Journal of Imaging Science and Technology® 57(5): 050501-1–050501-12, 2013.

Illumination Chromaticity Estimation Using Bayesian Kernel Methods

Xiangkun Zhao
Beijing Key Lab of Traffic Data Analysis and Mining, School of Computer and Information Technology,
Beijing Jiaotong University, Beijing, China
School of Biomedical Engineering, Capital Medical University, Beijing, China
E-mail: xkzhao.gm@gmail.com

Jian Yu
Beijing Key Lab of Traffic Data Analysis and Mining, School of Computer and Information Technology,
Beijing Jiaotong University, Beijing, China

Bangjun Wang
School of Computer Science and Technology, Soochow University, Suzhou, China

Abstract. *In this article, two Bayesian kernel methods, namely the Gaussian process regression (GPR) and relevance vector machine (RVM) techniques, are used to estimate illumination chromaticity and predict the reliability of the estimation process, which is not accessible for most machine learning techniques that have been used for color constancy. More than seven kinds of GPR covariance function and their combinations, and an RVM method using Gaussian, Laplace and Cauchy kernel functions, have been used on two real image sets. The experimental results show that the GPR method outperforms those based on RVM and ridge regression using stationary covariance functions, and GPR can almost achieve the same performance as support vector regression (SVR). The performance of the RVM for regression is almost the same as that of GPR using the dot product covariance function. The influence of outliers on the data with Gaussian noise is analyzed in detail via using heavy-tailed Laplace and Student-t kernel functions when GPR and the RVM are used for color constancy.* ©*2013 Society for Imaging Science and Technology.*

[DOI: 10.2352/J.ImagingSci.Technol.2013.57.5.050501]

INTRODUCTION

The color of an object varies under different illuminants in most circumstances. Color constancy is used to recognize the object color regardless of the light source, which is important for many computer vision applications, such as image retrieval, image classification, color object recognition, and video tracking.

For a Lambertian reflectance model, a color image $\mathbf{f} = (f_R, f_G, f_B)$ is composed of the multiplication of three terms, i.e. the color of the light source $e(\lambda, \mathbf{x})$, the surface reflectance properties $s(\lambda, \mathbf{x})$ and the camera sensitivity function $\mathbf{c}(\lambda)$:

$$\mathbf{f} = m_b \int_{\omega} e(\lambda, \mathbf{x}) s(\lambda, \mathbf{x}) \mathbf{c}(\lambda) d\lambda \qquad (1)$$

where m_b is the scale factor for shading, ω is the visible spectrum, $\mathbf{c}(\lambda) = (R(\lambda), G(\lambda), B(\lambda))$, λ is the wavelength of the light and $\mathbf{x}$ is the spatial coordinate in the image. Both the intrinsic property of a surface $s(\lambda, \mathbf{x})$ and the color of the illuminant $e(\lambda, \mathbf{x})$ have to be estimated, while only the product (i.e. the actual image $\mathbf{f}$) is known. This implies that illuminant estimation is an under-constrained problem. To obtain an object's color feature irrespective of the light source, some assumptions must be made or constraints imposed as regards the light source or object surface feature.

There have been many techniques used in addressing the color constancy problem. They can be categorized into three kinds: low-level feature based methods,[1–4] statistics based methods[5–10] and machine learning based methods.[11–18] Most methods can achieve good estimation of illumination chromaticity, but none of them can predict the reliability of the estimation process. The latter is very important for video tracking and video surveillance for unknown light scenes.

Bayesian kernel methods[19–22] have been proved to be effective tools for regression and classification. In this article, Bayesian kernel methods, including Gaussian process[23,19] and relevance vector machine (RVM)[20–22] ones, are used for the color constancy problem. The purpose of using Bayesian kernel methods for color constancy is rooted in three aspects. The first is that the Bayesian approach allows for an intuitive incorporation of prior knowledge into the process of estimation, and it not only can achieve illumination chromaticity estimation but also can make predictions about the reliability of the estimation process. The second aspect is that a cross-validation procedure must be used to avoid over-fitting for most machine learning methods, which is wasteful of both data and computation. The fully probabilistic framework is adopted for both methods and a prior over the model is governed by a set of hyperparameters; therefore, cross-validation is not needed. The third aspect is that kernel functions provide a powerful way of detecting the nonlinear relations alluded to above in relation to the

Received Oct. 12, 2012; accepted for publication Aug. 18, 2013; published online Sep. 1, 2013. Associate Editor: Jon Yngve Hardeberg.
1062-3701/2013/57(5)/050501/12/$20.00

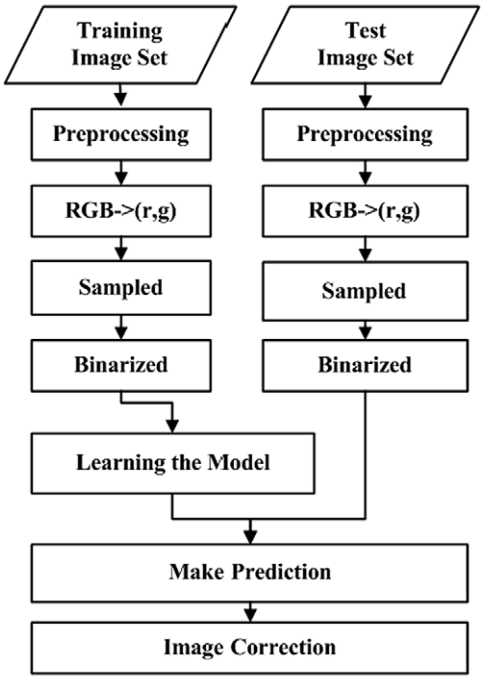


Figure 1. Illustration of the algorithm for color constancy.

color constancy problem. The approach decouples the design of the algorithm from the specification of the feature space, which not only increases the flexibility of the approach but also makes both the learning algorithms and the kernel design more amenable to formal analysis.[24]

In this article, the procedure of the Bayesian kernel based method for color constancy is described, see Figure 1. Firstly, all the images are preprocessed to remove dark pixels and filtered to reduce noise. Secondly, the color image is converted into the chromaticity space (r, g), sampled into $N \times N$ bins, and binarized to form the input. After that, the estimation of illumination chromaticity is implemented on the training image set to select the best model. Finally, the estimation of the illumination chromaticity of an unknown image is obtained using the selected best model and image correction is implemented. From Fig. 1, we can see that the Bayesian kernel method uses all the images with light ground-truth in the data set for training without the wasteful validation data needed by most machine learning methods.

The performances of both Bayesian kernel based methods—ridge regression (RR)[25] and support vector regression (SVR)[26]—are evaluated on two real image sets from Shi[27] and Bianco.[28] On the basis of the experimental results, it is shown that GPR outperforms RVM for regression when stationary covariance functions are used, and can almost achieve the same performance as SVR. The performance of RVM is almost the same as that of GPR when dot product covariance functions are used. The performance of the ridge regression method is poorer than those of three kernel based methods (the support vector machine is also kernel based). Finally, the outlier influence on the data with Gaussian noise is analyzed in detail when GPR and RVM are used for color constancy.

The rest of this article is organized as follows. An overview of related work is given in the second section. The third section outlines the algorithms of GPR and RVM for regression and the fourth section gives the data representation for regression based methods of color constancy. Two real image sets and error metrics are described in the fifth section. The sixth section gives experimental results, analysis of both methods, and comparisons with other methods. Seventh section gives some sample results from various methods applied to images with light ground-truth and unseen images without light ground-truth using optimized parameters. The outlier influences on GPR and RVM are discussed in detail in the eighth section. Finally, conclusions are drawn in last section.

