done. This results in a 2041×1359 (for Canon 1D) or 2193×1460 (for Canon 5D) linear image (gamma = 1) in camera RGB space. The Canon 5D has a black level of 129, which was subtracted. The Canon 1D's black level is zero.

Table 2. Comparison of Several Different Algorithms in Table 1 via the Wilcoxon Signed-Rank Test with Rejection of the Null Hypothesis at the 5% Significance Level[a]

	MaxRGB	MaxRGB+	GW	SoG	Edge1	Edge2	CbyC	SVR	GSI	TPS
MaxRGB	=	−	=	−	−	−	−	−	−	−
MaxRGB+	+	=	+	+	+	+	−	=	+	−
GW	=	−	=	−	−	−	−	−	−	−
SoG	+	−	+	=	=	+	=	=	=	−
Edge1	+	−	+	=	=	+	=	=	=	−
Edge2	+	−	+	−	−	=	−	=	=	−
CbyC	+	+	+	=	=	+	=	=	=	−
SVR	+	=	+	=	=	=	=	=	=	−
GSI	+	−	+	=	=	=	=	=	=	−
TPS	+	+	+	+	+	+	+	+	+	=

[a]Here CbyC represents "Color by Correlation using bright pixels only" by Hordley and Finlayson [24]. A "+" means the algorithm listed in the corresponding row is better than the one in corresponding column. A "−" indicates the opposite, and an "=" indicates that the performance of the respective algorithms is statistically equivalent. (Since TPS by leave-one-out and threefold validation ranked the same in the Wilcoxon test, they are not listed separately in the table.)

Table 3. Performance Comparison of MaxRGB [18] and MaxRGB with Preprocessing (Labeled MaxRGB+) of the Images by Bicubic Resizing to 64 × 64 [19], GrayWorld [20], Shades of Gray [21], Edge-Based [22], Color by Correlation [24], Gamut Mapping [25], *N*-Jet [26], Bayes-GT (with Threefold Cross-Validation) [17], and TPS[a]

	Angular Error			*L*2 Distance (×10²)				
	Median	RMS	Max	Median	B75	RMS	W25	Max
Do-nothing	4.8	13	37	3.1	3.0	9.3	30	30
GW	3.7	6.2	25	2.6	2.1	4.5	7.5	20
MaxRGB	9.1	13	51	7.8	5.9	12	19	55
MaxRGB+	3.4	8.0	33	2.5	2.0	6.5	12	30
SoG ($p = 6$)	4.5	8.7	36	3.5	2.9	7.5	13	37
Edge-based (first order)	3.8	9.4	38	3.0	2.7	8.0	14	40
Edge-based (second order)	4.4	10	47	3.5	3.2	8.7	15	50
Gamut Mapping(full data set training)	4.3	8.4	32	3.2	2.6	6.8	12	24
N-jet (complete one-jet)	4.2	8.2	32	3.2	2.6	6.5	11	24
N-jet (complete two-jet)	4.1	8.0	32	3.1	2.5	6.3	11	24
Bayes-GT	5.8	8.9	34	5.0	3.8	7.3	12	28
TPS (threefold)	2.8	4.6	17	2.1	1.6	3.4	5.6	16
TPS (leave-one-out)	2.4	4.1	19	1.7	1.4	3.1	5.0	13

[a]Results involve real-data training and testing using the 568 Canon images in the ColorChecker data set [17]. For TPS the images are converted to 8 × 8 thumbnails; the other algorithms use the original images. Sometimes the Gamut Mapping and one-jet methods fail to provide an illumination estimate (four times for Gamut Mapping, one for one-jet). In such cases, we assign the illumination estimate as white with chromaticity $(1/3, 1/3)$. The TPS errors are based on leave-one-out cross-validation and threefold cross-validation. (Gamut Mapping was trained using the entire data set.)

