# As-Projective-As-Possible Bias Correction for Illumination Estimation Algorithms

**Mahmoud Afifi[1], Abhijith Punnappurath[1], Graham Finlayson [2], and Michael S. Brown[1]**

[1] *Department of Electrical Engineering and Computer Science, York University, Toronto, Ontario, Canada*
[2] *School of Computing Sciences, The University of East Anglia, Norwich, United Kingdom*
[*] *{mafifi,pabhijith,mbrown}@eecs.yorku.ca, g.finlayson@uea.ac.uk*

**Abstract:** Illumination estimation is the key routine in a camera's onboard auto-white-balance (AWB) function. Illumination estimation algorithms estimate the color of the scene's illumination from an image in the form of an R,G,B vector in the sensor's raw-RGB color space. While learning-based methods have demonstrated impressive performance for illumination estimation, cameras still rely on simple statistical-based algorithms that are less accurate but capable of executing quickly on the camera's hardware. An effective strategy to improve the accuracy of these fast statistical-based algorithms is to apply a post-estimate *bias correction* function to transform the estimated R,G,B vector such that it lies closer to the correct solution. Recent work by Finlayson, Interface Focus, 2018 showed that a bias correction function can be formulated as a projective transform because the magnitude of the R,G,B illumination vector does not matter to the AWB procedure. This paper builds on this finding and shows that further improvements can be obtained by using an as-projective-as-possible (APAP) projective transform that locally adapts the projective transform to the input R,G,B vector. We demonstrate the effectiveness of the proposed APAP bias correction on several well-known statistical illumination estimation methods. We also describe a fast lookup method that allows the APAP transform to be performed with only a few lookup operations.



## 1. Introduction and Related Work

Color constancy is the term given to the human visual system's ability to perceive objects as the same color even when viewed under different illuminations [1]. Camera sensors lack this ability and unprocessed raw camera images contain noticeable color cast due to the scene's illumination. To compensate for scene illumination, cameras perform onboard computational color constancy that makes up their auto-white-balance (AWB) functionality. AWB is a two-step procedure that involves: (1) estimating the color of the illumination in the camera's color space, and (2) correcting the image based on the estimated illumination.

This paper addresses the illumination estimation step in computational color constancy. The goal of illumination estimation is to determine the R,G,B response of the camera's raw sensor to the scene's illumination. The most straightforward way to do this is to capture an image of an object that acts as a pure reflector–for example, an achromatic (i.e., white or neutral) object. Under pure white light, an ideal camera sensor's response to the achromatic object should lie along the achromatic "white line" (i.e., R=G=B). Under real illuminations, the camera's response to a pure reflector does *not* lie along the achromatic line and the R,G,B response represents the measurement of the illumination in the sensor's raw-RGB color space. An interesting observation is that the magnitude of the R,G,B response does not matter; any uniform scale factor (i.e., $\alpha$R,$\alpha$G,$\alpha$B) of the camera's response represents the same illumination color. The change in $\alpha$ is due to varying radiant power of the scene's illumination as well as settings on the camera, such as exposure and gain.

In practice, we do not have neutral patterns in our scenes and the color of the illumination