RELATED WORKS

There have been a lot of techniques suggested for addressing the color constancy problem. Techniques of the first kind are based on low-level features of the image. The white patch (WP) method[1] assumes that the maximum response in an image is caused by a perfect reflectance and the maximum response in each channel is the color of the light source. The gray world (GW) method[2] assumes that the average color in a scene is achromatic while the gray edge (GE)[3] algorithm assumes that the derivative of the image rather than the pixel is achromatic. A general framework which unifies WP, GW, the general gray world (GGW) method,[4] and the GE method is proposed by Van De Weijer.[3] Some higher-order statistical methods, like the Gamut mapping method[5–7] and the Bayesian color constancy method,[8–10] use statistical models to quantify the probability of each illuminant and then make an estimation from these probabilities. The third category for addressing the color constancy problem is that of the machine learning based methods, which learn the dependence between the object color and illuminant color from the training data. Cardei et al.[11,12] firstly used a multi-layer neural network to recover the illumination chromaticity given only an image of the scene and showed that neural networks achieved better color constancy than the color-by-correlation algorithm.[13] Other authors[14–16] used SVR for color constancy. Agarwal et al.[16] explained that the linear machine learning methods, such as ridge regression[16,17] and kernel regression,[18] outperform nonlinear machine learning methods such as SVR and neural networks. Excellent reviews concerning color constancy can be found in Refs.[29–32]

The Gaussian process and RVM are probabilistic instances of extended linear models. Many popular machine learning methods, such as ridge regression, kernel regression, neural networks and SVR, can all be considered as special cases of GPR with certain covariance functions to some extent as implied in Refs.[19,33] and some of the above methods have gained excellent results for color constancy. The most compelling feature of RVM[20] is its equivalent generalization performance as compared with support vector machines (SVM).[34,35] However, the number of relevance vectors is, in most cases, dramatically smaller than that of support vectors used by SVM to solve the same problem.

ALGORITHMS

In this section, two Bayesian kernel methods, namely the Gaussian process and RVM for regression, are discussed.

Given a data set D consisting of N input vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}^{\mathrm{T}}$ and corresponding noisy outputs $\mathbf{t} = \{t_1, t_2, \ldots, t_N\}^{\mathrm{T}}$, the regression task is to estimate an underlying function $y(\mathbf{x})$ to satisfy

$$t_i = y(\mathbf{x_i}) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \tag{2}$$

i.e. from this training set, to learn a target model depending on the inputs for making accurate predictions of t for previously unseen values of $\mathbf{x}$.

For a linear regression problem, this is generally based on finding a parameter vector w and an offset c such that we can predict y for an unknown input $\mathbf{x} \in \mathbb{R}^D$:

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + c \tag{3}$$

where $\mathbf{w} = (w_1, w_2, \ldots, w_M)^{\mathrm{T}}$, and M is the dimension of $\mathbf{x}$. In practice the offset c is usually incorporated into w.

If there is a nonlinear relationship between $\mathbf{x}$ and y, a basis function $\phi(\mathbf{x})$ can be used to implement nonlinear mapping, so Eq. (3) can be expressed as

$$y(\mathbf{x}, \mathbf{w}) = \sum_{i=0}^{M} w_i\phi_i(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}), \tag{4}$$

where $\phi(\mathbf{x}) = (\phi_0, \phi_1, \ldots, \phi_M)^{\mathrm{T}}$ and $\phi_0 = 1$. For notational clarity, the bias item will not be considered explicitly in the following sections.

A classical treatment of non-Bayesian regression such as SVR seeks a point estimate of the unknown parameter vector $\mathbf{w}$. By contrast, in a Bayesian approach, the uncertainty of $\mathbf{w}$ is characterized through a probability distribution $p(\mathbf{w})$. The prediction is made by integrating with respect to the posterior distribution of $\mathbf{w}$ given the data set D.

Gaussian Process Regression

The Gaussian process assumes that any functional values $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}^{\mathrm{T}}$ of $y(\mathbf{x})$ are multivariate Gaussian distributed with zero mean such as

$$p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K}) \tag{5}$$

where $\mathbf{K}$ is the covariance (or Gram) matrix, which can be expressed as $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = E[y_iy_j]$ and $K(\mathbf{x}_i, \mathbf{x}_j)$ is the covariance function. Evidently the likelihood of the noise model is

$$p(\mathbf{t}|\mathbf{y}, \mathbf{X}) = \mathcal{N}(\mathbf{y}, \sigma^2\mathbf{I}). \tag{6}$$

Integrating over the function variables, we can get

$$p(\mathbf{t}|\mathbf{X}) = \int p(\mathbf{t}|\mathbf{y}, \mathbf{X})p(\mathbf{y}|\mathbf{X})d\mathbf{y} = \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I}). \tag{7}$$

If we have N training input sets and T prediction input sets, evidently

$$p(\mathbf{y}, \mathbf{y}_T|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{N+T}), \tag{8}$$

where $\mathbf{K}_{N+T} = \begin{bmatrix} \mathbf{K}_N & \mathbf{K}_{NT} \\ \mathbf{K}_{TN} & \mathbf{K}_T \end{bmatrix}$.

If we include noise for t, we can get that

$$\begin{aligned} p(\mathbf{t}, \mathbf{t}_T|\mathbf{X}) &= \mathcal{N}(\mathbf{0}, \mathbf{K}_{N+T} + \sigma^2\mathbf{I}) \\ p(\mathbf{t}_T|\mathbf{t}, \mathbf{X}) &= \mathcal{N}(\mu_T, \Sigma_T), \end{aligned} \tag{9}$$

where

$$\mu_T = \mathbf{K}_{TN}[\mathbf{K}_N + \sigma^2\mathbf{I}]^{-1}\mathbf{t} \tag{10a}$$

$$\Sigma_T = \mathbf{K}_T - \mathbf{K}_{TN}[\mathbf{K}_N + \sigma^2\mathbf{I}]^{-1}\mathbf{K}_{NT} + \sigma^2\mathbf{I}. \tag{10b}$$

It can be shown that the GPR model not only provides fully probabilistic predictive distributions, but also includes estimates of the uncertainty of the predictions, which is in contrast to the case for many other commonly used regression techniques, only providing the best estimation. Furthermore, the Gaussian process predictor is based on priors over functions, rather than on priors over parameters; i.e. GPR is a nonparametric method, and therefore, it can be rigorously used to let the data speak more clearly for themselves without cross-validation.

One uses the training data to optimize the parameters to avoid over-fitting in most machine learning methods. However, covariance functions in GPR tend to have a small number of hyperparameters; therefore, over-fitting does not tend to be a problem. Secondly, the hyperparameter optimization takes place at a higher hierarchical level. It does not directly optimize the function variables themselves, but rather integrates over their uncertainty as in (7). A commonly used method is to minimize the negative log marginal likelihood $\mathcal{L}(\theta)$ with respect to the hyperparameters of the covariance θ:

$$\begin{aligned} \mathcal{L} &= -\log p(\mathrm{t}|\theta) \\ &= \frac{1}{2}\mathrm{logdet}\,\mathbf{C}(\theta) + \frac{1}{2}t^T\mathbf{C}^{-1}t + \frac{N}{2}\log(2\pi) \end{aligned} \tag{11}$$

where $\mathbf{C} = \mathbf{K}_N + \sigma^2\mathbf{I}$.

Equation (11) is a non-convex optimization task, which can be obtained using a gradient based method such as the conjugate gradient or quasi-Newton method. There may exist local minima for GPR, particularly when there is a small amount of data; therefore, it is often worth making several optimizations from random starting points and investigating the different minima.