Table 4. Comparison of Several Different Algorithms in Table 3 via the Wilcoxon Signed-Rank Test with Rejection of the Null Hypothesis at the 5% Significance Level[a]

	MaxRGB	MaxRGB+	GW	SoG	Edge1	Edge2	Gamut	One-Jet	Two-Jet	Bayes-GT	TPS
MaxRGB	=										
MaxRGB+	+	=									
GW	+	−	=								
SoG	+	−	−	=							
Edge1	+	−	−	+	=						
Edge2	+	−	−	+	−	=					
Gamut Mapping	+	−	−	+	−	+	=				
One-jet	+	−	−	+	−	+	+	=			
Two-jet	+	−	−	+	−	+	+	+	=		
Bayes-GT	+	−	−	−	−	=	−	−	−	=	
TPS	+	+	+	+	+	+	+	+	+	+	=

[a]A "+" means the algorithm listed in the corresponding row is better than the one in corresponding column. A "−" indicates the opposite, and an "=" indicates that the performance of the respective algorithms is statistically equivalent. (Since TPS evaluated either by leave-one-out or threefold cross-validation ranks the same in the Wilcoxon test, they are not listed separately in the table.)

The ColorChecker has six achromatic squares. We used the median of the RGB digital counts from the brightest achromatic square containing no digital count >3300 as the ground-truth measure of the illumination's chromaticity. The threshold eliminates any clipping or possible nonlinearity as the intensities approach the maximum of 4095. The median

Table 5. Performance Comparison of MaxRGB [18], MaxRGB with Preprocessing (Labeled MaxRGB+) of the Images by Bicubic Resizing to 64 × 64 [19], GrayWorld [20], Shades of Gray [21], Edge-Based [22], and TPS[a]

	Angular Distance			*L*2 Distance (×10²)				
Methods	Median	RMS	Max	Median	B75	RMS	W25	Max
Do-nothing	17	19	37	16	13	15	16	17
MaxRGB	7.4	13	51	5.6	4.9	11	18	55
MaxRGB+	3.3	8.2	33	2.4	1.9	6.2	11	30
GW	4.2	9.5	36	3.3	2.6	7.4	12	33
SoG	4.3	8.8	36	3.1	2.6	7.1	12	37
Edge1	3.8	9.1	38	2.8	2.5	7.3	13	40
Edge2	4.4	9.7	47	3.3	3.0	7.9	14	50
TPS (threefold)	2.4	4.7	33	1.7	1.4	3.3	5.6	20
TPS (leave-one-out)	1.8	4.0	25	1.4	1.1	2.9	4.8	15

[a]Results involve real-data training and testing using the combination of the 321 SFU Sony images [16] combined with the 568 Canon images of the ColorChecker data set [17]. For TPS the images are converted to 8 × 8 thumbnails; the other algorithms use the original images. The TPS errors are based on leave-one-out cross-validation and threefold cross-validation.