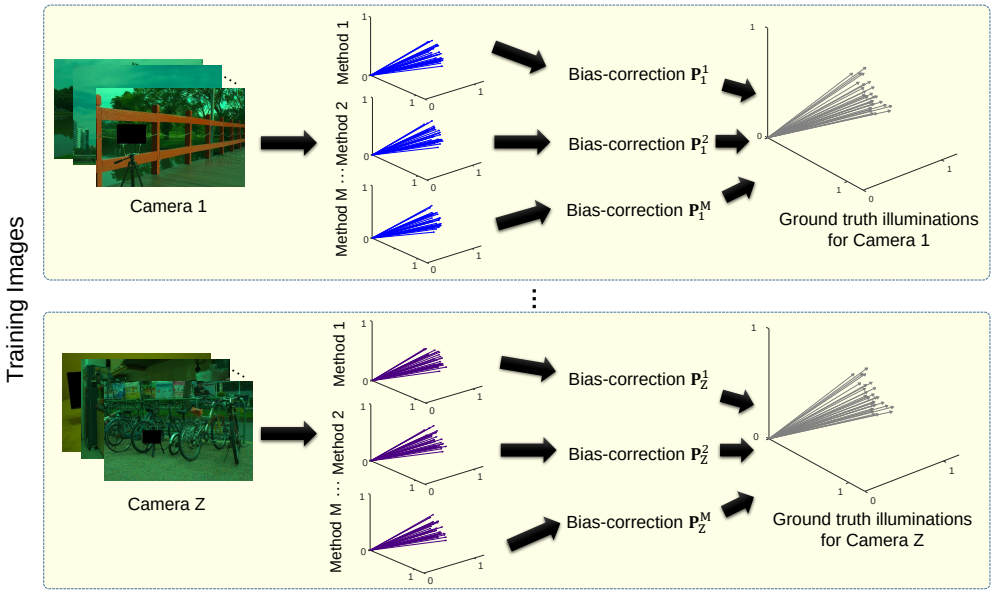


Fig. 1. This figure shows the basic idea of improving an illumination estimation algorithm using a bias-correction function (denoted as $\mathbf{P}_i^j$, where $i$ represents a particular camera model, and $j$ denotes the illumination estimation method). For a given camera model, statistical-based illumination estimation methods tend to fail in a systematic way. A bias-correction function can be estimated based on training data such that an illumination estimation method's estimates are corrected in order to lie closer to the ground truth solution.

must instead be estimated directly from the image. The task of illumination estimation is a well-studied area of computer vision. Illumination estimation methods fall roughly into two categories: statistical-based methods and learning-based methods. Statistical-based methods use statistics from an image's color distribution and spatial layout to determine the R,G,B illumination vector. Representative examples include gray world (GW) [2], general gray world (GGW) [3], gray edges (GE) [4], shades of gray (SoG) [5], white patch [6], bright pixels [7], and PCA [8]. Learning-based methods rely on training data with examples where the illumination is known (e.g., by placing a neutral object in the scene) and use various strategies to estimate or predict the illumination. Representative examples include [9–16]. In recent years, learning-based methods employing deep-learning techniques have shown state-of-the-art performance [16].

Learning-based methods, however, suffer from a substantial increase in complexity, with deep network architectures requiring millions of parameters. In the absence of specialized chips or GPUs, the computational and memory requirements associated with running these methods onboard the camera are still prohibitive. As a result, cameras currently rely on simple statistical methods even though these methods are not as accurate as their learning-based counterparts.

A less explored alternative to improve the accuracy of statistical methods, while retaining their low complexity characteristic, is to apply a post-estimate *bias correction*. The general idea of bias correction is to rectify systematic estimation errors (or biases) in an algorithm by mapping the algorithm's estimates closer to the ground truth using a bias correction function. The bias-correction function can be computed per camera for a given illumination estimation method. Figure 1 shows an illustration of this idea. The effectiveness of illuminant estimation bias correction was demonstrated by Finlayson [17]. In [17], Finlayson proposed a linear bias

correction function based on the first-, second-, and higher-order moments of the image colors or image color derivatives.

More recently, Finlayson [18] showed that bias correction could be formulated as a 3×3 projective matrix transform. The work in [18] noted that since the illuminant can be estimated only up to a scale factor, the three-dimensional R,G,B vector can be interpreted as a color *ray* in projective space. Thus, the role of the projective bias correction is to determine a projective function that maps estimated illuminant R,G,B rays to their corresponding ground truth R,G,B rays. This 'ray mapping' is analogous to the global projective transform employed in computer vision (referred to as a homography) that is used to map image rays between two camera views. The work in [18] demonstrated the effectiveness of the projective bias correction on GE [4].

**Contribution** The work in this paper builds on the idea proposed in [18]. First, we provide an overview of bias correction using a global projective transform and show its effectiveness when used with *several* different statistical-based algorithms. Next, we extend the global projective transformation to use an as-projective-as-possible (APAP) transform that locally adapts the transform to the input R,G,B vector. APAP provides a more accurate result, but has a higher computational demand because the local transformation needs to be estimated independently for each input. To reduce the computational load, we describe a lookup table (LUT) procedure that circumvents the need to compute an APAP projective transform for each input. We demonstrate the effectiveness of our APAP projective bias correction technique on several statistical methods and show that estimation errors can be reduced up to 24%-30% over the global transform.

## 2. Preliminaries

We begin with a short introduction to the idea of bias correction using a global projective transform proposed in [18]. A system of linear equations that has the following form appears frequently in projective geometry

$$\alpha_i \mathbf{y}_i = \mathbf{P}\mathbf{x}_i. \tag{1}$$

The form of equation (1) is akin to an ordinary system of linear equation except for the scale factor $\alpha_i$. Because of this, $\mathbf{P}$ is considered a projective transformation that can map $\mathbf{x}_i$ to $\mathbf{y}_i$ only within some unknown multiplicative factor.