The Relevance Vector Machine for Regression

Assuming that noise follows an independent, identical Gaussian distribution, the likelihood of $\mathbf{w}$ is

$$p(\mathbf{t}|\mathbf{w},\sigma^2)=\prod_{i=1}^{N}\mathcal{N}(t_i|y(\mathbf{x}_i;\mathbf{w}),\sigma^2)$$
$$=(2\pi\sigma^2)^{-\frac{N}{2}}\exp\left(-\frac{\|\mathbf{t}-\Phi\mathbf{w}\|^2}{2\sigma^2}\right),\quad(12)$$

where Φ is the $N\times(M+1)$ design matrix with $\Phi_{nm}=\phi_m(\mathbf{x}_n)$, $\Phi_{n0}=1$. With as many parameters in the model as training examples, the maximum likelihood estimation for w and σ^2 from (12) leads to severe over-fitting. To avoid this, the parameters are constrained by defining an explicit prior probability distribution over $\mathbf{w}$:

$$p(w_i|\alpha_i)=\mathcal{N}(w_i|0,\alpha_i^{-1}),\quad(13)$$

where $\alpha=[\alpha_0,\alpha_1,\alpha_2,\ldots,\alpha_M]^{\mathrm{T}}$, i.e. an individual hyperparameter α_i is associated with each weight w_i. This implies

$$p(\mathbf{w}|\alpha)=\prod_{i=0}^{M}\mathcal{N}(0,\alpha_i^{-1})=\prod_{i=0}^{M}\left(\frac{\alpha_i}{2\pi}\right)^{\frac{1}{2}}\exp\left(-\frac{\alpha_i w_i^2}{2}\right).\quad(14)$$

Given a new test point $\mathbf{x}_*$, predictions are made for the corresponding target t_*, in terms of the predictive distribution:

$$p(t_*|\mathbf{t})=\int p(t_*|\mathbf{w},\alpha,\sigma^2)p(\mathbf{w},\alpha,\sigma^2|\mathbf{t})d\mathbf{w}d\alpha d\sigma^2.\quad(15)$$

The first item on the right hand side is

$$p(t_*|\mathbf{w},\alpha,\sigma^2)=p(t_*|\mathbf{w},\sigma^2)=\mathcal{N}(t_*|y(\mathbf{x}_*;\mathbf{w}),\sigma^2)\quad(16)$$

and the second item on the right hand side is

$$p(\mathbf{w},\alpha,\sigma^2|\mathbf{t})=p(\mathbf{w}|\mathbf{t},\alpha,\sigma^2)p(\alpha,\sigma^2|\mathbf{t}).\quad(17)$$

In (17), the posterior distribution over weights is given by

$$p(\mathbf{w}|\mathbf{t},\alpha,\sigma^2)=\frac{p(\mathbf{t}|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha)}{p(\mathbf{t}|\alpha,\sigma^2)}.\quad(18)$$

Using (16) and (14), the denominator is given by

$$p(\mathbf{t}|\alpha,\sigma^2)=\int p(\mathbf{t}|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha)d\mathbf{w}$$
$$=(2\pi)^{-\frac{N}{2}}|\Omega|^{-\frac{1}{2}}\exp\left(-\frac{\mathbf{t}^{\mathrm{T}}\Omega^{-1}\mathbf{t}}{2}\right).\quad(19)$$

So Eq. (18) can be re-expressed as

$$p(\mathbf{w}|\mathbf{t},\alpha,\sigma^2)=(2\pi)^{-\frac{M+1}{2}}|\Sigma|^{-\frac{1}{2}}$$
$$\times\exp\left(\frac{(\mathbf{w}-\mu)^{\mathrm{T}}\Sigma^{-1}(\mathbf{w}-\mu)}{-2}\right)\quad(20)$$

where

$$\Sigma=(\sigma^{-2}\Phi^{\mathrm{T}}\Phi+\Lambda)^{-1}\quad(21a)$$
$$\mu=\sigma^{-2}\Sigma\Phi^{\mathrm{T}}\mathbf{t}\quad(21b)$$
$$\Omega=\sigma^{-2}\mathrm{I}+\Phi\Lambda^{-1}\Phi^{\mathrm{T}}\quad(21c)$$

and $\Lambda=\mathrm{diag}(\alpha_0,\alpha_1,\ldots,\alpha_M)$.

Having obtained the first item on the right hand side of (17), the second term $p(\alpha,\sigma^2|\mathbf{t})$ can be represented by a delta function at its mode, and then the prediction can be re-expressed as

$$p(t_*|\mathbf{t})=\int p(t_*|\mathbf{w},\alpha,\sigma^2)p(\mathbf{w}|\mathbf{t},\alpha,\sigma^2)p(\alpha,\sigma^2|\mathbf{t})d\mathbf{w}d\alpha d\sigma^2$$
$$\approx\int p(t_*|\mathbf{w},\alpha,\sigma^2)p(\mathbf{w}|\mathbf{t},\alpha,\sigma^2)\delta(\alpha_{\mathrm{MP}},\sigma^2_{\mathrm{MP}})d\mathbf{w}d\alpha d\sigma^2$$
$$=\int p(t_*|\mathbf{w},\alpha_{\mathrm{MP}},\sigma^2_{\mathrm{MP}})p(\mathbf{w}|\mathbf{t},\alpha_{\mathrm{MP}},\sigma^2_{\mathrm{MP}})d\mathbf{w}\quad(22)$$

where $(\alpha_{\mathrm{MP}},\sigma^2_{\mathrm{MP}})=\arg\max_{\alpha,\sigma^2}p(\alpha,\sigma^2|\mathbf{t})$. Because Eqs. (20) and (16) are all Gaussian, Eq. (22) can be re-expressed as

$$p(t_*|\mathbf{t})=\mathcal{N}(t_*|y_*,\sigma_*^2),\quad(23)$$

where

$$y_*=\mu^{\mathrm{T}}\phi(\mathbf{x}_*)\quad(24a)$$
$$\sigma_*^2=\sigma^2_{\mathrm{MP}}+\phi(\mathbf{x}_*)^{\mathrm{T}}\Sigma\phi(\mathbf{x}_*).\quad(24b)$$

It can be shown that the RVM model not only provides fully probabilistic predictive distributions, but also includes estimates of the uncertainty of the predictions, like GPR.

To obtain $(\alpha_{\mathrm{MP}},\sigma^2_{\mathrm{MP}})$, we can use

$$p(\alpha,\sigma^2|\mathbf{t})\propto p(\mathbf{t}|\alpha,\sigma^2)p(\alpha)p(\sigma^2)\quad(25)$$

and ignore $p(\alpha)$ and $p(\sigma^2)$ for the case of uniform hyperpriors, so only maximizing $p(\mathbf{t}|\alpha,\sigma^2)$ i.e. Eq. (19). This can be obtained by taking derivatives of (19) and setting them to zero:

$$\alpha_i^{\mathrm{new}}=\frac{\gamma_i}{\mu_i^2}$$
$$(\sigma^2)^{\mathrm{new}}=\frac{\|\mathbf{t}-\Phi\mu\|^2}{N-\Sigma_i\gamma_i},\quad(26)$$

where μ_i is the ith posterior mean weight from (21b) and $\gamma_i=1-\alpha_i\Sigma_{ii}$.