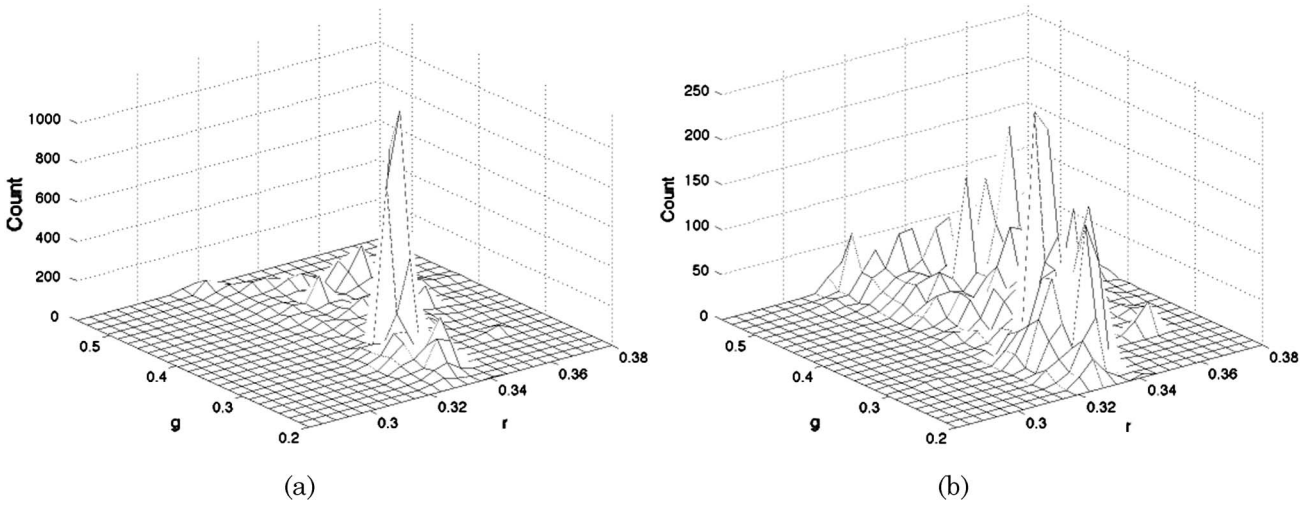

Fig. 2. (a) Histogram of the measured illumination chromaticities from the full set of 11,346 images showing a distinct peak around gray (0.33, 0.33); (b) corresponding histogram for the reduced data set of 7661 images showing a more uniform distribution of illumination chromaticities. The x axis is the chromaticity r, the y axis is the chromaticity g, and the z axis is the count of the number of the same illuminations.

was chosen instead of the mean because the median automatically excludes any of the black pixels surrounding each square that might have been incorrectly included in the square due to the inexactness in the hand labeling of a ColorChecker's position.

For testing, the ColorCheckers were overwritten by zeros in all images. TPS's performance was evaluated using both the leave-one-out [16] and threefold cross-validation. Tables 3 and 4 provide a comparison to the performance of MaxRGB [18], MaxRGB with preprocessing [19], GrayWorld [20], Shades of Gray [21] (Minkowski norm = 6), first- and second-order edge-based approaches [22] (Minkowski norm = 6, sigma = 2), Gamut Mapping [25], N-jet [26] and Bayes-GT [17]. To give the Gamut Mapping and the two N-jet methods the best possible training data, the full set of 568 images with the ColorCheckers *included* was used. Testing was then done on images with the ColorCheckers removed. Since this data set consists of images from two different camera models, training and testing were done separately for each of the two corresponding image subsets. The results were then combined. For Bayes-GT [17], it would also have been interesting to be able to include leave-one-out results in Tables 3 and 4 as well, but unfortunately that method is so computationally intensive that this is not practical. Bayes-GT requires 4 min per image so even the threefold validation required two days of computation. It seems doubtful that it could ever be sped up enough to be of any more than theoretical interest.

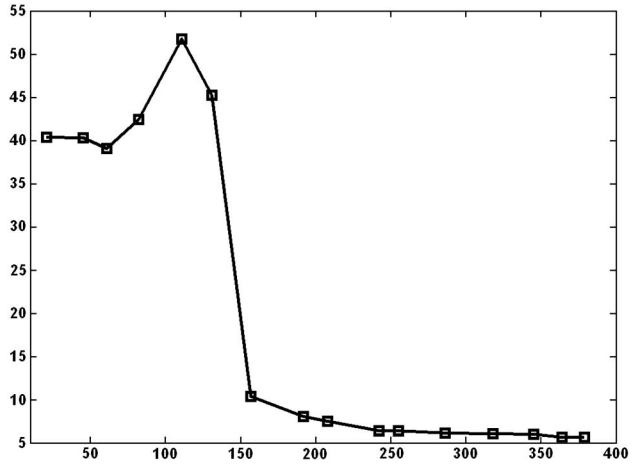

Fig. 3. Plot of the median angular error (y) from Table 9 (excluding the last row) as a function of the size of the set of representative training images (x axis).