In the illumination estimation problem, $\mathbf{P}$ will take the form of a 3×3 matrix. We can define $\mathbf{x} = [\ R\ G\ B\ ]^T$ to be an R,G,B estimate of the illumination by some algorithm, and $\mathbf{y}$ is used to represent the corresponding ground truth R,G,B value of the illumination. From training data, we can obtain a set of $N$ corresponding $(\mathbf{x}_i, \mathbf{y}_i)$ where $i = 1, ..., N$. As discussed in Section 1, the magnitude of the R,G,B response does not matter in an AWB problem and the problem takes on the same form as equation (1).

In [18], Finlayson proposed an alternating linear least squares procedure to solve for $\mathbf{P}$. The estimated $\mathbf{P}$ served as a bias-correction function that mapped newly estimated $\mathbf{x}_t$ to a more accurate estimate $\mathbf{y}_t$. This global projective mapping is illustrated in Figure 2. The work in this paper builds on this initial idea but extends the global $\mathbf{P}$ to be a locally weighted projective transformation.

## 3. Proposed APAP Method

In this section, we first describe how the global projective transform is solved using an alternating least squares solver. We then develop the APAP model that improves upon the global case by appropriately weighting the training data based on the input. Finally, we explain how the entire bias correction framework can be reduced to simple lookup operations followed by multiplication with the selected 3×3 projective transform matrices.

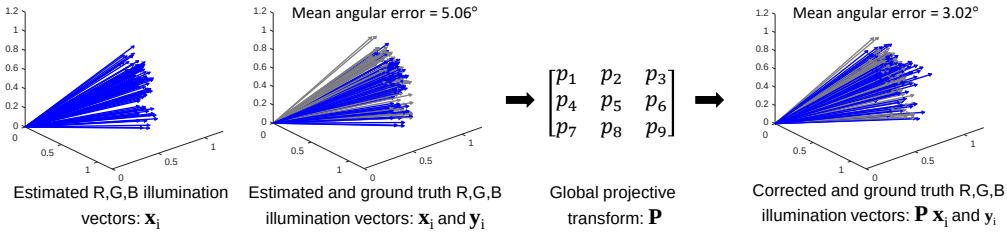


Fig. 2. This figure provides an overview of the use of global projective bias correction for illumination estimation. A statistical-based algorithm is used to estimate the R,G,B illumination vectors from a set of training images. These are denoted as $\mathbf{x}_i$. The corresponding R,G,B ground truth values are obtained from color rendition charts placed in the training images' scenes. These are represented as $\mathbf{y}_i$. A global 3×3 transform $\mathbf{P}$ is solved according to equation (1). The matrix $\mathbf{P}$ is used to map $\mathbf{x}_i$ such that it lies closer to the true solution $\mathbf{y}_i$.

### *3.1. Global projective transformation*

As described in the previous section, a global projective transform $\mathbf{P}$ can be estimated from a training set of $N$ images with associated ground truth. The training images have a color rendition chart added into the scene. For each training image, the ground truth R,G,B value of the scene's illumination is obtained by averaging the R,G,B values of the achromatic patches on the imaged color chart.

A matrix $\mathbf{A}$ of size $3\times N$ is constructed to contain the algorithm's R,G,B estimate of the illuminants $\{\mathbf{x}_i\}_{i=1}^{N}$, while a $3\times N$ matrix $\mathbf{B}$ holds the corresponding ground truth values $\{\mathbf{y}_i\}_{i=1}^{N}$. The projective transformation matrix $\mathbf{P}$ for bias correction can then be obtained by solving the following least squares problem:

$$\{\mathbf{P}, \mathbf{D}\} = \arg\min_{\mathbf{P},\mathbf{D}} ||\mathbf{PAD} - \mathbf{B}||_F, \tag{2}$$

where the $N\times N$ diagonal matrix $\mathbf{D}$ is introduced to account for the differences in scale between the input and output vectors, and $||.||_F$ denotes the Frobenius norm. The auxiliary variable $\mathbf{D}$ allows us to reduce the projective estimation problem to an ordinary linear least squares problem.

However, since both $\mathbf{P}$ and $\mathbf{D}$ are unknown, we solve for the two variables using an alternating least squares [19, 20] procedure described in Algorithm 1. Please refer to [19] for a detailed analysis of the convergence of Algorithm 1. Solving for $\mathbf{P}$ in this fashion finds a projective transformation that achieves the best *global* fit to the data. It is important to note that once we have solved for $\mathbf{P}$, we do not need to consider the magnitude of the estimate, since we can recover the illuminant only up to a scalar. Note also that the matrix $\mathbf{D}$ is required only when solving equation (2) and plays no role when $\mathbf{P}$ is used for bias correction.