The learning algorithm thus proceeds by repeated application of (26), concurrently updating the posterior statistics μ and Σ from (21b) and (21a), until some suitable convergence criteria have been satisfied. In practice, it can be found that most of them go to infinity when maximizing the evidence with respect to these hyperparameters, and the corresponding weight parameters have posterior distributions that are concentrated at zero. The basis functions associated with these parameters play no role in the predictions and are effectively pruned out, which results in a sparse model.[22]

DATA REPRESENTATION

The regression task is to set up the relationship between input vectors and their corresponding outputs. Thereby, the first step is to construct input $\mathbf{X}$ and corresponding noisy outputs t. For color constancy, the image intensity is often ignored, so we use the chromaticity space (r, g) as expressed in (27), which is commonly adopted in previous articles[12,14,16,36] on color constancy. The advantage of using the chromaticity space (r, g) is that it reduces the dimensionality of the data set from 3D RGB space to 2D (r, g) space:

$$\begin{aligned} r &= R/(R+G+B) \\ g &= G/(R+G+B). \end{aligned} \tag{27}$$

Before the image is converted to the chromaticity space (r, g), those pixels whose values are under a threshold value of 7 (on a 0–255 scale) in any of the three RGB color channels are removed. However, this will eliminate far more pixels than just the dark pixels. For instance, saturated colors can have high values for one or two of the color channels, but low values for the third channel. In most cases, saturated colors are also removed. After removing the dark pixels, the image is averaged using a 5 × 5 local filter to reduce the noise. Finally, the color image is converted into the chromaticity space (r, g) and sampled into $N \times N$ bins. Because the color constancy method commonly neglects the intensity of the image, $N \times N$ bins are binarized to obtain a 2D binary chromaticity histogram of 1s and 0s. 1s represent the presence of illumination chromaticity in the bins and 0s represent the absence of illumination chromaticity in the bins. The 2D $N \times N$ binary histogram is represented as a vector of dimensions $1 \times N^2$: an input vector $\mathbf{x}_i$, and all the vectors of the real image set make up the input vectors $\mathbf{X}$. The corresponding output comes from light ground-truth information. In terms of illumination chromaticity estimation, the setting of $N = 32$ as the bin width is empirically selected, and provides the best results. Other larger bin width selections (48 and 64) do not improve the results much, but slow the process drastically, while smaller bin width selections (8 and 16) perform poorly.[16]

THE REAL IMAGE SET AND ERROR MEASURES

Two Real Image Sets

To validate the performances of both Bayesian kernel methods, two real image sets are selected. The first image set was extracted by Bianco et al.,[28] a representative subset of 1135 images that are much less correlated from the Ciurea and Funt[37] image set. The latter was extracted from 2 h of video clips recorded in many places: indoor, outdoor, desert, cityscape, etc.; the ground-truth was acquired using a gray ball mounted before the camera. Another image set was obtained from Shi.[27] The image quality of this image set is high and the ground-truth is accurate, using the Macbeth color-checker, which must be excluded during validating the algorithm. The disadvantage of the image set is that it only includes 568 images.

Table I. Optimized values of SVR free parameters on two image sets.

	Bianco[28] image set		Shi[27] image set	
	r-SVR	g-SVR	r-SVR	g-SVR
Cost	0.0625	0.0156	0.1250	0.1250
γ	0.0156	0.0156	0.0156	0.0156
ϵ	0.0078	0.0010	0.0008	0.0039

Table II. Optimized kernel width on two image sets.

	Gauss	Laplace	Cauchy
Shi[27] image set	5.0	5.0	4.0
Bianco[28] image set	9.0	11.0	8.0

Error Measures

The commonly accepted measure for evaluating the performance of color constancy algorithms is angular error, which correlates reasonably well with the perceived quality of the output image.[29] Given pixel illuminant estimation $\mathbf{e}_u$ and the actual light source (ground-truth) $\mathbf{e}_c$, the angular error is

$$\varepsilon(\mathbf{e}_u, \mathbf{e}_c) = \cos^{-1}\left(\frac{\mathbf{e}_u \cdot \mathbf{e}_c}{\|\mathbf{e}_u\|\,\|\mathbf{e}_c\|}\right). \tag{28}$$

The less the angular error, the better the performance of the algorithm.

RESULTS

Optimized Parameters for SVR, RVM and RR

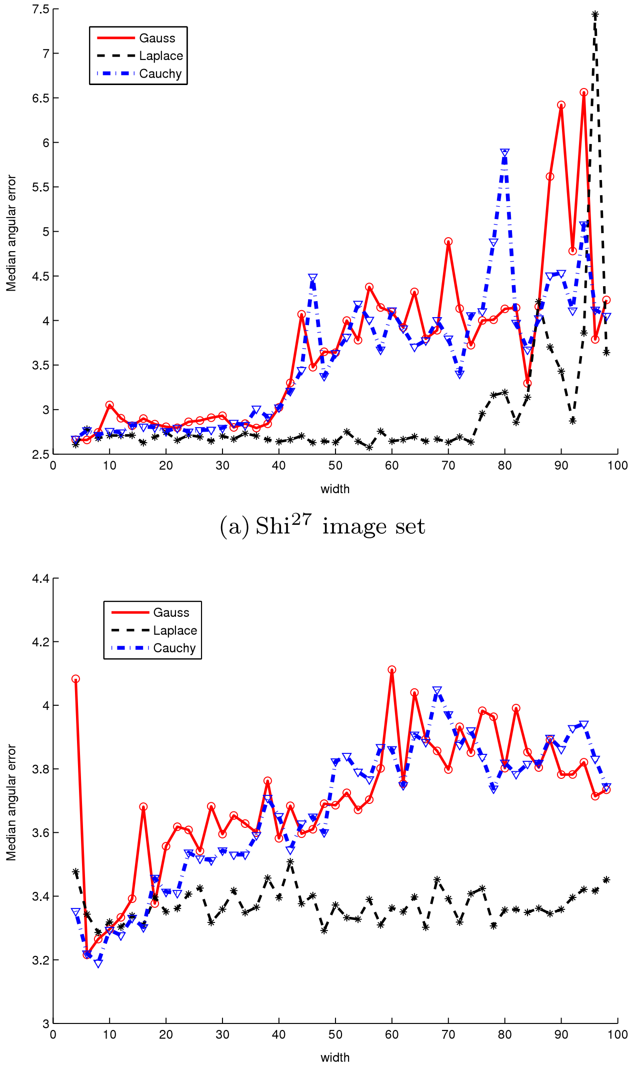
To compare three kernel based methods, SVR for color constancy is also presented in this article. The optimized parameters of SVR are listed in Table I using the radial basis function as the kernel function, an ϵ-insensitive error function, and K-fold cross-validation.

The kernel width for RVM directly influences the result. The width is set from 1 to 100 and the median angular error of each width on two image sets is calculated. The relationships between the kernel width and median angular errors on two image sets are shown in Figure 2. From Fig. 2, we can see that the Laplace kernel width has less influence on the result than the Gaussian and Cauchy kernels. The optimized width parameters for each RVM kernel are listed in Table II.

To compare linear regression approaches and nonlinear regression approaches, ridge regression is conducted on two image sets. The optimized parameter of ridge regression in channel (r, g) on Bianco's image set[28] is (1764, 1688) and it is (622, 304) on the Shi[27] image set.

Covariance Function Selection for GPR

The covariance function is the crucial ingredient in GPR. Compared with other methods, GPR has the advantage of



(a) Shi[27] image set

(b) Bianco[28] image set

Figure 2. Relationships between the kernel width and median angular error for RVM on two image sets.

directly selecting covariance hyperparameters from the training data rather than using a scheme such as cross-validation. To find the most suitable covariance function for color constancy, four stationary covariance functions of $\mathbf{x}-\mathbf{x}'$, three dot product covariance functions which only depend on $\mathbf{x}\cdot\mathbf{x}'$ and combinations of these covariance functions are selected.

- The diagonal squared exponential covariance function (SEiso):

$$K(\mathbf{x},\mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2l^2}(\mathbf{x}-\mathbf{x}')^{\mathrm{T}}(\mathbf{x}-\mathbf{x}')\right). \tag{29}$$

SEiso is probably the most widely used kernel within the kernel machines field.

- The rational quadratic covariance function (RQiso):

$$K(\mathbf{x},\mathbf{x}') = \sigma^2 \left(1+\frac{1}{2\alpha l^2}(\mathbf{x}-\mathbf{x}')^{\mathrm{T}}(\mathbf{x}-\mathbf{x}')\right)^{-\alpha} \tag{30}$$

with $\alpha, l > 0$ can be seen as a scaled mixture of squared exponential covariance functions with different length scales.