As a third test, we combine the 321 images of the Simon Fraser University (SFU) Sony data set [16] with the 568 Canon images of ColorChecker data set [17] into a single data set. The results are shown in Tables 5 and 6. These two data sets are similar in size so neither set will dominate. They are also from completely different cameras. As the SFU data set includes camera calibration data, it is possible to balance the sensor sensitivities so that $R = G = B$ for an ideal white reflectance under the canonical light source. On the other hand, the sensor sensitivities of the two Canon cameras of the ColorChecker data set were unknown, so the sensors were left unbalanced. Tables 5 and 6 compare the performance on the combined data set of 889 images of MaxRGB [18], MaxRGB with preprocessing (MaxRGB+) [19], GrayWorld [20], Shades of Gray [21] (Minkowski norm = 6), first- and second-order edge-based approaches [22] (Minkowski norm = 6, sigma = 2), and TPS by leave-one-out and threefold cross-validation.

As a final test with real images, we use a subset of 7661 out of the 11,346 real images of the Ciurea and Funt [13] Grayball data set. A matte gray ball appears at a fixed location near the right-bottom corner of each image of the data set and provides a measure of the illumination chromaticity. To ensure that the gray ball has no effect on our results, all images were cropped on the right to remove the gray ball. The resulting images are 240 by 240 pixels. Each image was converted to a 128-dimensional vector by shrinking the images into 8×8 thumbnails in chromaticity space. In this data set, the images are nonlinear; however, the actual gamma or tone curve used is unknown.

The 7661 images making up the subset were selected to give a somewhat uniform distribution of the rg-chromaticity values of the scene illuminants than occurs in the full data set. Many of the images in the full data set already have very good color balance (i.e., the gray ball has $R = G = B$), which potentially could bias the results. Therefore, we eliminated from the data

Table 6. Comparison of Several Different Algorithms in Table 5 via the Wilcoxon Signed-Rank Test with Rejection of the Null Hypothesis at the 5% Significance Level[a]

	MaxRGB	MaxRGB+	GW	SoG	Edge1	Edge2	TPS
MaxRGB	=						
MaxRGB+	+	=					
GW	+	−	=				
SoG	+	−	−	=			
Edge1	+	−	+	=	=		
Edge2	+	−	−	−	−	=	
TPS	+	+	+	+	+	+	=

[a]A "+" means the algorithm listed in the corresponding row is better than the one in corresponding column. A "−" indicates the opposite, and an "=" indicates that the performance of the respective algorithms is statistically equivalent. (Since TPS by leave-one-out and threefold validation ranked the same, they are not listed separately in the table.)

Table 7. Performance Comparison of MaxRGB [18] and MaxRGB with Preprocessing of the Images by Bicubic Resizing to 64 × 64 [19], GrayWorld [20], 3D Support Vector Regression [4], Shades of Gray [21], Edge-Based [22], Gray Surface Identification [23], Color by Correlation [24], *N*-Jet [26], and TPS[a]

	Angular Error			*L2* Distance ($\times 10^2$)				
	Median	RMS	Max	Median	B75	RMS	W25	Max
Do-nothing	10	13	27	7.2	7.7	9.4	22	22
MaxRGB	9.4	12	27	6.6	5.1	8.7	14	22
MaxRGB+	8.1	11	30	5.9	4.7	8.2	13	22
GW	6.6	9.4	44	5.1	3.9	7.6	12	40
SoG ($p = 6$)	6.6	8.8	36	4.7	3.8	6.5	10	28
Edge-based (first order)	7.2	10	33	5.2	4.2	7.5	12	23
Edge-based (second order)	7.8	11	34	5.6	4.5	7.9	13	25
CbyC (Table IV, Gijsenij [26])	8.5	11	39	-	-	-	-	-
Leave-*N*-Out *n*-jet (Table IV, Gijsenij [26])	6.5	8.8	43	-	-	-	-	-
GSI	5.5	8.0	39	4.2	3.3	6.2	10	32
3D SVR	4.9	7.0	25	3.6	-	5.2	-	19
TPS	4.6	6.9	34	3.4	2.7	5.1	8.3	26

[a]Results involve real-data training and testing on disjoint sets of 7661 images from the Ciurea data set.

set the majority of the correctly balanced images in order to make the overall distribution of the illumination chromaticity more uniform. The distributions of the illuminant chromaticity of the full and reduced sets are shown in Fig. 2. The resulting data set contains 7661 images.