For a new test image with an estimated illuminant $\mathbf{x}_t$, the bias-corrected illuminant $\mathbf{y}_t$ can be obtained as $\mathbf{y}_t = \mathbf{P}\mathbf{x}_t$.

### *3.2. As-projective-as-possible transformation*

An even better fit to the data can be achieved if the projective transform is modified to locally adapt to the training data. This idea was demonstrated by Zaragoza et al. [21] for computing projective planar transforms for use in image stitching. The basic idea of the APAP transform is to weight the contribution of training data closer to the test input data when estimating the projective transform.

We adapt the method of [21] to our problem of bias correction. Towards this end, we first define a function that returns a weight between 0 and 1, indicating how close the test illuminant

**Algorithm 1** Estimating a projective transformation using alternating least squares.

**Input:** Matrix $\mathbf{A}$ containing the $N$ R,G,B estimates of the illuminant obtained using the chosen illumination estimation algorithm, and matrix $\mathbf{B}$ containing the corresponding ground truth R,G,B values of the light

**Output:** The 3×3 projective bias correction matrix $\mathbf{P}$, and an auxiliary variable $\mathbf{D}$ that compensates for the difference in magnitude between the estimated illuminants and their corresponding ground truths

1: $q = 0, \mathbf{D}^{(0)} = \arg\min_{\mathbf{D}} ||\mathbf{AD} - \mathbf{B}||_F$
2: **do**
3: $\quad q = q + 1$
4: $\quad \mathbf{P}^{(q)} = \arg\min_{\mathbf{P}} ||\mathbf{PAD}^{(q-1)} - \mathbf{B}||_F$
5: $\quad \mathbf{D}^{(q)} = \arg\min_{\mathbf{D}} ||\mathbf{P}^{(q)}\mathbf{AD} - \mathbf{B}||_F$
6: **while** $||\mathbf{D}^{(q)} - \mathbf{D}^{(q-1)}||_F >$ threshold

estimate $\mathbf{x}_t$ is to the training data $\{\mathbf{x}_i\}_{i=1}^{N}$. The weights are calculated as

$$w_{i_t} = \exp(-\Theta(\mathbf{x}_t, \mathbf{x}_i)/\sigma_w^2), \tag{3}$$

where $\Theta$ is the angular error [22] between $\mathbf{x}_t$ and $\mathbf{x}_i$, and $\sigma_w$ is a scale parameter. The scalar weights $w_{i_t}$ give higher importance to data that are closer to the test image's illuminant estimate $\mathbf{x}_t$.

For a given test image with estimated illuminant $\mathbf{x}_t$, we solve a weighted least squares problem of the form

$$\{\mathbf{P}_t, \mathbf{D}_t\} = \arg\min_{\mathbf{P},\mathbf{D}} ||\mathbf{PAW}_t\mathbf{D} - \mathbf{BW}_t||_F, \tag{4}$$

where $\mathbf{W}_t = \text{diag}(w_{i_t}), i = 1$ to $N$.

The intuition behind using $\mathbf{W}_t$ is that assigning higher weights to data closer to $\mathbf{x}_t$ produces a projective transformation $\mathbf{P}_t$ that better respects the local structure around $\mathbf{x}_t$. This is in contrast to our first model that uses a single and global $\mathbf{P}$ for all $\mathbf{x}_t$. The net effect of the weighting matrix $\mathbf{W}_t$ is to produce an overall transformation that adapts flexibly to the data, yet preserving the projective trend of the transformation. The weighted least squares fit $\mathbf{P}_t$ can be obtained by solving equation (4) using the same alternating least squares procedure described in Algorithm 1, and the bias-corrected estimate can be calculated as $\mathbf{y}_t = \mathbf{P}_t\mathbf{x}_t$.

In practice, we compute the weights $w_{i_t}$ by introducing a small offset $\gamma$ within 0 to 1 as

$$w_{i_t} = \max\left(\exp(-\Theta(\mathbf{x}_t, \mathbf{x}_i)/\sigma_w^2), \gamma\right). \tag{5}$$

This is to prevent numerical instabilities while solving equation (4) in cases where $\mathbf{x}_t$ is an extrapolation region and most of the weights are insignificant. Note that as $\gamma$ approaches unity, the transformation reduces to the original global transformation.