- The piecewise polynomial covariance with compact support:

$$K(\mathbf{x},\mathbf{x}') = \sigma^2 \max(0, 1-\mathrm{r})\cdot f_\nu, \tag{31}$$

where f_ν is a piecewise polynomial and $\mathbf{r} = \frac{\|\mathbf{x}-\mathbf{x}'\|}{l}$. Because the covariances between points become exactly zero when their distance exceeds a certain threshold, the covariance matrix will become sparse, which leads to the possibility of computational advantages. Compact support polynomials of degree 2 (PPiso2) and degree 3 (PPiso3) are selected.

- The Matérn class covariance function:

$$K(\mathbf{x},\mathbf{x}') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu r}}{\lambda}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu r}}{\lambda}\right), \tag{32}$$

where $r = \|\mathbf{x}-\mathbf{x}'\|$ and K_ν is a modified Bessel function. We select $\nu = 1/2$, which corresponds to a Laplace function (Matern1), and $\nu = 3/2$, as a covariance function (Matern3).

- The polynomial covariance function:

$$K(\mathbf{x},\mathbf{x}') = (\mathbf{x}^{\mathrm{T}}\mathbf{x}' + c)^d. \tag{33}$$

A linear polynomial (Poly1), which is equivalent to that of Bayesian linear regression, a quadratic polynomial (Poly2), and a cubic polynomial (Poly3) are selected.

- The squared exponential covariance function (SEisoU):

$$K(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{1}{2l^2}\mathbf{x}^{\mathrm{T}}\mathbf{x}'\right). \tag{34}$$

- The neural network covariance function (NNone):

$$K(\mathbf{x},\mathbf{x}') = \sigma^2 \sin^{-1}\left(\frac{\mathbf{x}^{\mathrm{T}}\Lambda^{-2}\mathbf{x}'}{\sqrt{f(\mathbf{x})f(\mathbf{x}')}}\right), \tag{35}$$

where $f(\mathbf{x}) = 1 + \mathbf{x}^{\mathrm{T}}\Lambda^{-2}\mathbf{x}$ and Λ is the diagonal matrix.

Kernel Function Selection for RVM

Three kernel types are used in this article for RVM. The Gaussian kernel (RvmG) and the Laplace kernel (RvmL) are the same as SEiso and Matern1 of GPR, and the Cauchy kernel (RvmC) is

$$K(\mathbf{x},\mathbf{x}') = \frac{1}{1+\eta(\mathbf{x}-\mathbf{x}')^{\mathrm{T}}(\mathbf{x}-\mathbf{x}')}. \tag{36}$$

Experimental Results

To evaluate the performances of different machine learning methods, all algorithms are trained using the same setting, based on K-fold cross-validation. Here K is set to 15, i.e. training is performed by dividing the image set into 15

Table III. Angular errors of GPR, using different covariance functions and several other methods on the Shi[27] image set.

	Median	Mean	Best-25%	Worst-25%
Matern1	**2.3841**	3.1927	0.8695	6.7977
Matern3	**2.4524**	3.2188	0.9029	6.8424
PPiso2	**2.4794**	3.2285	0.9091	6.8542
PPiso3	**2.4789**	3.2353	0.9124	6.8618
SEiso	2.5163	3.2598	0.9238	6.8890
RQiso	**2.4379**	3.2055	0.8861	6.8725
NNone	2.5165	3.2529	0.9086	6.8933
SEisoU	2.7781	3.4525	1.0028	7.1565
Poly1	2.7827	3.4529	1.0014	7.1607
Poly2	2.6353	3.3488	0.9354	7.0521
Poly3	2.6445	3.3320	0.9344	7.0094
RvmG	2.5967	3.4527	0.9809	7.4165
RvmL	**2.5764**	3.4103	1.0043	7.1658
RvmC	2.6716	3.4103	0.9787	7.1934
SVR	2.4882	3.2300	0.9091	6.8048
RR	2.6911	3.4397	1.0253	7.0765
GW	6.2632	6.3504	2.3351	10.5997
WP	5.7963	7.6060	1.4872	16.2017
GGW	3.5974	5.2970	1.0049	12.2437
GE1	4.5764	5.3276	1.8641	10.0140
GE2	4.5829	5.3273	1.9409	9.8195

Table IV. Angular errors of GPR, using different covariance functions and several other methods on the Bianco[28] image set.

	Median	Mean	Best-25%	Worst-25%
Matern1	3.0842	4.0195	0.8610	8.7668
Matern3	**3.0309**	3.9945	0.8622	8.7660
PPiso2	**3.0295**	3.9957	0.8731	8.7755
PPiso3	3.0452	3.9977	0.8781	8.7827
SEiso	3.0408	4.0124	0.8903	8.8111
RQiso	**3.0324**	4.0014	0.8807	8.7948
NNone	3.2895	4.1307	0.9170	8.9049
SEisoU	3.5952	4.3881	1.0586	9.2141
Poly1	3.5919	4.3874	1.0551	9.2192
Poly2	3.2978	4.2263	0.8948	9.1399
Poly3	3.2774	4.1896	0.8618	9.1281
RvmG	**3.1859**	4.1499	0.9392	8.8410
RvmL	3.2498	4.2084	1.0038	8.9294
RvmC	3.1906	4.1681	1.0086	8.8604
SVR	3.0350	3.9653	0.8095	8.6863
RR	3.6596	4.4383	1.0334	9.2306
GW	6.1732	7.2764	1.9699	14.4307
WP	6.3479	7.8873	1.0583	17.6045
GGW	6.2485	7.1625	1.8322	14.2608
GE1	5.2715	6.2498	1.6448	12.4847
GE2	5.4248	6.3011	1.6912	12.5822

parts, and the method is trained on 14 parts of the data and tested on the remaining part. This procedure is repeated 15 times, so every image is in the test set exactly once and all images from the same scene will either be in the training set or in the test set at the same time.

Four criteria, namely the median and mean angular error, the mean of the best 25% of the image set with the smallest angular error, and the mean of the worst 25% of the image set with the largest angular error, are evaluated on two image sets. Tables III and IV show the performances of GPR under different covariance functions and RVM under different kernel functions on two selected real image sets. To compare the differences, the results from SVR, RR and the traditional methods GW, WP, GGW, GE1 (first-order GE), and GE2 (second-order GE) are also listed. All the results from the machine learning based method are obtained using optimized parameters in fifth subsection. The commonly accepted parameters for the traditional methods GW, WP, GGW, GE1, and GE2 used in this article were obtained from Van De Weijer.[3] The additive combinations of the different covariance functions perform less well than each separately, so the results from GPR obtained using combinations of covariance functions are not listed.

Besides using the K-fold cross-validation method to evaluate the performances of algorithms, we also conduct experiments on the Bianco[28] image set using the Shi[27] image set as the training set and vice versa. The root mean square errors (RMSE) of the algorithms are listed in Table V.

Table V. RMSE of the algorithms on two image sets using different cross-validation methods.

	Bianco[28] image set		Shi[27] image set	
	K-fold	Shi[27] (training set)	K-fold	Bianco[28] (training set)
Matern1	0.0316	0.0695	0.0257	0.0748
SEiso	0.0317	0.0760	0.0258	0.0749
RvmG	0.0322	0.0926	0.0277	0.0741
RvmL	0.0068	0.0829	0.0269	0.0743
SVR	0.0313	0.0665	0.0256	0.0752
RR	0.0339	0.0716	0.0265	0.0790

Analysis

From Tables III and IV, we can see that there exist small differences of median angular error for some methods, so the difference may not be statistically significant. The Wilcoxon signed-rank test is used to evaluate whether a difference is statistically significant. Here, the error rate for accepting or rejecting the null hypothesis is always set to 0.05. Table VI summarizes the Wilcoxon test, among several algorithms.

On the basis of the results shown in Tables III, IV and VI, we can see that, for GPR using different covariance functions:

- All the dot product covariance functions perform less well than stationary covariance functions for color constancy problem.
- The Matérn class covariance function achieves better performance on both image sets than most other covariance functions.
- The Poly1 covariance function performs less well than any other covariance function.