Since neighboring images in the database tend to be related, we partitioned the 7661 images into two independent sets based on geographical location. Subset A includes 3581 images, and subset B includes 4080. Subset A contains images from the Apache Trail, Burnaby Mountain, Camelback Mountain, CIC 2002, and Deer Lake. Subset B contains images from completely different locations including False Creek, Granville Island Market, Marine Drive, Metrotown shopping center, Scottsdale, Simon Fraser University, and Whiteclyff Park. We use Subset A for training and B for testing and then vice versa. Since the training and test sets are geographically distinct, any similarity between the images in the two sets is guaranteed to be coincidental and not a by-product of the fact that they are extracted from a video sequence. The combined errors and corresponding Wilcoxon signed-rank test results from both tests are shown in Tables 7 and 8.

Table 8. Comparison of the Performance Based on the Wilcoxon Signed-Rank Test with Rejection of the Null Hypothesis at the 5% Significance Level[a]

	MaxRGB	MaxRGB+	GW	SoG	Edge1	Edge2	GSI	CbyC	SVR	*N*-Jet	TPS
MaxRGB	=										
MaxRGB+	+	=									
GW	+	+	=								
SoG	+	+	+	=							
Edge1	+	+	−	−	=						
Edge2	+	=	−	−	−	=					
GSI	+	+	+	+	+	+	=				
CbyC	+	−	−	−	−	−	−	=			
SVR	+	+	+	+	+	+	+	+	=		
N-jet	+	+	+	=	+	+	−	+	−	=	
TPS	+	+	+	+	+	+	+	+	+	+	=

[a]Labeling "+," "−," "=" as in Table 7.

Table 9. Median Angular Error of TPS Illumination Estimates Taken over 4080 Images along with Training and Test Times as a Function of the Size of the Reduced Training Set[a]

Number of Clusters k	Median Angular Error	Training Time (Seconds)	Averaging Test Time per Image (Milliseconds)
21	40	0.06	1.60
45	40	0.09	2.27
61	39	0.11	2.34
82	43	0.15	3.06
111	52	0.29	3.62
131	45	0.37	4.52
157	10	0.51	5.12
192	8.0	0.76	5.99
208	7.5	0.92	6.03
242	6.4	1.24	6.94
255	6.4	1.33	6.77
286	6.1	1.67	7.26
318	6.1	2.07	7.04
345	6.0	2.43	7.42
364	5.7	2.73	7.69
379	5.7	2.94	7.84
3581	**4.8**	283.18	67.24

[a]The last row ($k = 3581$) means that all the images in subset A were used for training. The angular error is plotted in Fig. 3.

C. Tests on Reduced Training Data Set

TPS training requires a matrix inverse operation of complexity $O((N+D)^3)$, and TPS testing has complexity $O(\log ND)$. Since both grow with the training size N while the dimensionality D of the input image is fixed (128 in our tests), the way to reduce the computation time is to reduce the training size. To reduce the training-set size without losing important training information, we select a subset of the training set using k-median clustering. Starting with the 7761 images [13] described above and with increasing values of k, we select the k centroids from subset A and then test using all images in subset B. The median angular errors are listed in Table 9. The time recorded is CPU time on a Windows computer equipped with a 3 GHz CPU and 1 GB RAM. The plot in Fig. 3 shows that there is significant performance penalty incurred for training sizes under 150. Table 9 shows that, although there is a slight performance penalty in reducing the training set from $k = 3581$ (error is 4.8) to $k = 364$ (error is 5.7) images, the training time drops from 283.18 to 2.73 s and the testing time for each image from 67.24 to 7.69 ms—factors of approximately 100 and 10, respectively. Additionally, the computer memory required for training is reduced from $\sim(D+N)^2$ to $\sim(D+k)^2$. In this case, for $D = 128$, $N = 3581$, and $k = 364$, only 2% of memory is required for the reduced set in comparison to the full set.