For the first item, we can explain that under the Gaussian process view, points with inputs x which are close are likely to have similar target values y, and thus training points that are near to a test point should be informative as regards the prediction at that point. Therefore, the stationary covariance function of $x - x'$ has more similarity for prediction than the dot product covariance function of $x \cdot x'$ for the color constancy problem, which mostly results from the light estimation depending much more on the difference of adjacent pixels than their product. For the second item, as noted by Stein,[38] the SEiso covariance function may be strongly smooth and unrealistic for modeling many physical processes; in addition, RQiso with $\alpha, l > 0$ can be seen as a scaled mixture of squared exponential covariance functions with different length scales; thereby, the Matérn class is an alternative.

The performances of RVM using different kernel functions are almost the same as regards the standard deviations (listed in Table VII) for each kernel method. The weak difference can be described as follows:

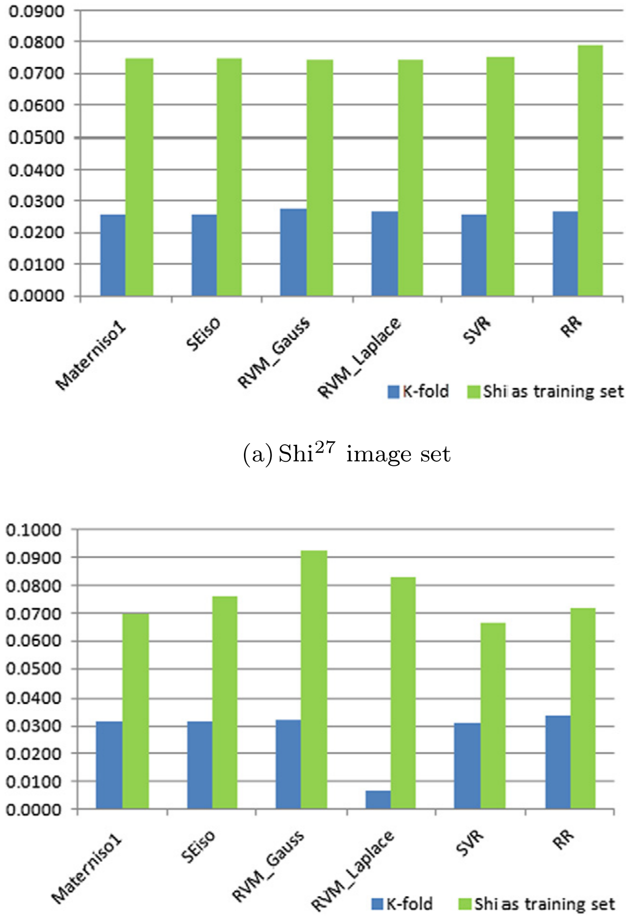
- On Shi's image set, the performance of the RVM method is better than those of other kernels regardless of whether one considers the median or worst-25% angular error when the Laplace kernel is used.
- On Bianco's image set, the performance of the RVM method is better than those of other kernels regardless of whether one considers the median or worst-25% angular error when the Gaussian kernel is used.

Compared with SVR, RR, and traditional methods:

- GPR outperforms RVM for regression on two real image sets and can achieve almost the same results as SVR when the stationary covariance function is used.
- The performance of GPR is almost the same as that of RVM for regression when the dot product covariance function is used.
- The performances of the three kernel based methods are better than that of ridge regression.
- All the machine learning based methods perform better than any other traditional methods, which is self-evident.

The performance histogram of the different algorithms based on the results shown in Table V is shown in Figure 3. We can see that:

- The RMSE when using the Laplace kernel for GPR and RVM is smaller than that when using the Gaussian kernel on two image sets.



(a) Shi[27] image set

(b) Bianco[28] image set

Figure 3. RMSE for the algorithm on two image sets.

- On Bianco's image set, the RMSE when using the Laplace kernel for RVM is the smallest among those for the four machine learning methods.
- On Shi's image set, the RMSE of GPR is smaller than that of RVM.
- The performances of all algorithms using K-fold cross-validation are much better than those using other image sets as the training set.

So we can conclude that, for images without light ground-truth, the best performance can be achieved by using traditional methods to get their initial light estimations, incorporating them into the training image set to get optimized k-fold cross-validation parameters, and then using a regression based method with optimized parameters to get the best light estimations of them.

The Reliability of the Estimation Process

As noted above, both GPR and RVM models not only provide fully probabilistic predictive distributions, but also include estimates of the uncertainties of the predictions. These distributions provide a useful way to quantify the uncertainty in model estimates, and to exploit our knowledge of this uncertainty in order to make more robust predictions

Table VI. Comparison of the different algorithms via the Wilcoxon signed-rank test on two image sets. A '+' means that the algorithm listed in the corresponding row is better than the one in the corresponding column; a '−' indicates the opposite; an '=' indicates that the performances of the respective algorithms are statistically equivalent.

(a) On the Shi[27] image set

	Matern1	Matern3	PPiso2	PPiso3	SEiso	RQiso	NNone	SEisoU	Poly1	Poly2	Poly3	RvmG	RvmL	RvmC	SVR	RR
Matern1		+	+	+	+	=	+	+	+	+	+	+	+	+	=	+
Matern3	−		+	+	+	=	=	+	+	+	+	+	+	+	=	+
PPiso2	−	−		−	+	=	=	+	+	+	+	+	+	+	=	+
PPiso3	−	−	+		+	=	=	+	+	+	+	+	+	+	=	+
SEiso	−	−	−	−		−	=	+	+	+	+	+	+	+	=	+
RQiso	=	=	=	=	+		=	+	+	+	+	+	+	+	=	+
NNone	−	=	=	=	=	=		+	+	+	+	+	+	+	=	+
SEisoU	−	−	−	−	−	−	−		=	−	−	−	−	−	−	=
Poly1	−	−	−	−	−	−	−	=		−	−	−	−	−	−	=
Poly2	−	−	−	−	−	−	−	+	+		+	=	=	=	−	=
Poly3	−	−	−	−	−	−	−	+	+	−		=	=	=	−	+
RvmG	−	−	−	−	−	−	−	+	+	=	=		=	=	−	=
RvmL	−	−	−	−	−	−	−	+	+	=	=	=		=	−	=
RvmC	−	−	−	−	−	−	−	+	+	=	=	=	=		−	=
SVR	=	=	=	=	=	=	=	+	+	+	+	+	+	+		+
RR	−	−	−	−	−	−	−	=	=	=	−	=	=	=	−	
(b) On the Bianco[28] image set																
Matern1		−	=	=	=	=	+	+	+	+	+	+	+	+	−	+
Matern3	+		=	=	=	=	+	+	+	+	+	+	+	+	=	+
PPiso2	=	=		=	=	=	+	+	+	+	+	+	+	+	=	+
PPiso3	=	=	=		−	=	+	+	+	+	+	+	+	+	=	+
SEiso	=	=	=	+		−	+	+	+	+	+	+	+	+	=	+
RQiso	=	=	=	=	+		+	+	+	+	+	+	+	+	=	+
NNone	−	−	−	−	−	−		+	+	+	=	=	−	+	−	+
SEisoU	−	−	−	−	−	−	−		=	−	−	−	−	=	−	=
Poly1	−	−	−	−	−	−	−	=		−	−	−	−	=	−	=
Poly2	−	−	−	−	−	−	−	+	+		−	=	=	+	−	+
Poly3	−	−	−	−	−	−	=	+	+	+		=	=	+	−	+
RvmG	−	−	−	−	−	−	=	+	+	=	=		=	+	−	+
RvmL	−	−	−	−	−	−	+	+	+	=	=	=		+	−	+
RvmC	−	−	−	−	−	−	−	=	=	−	−	−	−		−	=
SVR	+	=	=	=	=	=	+	+	+	+	+	+	+	+		+
RR	−	−	−	−	−	−	−	=	=	−	−	−	−	=	−	

on new test points. On the basis of the prediction variance of the estimations in (10*b*) and (24*b*) separately, we can predict the confidence interval of illumination chromaticity estimation $\mathbf{e}_u$ is $\mathbf{e}_u \pm 1.96\sqrt{(\mathrm{var}(\mathbf{e}_u))}$ for a confidence level of 95%.

With the noise level given as 0.2, the standard deviations of the algorithm are as listed in Table VII. It can be seen that the mean of all the predictive standard deviations of the image set using different GPR covariance functions is around 2.2% on Shi's image set and 2.7% on Bianco's image set, with little difference. The mean of all the predictive standard deviations of the image set using different RVM kernels is around 1.5% on Shi's image set and 2.2% on Bianco's image set, with little difference. Moreover, the standard deviation of RVM is less than that of GPR.

IMAGE CORRECTION

On obtaining the chromaticity information from all algorithms, diagonal transformation[39] is used to correct the color image. Figure 4 shows some example results from various methods applied to the Bianco[28] image set.

In general, it is necessary to use the training image set to obtain the best model for light prediction for the unseen image (e.g. web images). We can choose the best kernel covariance function with the least median angular error based on the analysis in fifth subsection and the least variance

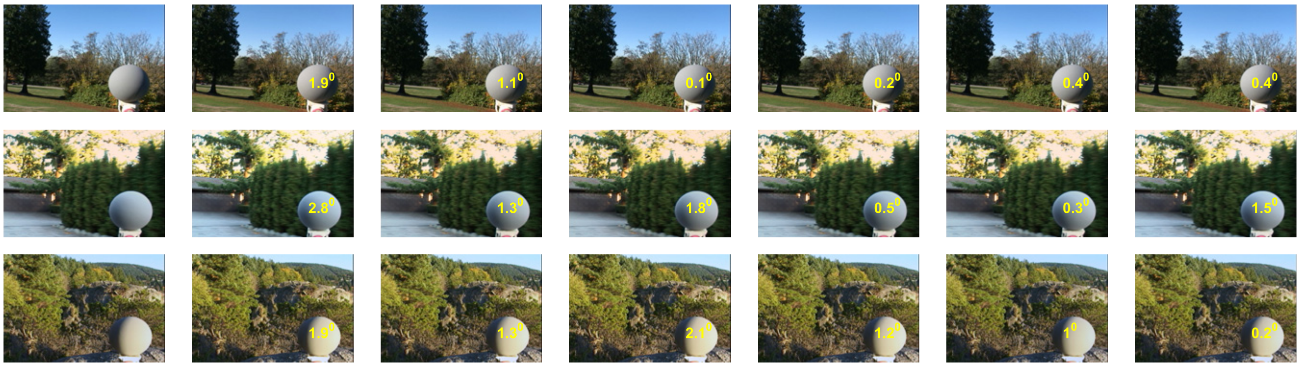


Figure 4. Some sample results from various methods applied to the Bianco[28] image set. The angular error is shown in the bottom right corner of the image. The methods used are, from left to right: perfect color constancy using ground-truth, ridge regression, support vector regression using the radial basis function, GPR using SEiso and Matern1 covariance functions, and RVM for regression using Gaussian and Laplace kernels.

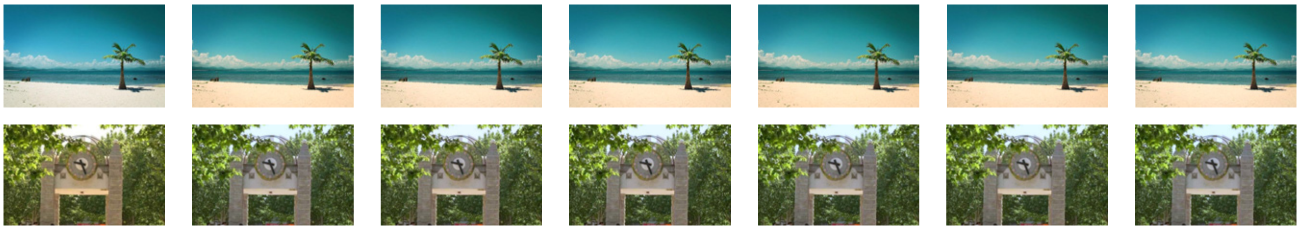


Figure 5. Some sample results from various methods applied to web images. The leftmost image is the source image. The methods used are, from left to right: ridge regression, support vector regression using the radial basis function, GPR using SEiso and Matern1 covariance functions, and RVM for regression using Gaussian and Laplace kernels.

Table VII. Standard deviations of the algorithm on two image sets.

	Shi[27] image set	Bianco[28] image set
Matern1	0.0220	0.0270
Matern3	0.0222	0.0265
PPiso2	0.0223	0.0264
PPiso3	0.0223	0.0264
SEiso	0.0225	0.0263
RQiso	0.0222	0.0263
NNone	0.0219	0.0277
SEisoU	0.0227	0.0287
Poly1	0.0227	0.0287
Poly2	0.0231	0.0274
Poly3	0.0237	0.0275
RvmG	0.0165	0.0228
RvmL	0.0140	0.0222
RvmC	0.0150	0.0222

analyzed in sixth subsection. Some sample results from the web are shown in Figure 5. Here, the Bianco[28] image set is selected as the training image set because of it having less correlation among images.

THE OUTLIER INFLUENCE FOR GPR AND RVM

The data set with Gaussian noise leads to poor results if the data are prone to outliers due to the light tails of the noise distribution when GPR and RVM are used. We assume that the images with large angular error can be thought of as outliers among test images. The mean angular error of the worst 1% and 5% of all the image sets with the largest angular error are used to evaluate the outlier influence. Heavy-tailed Laplace and Student-t distributions (with $\nu = 1$ selected, also named as the Cauchy distribution) are used as the likelihood function for GPR or the kernel function for RVM against outliers. Because exact inference is only tractable for Gaussian likelihood, the variational Bayesian inference method is used for the three likelihood functions to maintain consistency among them.

Tables VIII and IX show angular errors, using three different likelihood functions with the Matern1 covariance function for GPR via *leave-one-out* cross-validation on Shi's image set and 15-fold cross-validation on Bianco's image set separately.

From Tables VIII and IX, we can see that the median, mean, and average angular errors of the worst 5% and 1% increase when Laplace and Student-t distributions are used as likelihood functions compared with a Gaussian distribution, which implies that heavy-tailed distributions cannot enhance the performance of GPR for color constancy. Among the three likelihood functions, the Gaussian distribution performs the best, whereas the Laplace and Student-t distributions are computationally expensive. Using a Gaussian distribution as the likelihood function assures the best median angular error and the smallest outlier influence.

Table VIII. Angular errors of GPR, using different likelihoods on the Shi[27] image set.

Likelihood	Median	Mean	Worst-1%	Worst-5%
Gauss	2.4437	3.2075	16.6449	11.4277
Laplace	2.6281	3.2691	17.2114	11.4379
Cauchy	2.7724	3.6563	18.4166	13.2057

Table IX. Angular errors of GPR, using different likelihoods on the Bianco[28] image set.

Likelihood	Median	Mean	Worst-1%	Worst-5%
Gauss	3.0758	4.0108	17.5956	13.5710
Laplace	3.5545	4.9095	17.9537	15.7990
Cauchy	3.5465	4.3236	17.7222	13.6743

Table X. Angular errors of RVM, using different kernels on the Shi[27] image set.

Kernel	Median	Mean	Worst-1%	Worst-5%
Gauss	2.5967	3.4527	17.5775	12.4885
Laplace	2.5764	3.4103	17.4340	11.9952
Cauchy	2.6716	3.4103	17.3602	12.0775

Table XI. Angular errors of RVM, using different kernels on the Bianco[28] image set.