The results of the above tests demonstrate that training-set reduction via k-median clustering significantly decreases the time required both for training and for testing (i.e., illumination estimation), with only a modest sacrifice in accuracy, even when the subset contains only 364 centroids from the full training set. As both the more time-consuming phases —incremental k medians and TPS training—are only done once and can be completed offline, TPS-based illumination estimation is potentially suitable for practical applications since the time to evaluate an individual image is only a few milliseconds.

5. DISCUSSION AND CONCLUSIONS

The problem of estimating the chromaticity of the overall scene illumination is formulated in terms of interpolation over a nonuniformly sampled data set. The illumination chromaticity is viewed as a function of the image, and the set of training images is nonuniformly spaced. TPS interpolation is an excellent interpolation technique for these conditions and is shown here to work well for illumination estimation. TPS calculates its result based on a weighted combination of the entire set of training data. For a modest sacrifice in accuracy the training and test time can be significantly reduced by using incremental k medians to select a smaller, representative set of training images from a given training set.

Although it might be argued that the superior performance of TPS is due to the fact that some of the scenes in the data sets are highly correlated, as they are in the 321 data set, this cannot be the explanation for the other data sets. For the 568 data set, the images are still photographs from varied locations and appear quite independent. Since it originated from digital video, the 7661 data set may contain highly correlated images [28], but any correlation cannot be between the images in the training and test sets because they were from different geographical locations. Based on the results of the tests (shown in Tables 1–9) on three different real-image data sets and one combined data set, TPS consistently demonstrates statistically significant superior performance compared to all the other methods tested, including those that are known as some of the best available.

ACKNOWLEDGMENTS

This research was funded by the Samsung Advanced Institute of Technology and the Natural Sciences and Engineering Research Council of Canada. The authors wish to thank Arjan Gijsenij for providing the illumination estimates from the N-jet and Color-by-Correlation algorithms for the Ciurea data set.

REFERENCES

1. G. D. Knott, *Interpolating Cubic Splines* (Birkhauser, 2000).
2. F. L. Bookstein, "Principal warps: thin-plate splines and decomposition of deformations," IEEE Trans. Pattern Anal. Mach. Intell. **11**, 567–585 (1989).
3. V. Cardei, B. Funt, and K. Barnard, "Estimating the scene illumination chromaticity using a neural network," J. Opt. Soc. Am. A **19**, 2374–2386 (2002).
4. B. V. Funt and W. H. Xiong, "Estimating illumination chromaticity via Support Vector Regression," J. Imaging Sci. Technol. **50**, 341–348 (2006).
5. M. H. Davis, A. Khotanzad, D. Flamig, and S. Harms, "A physics-based coordinate transformation for 3-d image matching," IEEE Trans. Med. Imaging **16**, 317–328 (1997).
6. N. Arad and D. Reisfeld, "Image warping using few anchor points and radial functions," Comput. Graph. Forum **14**, 35–46 (1995).
7. A. M. Bazen and S. H. Gerez, "Elastic minutiae matching by means of thin-plate spline models," in *Proceedings of 16th International Conference on Pattern Recognition* (IEEE, 2002), pp. 985–988.
8. W. Xiong and B. V. Funt, "Nonlinear RGB-to-XYZ mapping for device calibration," in *Proceedings of Imaging Science and Technology Thirteenth Color Imaging Conference* (Society for Imaging Sciences and Technology, 2005), pp. 200–204.
9. A. Zandifar, S. Lim, R. Duraiswami, N. Gumerov, and L. S. Davis, "Multi-level fast multipole method for thin plate spline evaluation," in *Proceedings of IEEE International Conference on Image Processing* (IEEE, 2004), pp. 1683–1686.