Kernel	Median	Mean	Worst-1%	Worst-5%
Gauss	3.1859	4.1499	17.6866	14.3565
Laplace	3.2498	4.2084	17.6069	14.3056
Cauchy	3.1906	4.1681	17.6052	14.3081

Tables X and XI show angular errors using three different kernel functions for RVM with 15-fold cross-validation on two image sets. All the results are obtained via using the optimized kernel width listed in Table II.

From Tables X and XI, we can see that the average angular errors of the worst 5% and 1% decrease when Laplace and Student-*t* kernel functions are used, which implies that heavy-tailed distributions can enhance the performance of RVM for color constancy, but with large median and mean angular errors.

CONCLUSION

Two Bayesian kernel methods, namely GPR and RVM for regression, are used for addressing color constancy. More than seven kinds of covariance functions and their combinations for GPR and three kernel functions for RVM are used on two real image sets to find the best kernel function for color constancy.

Experimental results show that GPR using Matérn class covariance functions performs better than other covariance functions. Among the three kernel based methods, GPR outperforms RVM when using stationary covariance functions and can almost achieve the same performance as SVR. The performance of RVM is almost the same as that of GPR using a dot product covariance function. However, the predictive standard deviation of RVM is less than that of GPR on the same image set. The performances of the three kernel based methods are better than that of ridge regression. The analyses of the influence of outliers on data with Gaussian noise show that using heavy-tailed Laplace and Student-*t* likelihood functions cannot achieve better performance than using a Gaussian form for GPR. However, Laplace and Student-*t* kernels can decrease the average angular errors of the worst 5% and 1% for RVM, at the risk of large median and mean angular errors.

ACKNOWLEDGMENT

We thank all the reviewers for their helpful comments. This work was supported by the Natural Science Foundation of China under grants Nos 90820013 and 61033013. We would like to thank Professor C. E. Rasmussen and H. Nickisch for their free software GPML toolbox[40] and A. Thayananthan et al. for their free software multivariate RVM.[41]

REFERENCES

[1] E. Land, "The retinex theory of color constancy," Sci. Am. **237**, 108–129 (1977).

[2] G. Buchsbaum, "A spatial processor model for object colour perception," J. Franklin Inst. **310**, 1–26 (1980).

[3] J. Van De Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," IEEE Trans. Image Process. **16**, 2207–2214 (2007).

[4] G. Finlayson and E. Trezzi, "Shades of gray and colour constancy," *Proc. IS&T/SID Twelfth Color Imaging Conf.* (IS&T, Springfield, VA, 2004), pp. 37–41.

[5] D. Forsyth, "A novel algorithm for color constancy," Int. J. Comput. Vision **5**, 5–36 (1990).

[6] G. Finlayson and S. Hordley, "Improving gamut mapping color constancy," IEEE Trans. Image Process. **9**, 1774–1783 (2000).

[7] A. Gijsenij, T. Gevers, and J. van de Weijer, "Generalized gamut mapping using image derivative structures for color constancy," Int. J. Comput. Vision **86**, 127–139 (2010).

[8] D. Brainard and W. Freeman, "Bayesian color constancy," J. Opt. Soc. Am. A **14**, 1393–1411 (1997).

[9] C. Rosenberg, T. Minka, and A. Ladsariya, "Bayesian color constancy with non-gaussian models," Adv. Neural Inf. Process. Syst. **16**, (2003).

[10] P. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian color constancy revisited," *IEEE Conf. on Computer Vision and Pattern Recognition, 2008* (IEEE, 2008), pp. 1–8.

[11] V. Cardei, B. Funt, and K. Barnard, "Modeling color constancy with neural networks," *Proc. Int'l. Conf. on Vision, Recognition, and Action: Neural Models of Mind and Machine* (1997), pp. 29–31.

[12] V. Cardei, B. Funt, and K. Barnard, "Estimating the scene illumination chromaticity by using a neural network," J. Opt. Soc. Am. A **19**, 2374–2386 (2002).

[13] 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).

[14] W. Xiong and B. Funt, "Estimating illumination chromaticity via support vector regression," J. Imaging Sci. Technol. **50**, 341–348 (2006).

[15] N. Wang, D. Xu, and B. Li, "Edge-based color constancy via support vector regression," IEICE Trans. Inform. Syst. **92**, 2279–2282 (2009).

[16] V. Agarwal, A. Gribok, A. Koschan, B. Abidi, and M. Abidi, "Illumination chromaticity estimation using linear learning methods," J. Pattern Recognit. Res. **4**, 92–109 (2009).

[17] V. Agarwal, A. Gribok, and M. Abidi, "Machine learning approach to color constancy," Neural Networks **20**, 559–563 (2007).

[18] V. Agarwal, A. Gribok, A. Koschan, and M. Abidi, "Estimating illumination chromaticity via kernel regression," *IEEE Int'l. Conf. on Image Processing* (IEEE, 2006), pp. 981–984.

[19] C. E. Rasmussen and C. K. I. Williams, "Gaussian processes for machine learning," *Adaptive Computation and Machine Learning* (The MIT Press, Cambridge, Massachusetts, London, England, 2006).

[20] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," J. Mach. Learn. Res. **1**, 211–244 (2001).

[21] C. M. Bishop, "Bayesian regression and classification," *Advances in Learning Theory: Methods, Models and Applications* (IOS Press, 2003), pp. 267–285.

[22] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006).

[23] R. M. Neal, "Bayesian learning by neural networks," Ph.D. Thesis (University of Toronto, 1995).

[24] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis* (Cambridge Univ. Press, 2004).

[25] A. E. Hoerl and R. W. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," Technometrics **12**, 55–67 (1970).

[26] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Stat. Comput. **14**, 199–222 (2004).

[27] L. Shi and B. V. Funt, "Reprocessed version of the Gehler color constancy database of 568 images," 2011. http://www.cs.sfu.ca/colour/data.

[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).

[29] A. Gijsenij, T. Gevers, and J. van de Weijer, "Computational color constancy: survey and experiments," IEEE Trans. Image Process. **20**, 2475–2489 (2011).

[30] M. Ebner, *Color Constancy* (John Wiley & Sons, England, 2007).

[31] S. Hordley, "Scene illuminant estimation: past, present, and future," Color Res. Appl. **31**, 303–314 (2006).

[32] V. Agarwal, B. Abidi, A. Koschan, and M. Abidi, "An overview of color constancy algorithms," J. Pattern Recogn. Res. **1**, 42–54 (2006).

[33] E. L. Snelson, "Flexible and efficient Gaussian process models for machine learning," Ph.D. Thesis (Gatsby Computational Neuroscience Unit, University College London, 2007).

[34] V. Vapnik, *Statistical Learning Theory* (Wiley, New York, 1998).

[35] V. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," *Advances in Neural Information Processing Systems 9* (MIT Press, 1996).

[36] K. Barnard, V. Cardei, and B. Funt, "A comparison of computational color constancy algorithms. i: Methodology and experiments with synthesized data," IEEE Trans. Image Process. **11**, 972–984 (2002).

[37] F. Ciurea and B. Funt, "A large image database for color constancy research," *Proc. IS&T/SID Eleventh Color Imaging Conf.* (IS&T, Springfield, VA, 2003) pp. 160–164.

[38] M. Stein, *Interpolation of Spatial Data: Some Theory for Kriging* (Springer-Verlag, 1999).

[39] J. Von Kries, "Influence of adaptation on the effects produced by luminous stimuli," Sources of Color Vision 109–119 (1970).

[40] C. E. Rasmussen and H. Nickisch, "Gpml: Gaussian processes for machine learning toolbox," 2010. http://gaussianprocess.org/gpml/code .

[41] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and R. Cipolla, "Multivariate relevance vector machines for tracking," *Proc. 9th European Conf. on Computer Vision–ECCV 2006* (Springer, 2006), pp. 124–138.