10. F. L. Bookstein, *Morphometric Tools for Landmark Data Geometry and Biology* (Cambridge University, 1991).
11. J. Meinguet, "Multivariate interpolation at arbitrary points made simple," Z. Angew Math. Phys. **30**, 292–304 (1979).
12. R. Gershon, A. D. Jepson, and J. K. Tsotsos, "From $[R,G,B]$ to surface reflectance: computing color constant descriptors in images," in *Proceedings of the 10th International Joint Conference on Artificial Intelligence* (Morgan Kaufmann, 1987), pp. 755–758.
13. F. Ciurea and B. Funt, "A large image database for color constancy research," in *Proceedings of IS&T/SID Eleventh Color Imaging Conference* (Society for Imaging Science and Technology, 2003), pp. 160–163.
14. G. D. Finlayson, S. D. Hordley, and P. M. Hubel, "Color by correlation: a simple, unifying framework for color constancy," IEEE Trans. Pattern Anal. Mach. Intell. **23**, 1209–1221 (2001).
15. S. D. Hordley and G. D. Finlayson, "Re-evaluating colour constancy algorithms," in *Proceedings of 17th International Conference on Pattern Recognition* (IEEE, 2004), pp. 76–79.
16. K. Barnard, L. Martin, A. Coath, and B. Funt, "A comparison of computational color constancy algorithms. Part two: experiments on image data," IEEE Trans. Image Process. **11**, 985–996 (2002).
17. P. V. Gehler, C. Rother, A. Blake, T. P. Minka, and T. Sharp, "Bayesian color constancy revisited," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 2008), pp. 1–8.
18. G. D. Finlayson, "Retinex viewed as a Gamut Mapping theory of color constancy," in *Proceedings of AIC International Color Association Color 97* (International Color Association, 1997), Vol. 2, pp. 527–530.
19. B. Funt and L. Shi, "The rehabilitation of MaxRGB," in *Proceedings of the Eighteenth IS&T Color Imaging Conference* (Society for Imaging Science and Technology, 2010), pp. 256–259.
20. B. Buchsbaum, "A spatial processor model for object color perception," J. Franklin Inst. **310**, 1–26 (1980).
21. G. D. Finlayson and E. Trezzi, "Shades of Gray and colour constancy," in *Proceedings of IS&T/SID Twelfth Color Imaging Conference: Color Science, Systems and Applications* (Society for Imaging Science and Technology, 2004), pp. 37–41.
22. J. van de Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," IEEE Trans. Image Process. **16**, 2207–2214 (2007).
23. W. Xiong, B. Funt, L. Shi, S. Kim, B. Kang, and S. D. Lee, "Automatic white balancing via Gray Surface Identification," in *Proceedings of the Fifteenth IS&T Color Imaging Conference* (Society for Imaging Science and Technology, 2007), pp. 143–146.
24. S. Hordley and G. Finlayson, "A reevaluation of color constancy algorithm performance," J. Opt. Soc. Am. A **23**, 1008–1020 (2006).
25. G. Finlayson, S. Hordley, and I. Tastl, "Gamut constrained illumination estimation," Int. J. Comput. Vis. **67**, 93–109 (2006).
26. A. Gijsenij, T. Gevers, and J. van deWeijer, "Generalized Gamut Mapping using image derivative structures for color constancy," Int. J. Comput. Vis. **86**, 127–139 (2008).
27. D. Coffin, "Dcraw," http://en.wikipedia.org/wiki/Dcraw.
28. S. Bianco, G. Ciocca, C. Cusano, and R. Schettini, "Improving color constancy using indoor–outdoor image classification," IEEE Trans. Image Process. **17**, 2381–2392 (2008).