### *3.3. Lookup table speedup*

A disadvantage of the APAP bias correction technique in Section 3.2 is the computational overhead in estimating the local projective transformation. While the global projective transform in Section 3.1 can be computed offline and stored, the local APAP correction matrix has to be estimated separately for each input image at test time.

We address this issue by using a lookup table based on the observation that estimated illuminant values in the neighbourhood of $\mathbf{x}_t$ will produce similar weights (3), and thereby

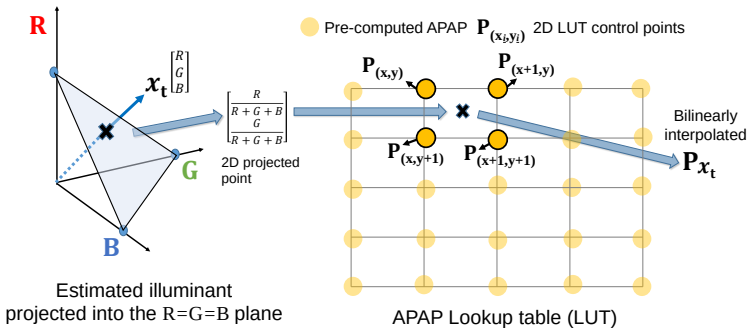


Fig. 3. To reduce the computational overhead in estimating the local APAP transformation, we build a lookup table by binning the space of estimated illuminant values into an $L \times L$ grid. The blue mark represents the initial estimated illuminant vector $\mathbf{x}_t$. The dark yellow circles represent the four nearest pre-computed illuminants in the LUT, which are used for bilinear interpolation.

similar local transforms. Note that the information in an estimated illuminant $\mathbf{x} = [\ R\ G\ B\ ]^T$, which is in homogeneous projective form, can be encoded by the inhomogeneous two-vector $\left[\ \frac{R}{R+G+B}\ \ \frac{G}{R+G+B}\ \right]^T$ since the absolute scale is not important. We can therefore discretize the 3D space of estimated illuminant values into an 2D $L \times L$ grid. We then compute the weights and the projective transformations associated with each node, and store these transformations in a 2D LUT.

At test time, for an input image having an estimated illuminant $\mathbf{x}_t$, we bilinearly interpolate the answer using our lookup table to obtain the bias-corrected value. An example of this process is shown in Figure 3. The lookup operation is extremely fast and reduces the running time of the APAP approach to be comparable to the global transformation case. In our experiments, we found that a 16×16 LUT provides results comparable to using the actual $\mathbf{x}_t$ value in equation (3). If the entries in the table are saved as type `float64`, then the size of the LUT is only about 18 KB, making the memory requirements associated with storing the LUT negligible.

## 4. Results

We evaluated our method on three different datasets, which are: (i) the NUS dataset [8], (ii) the reprocessed version [23] of Gehler *et al.*'s dataset [11], and (iii) the INTEL-TUT dataset [26]. As there are different ground-truth illuminants available [27] for the Gehler-Shi dataset, we recalculated the ground-truth illuminants from the color rendition chart provided in each image of the dataset. For both NUS and Gehler-Shi datasets, the images were normalized using the black and saturation levels provided per camera in each dataset. For the INTEL-TUT dataset, the images already have black-level correction applied, and only saturation stretching was applied. For the mobile phone images in the INTEL-TUT dataset, the experiments were performed with and without color shading correction.

We applied our proposed projective bias correction transformations to four different statistical-based methods – namely (i) GW [2], (ii) GE [4], (iii) SoG [5], and (iv) distribution PCA [8]. In our experiments, we used $\sigma_w = 3.0$ and $\gamma = 0.0625$ for our APAP bias correction. The LUT consisted of 16×16 bins. As the camera hardware may use a down-sampled version of the captured image for AWB estimation, following [16], our correction was applied on the illuminants estimated using 384×256 16-bit raw-RGB images, instead of the original full-resolution raw-RGB images. For SoG, the Minkowski norm ($p$) was set to 4. The first and second differentiations were used for GE with $p = 6$ and $\sigma = 2$. For the distribution PCA method [8], the selected

Table 1. Results on the **NUS 8-Camera** [8] dataset. The recovery angular error is reported for the statistical-based methods, learning-based methods, and the proposed projective correction transformations applied to the statistical-based methods. The statistical-based methods are: gray world (GW) [2], shades of gray (SoG) [5], the first-order gray edges (GE-1) and the second-order gray edges (GE-2) [4], and the distribution PCA [8]. The learning-based methods are: Bayesian [11], convolutional color constancy (CCC) [12], deep specialized network (DS-Net) [13], the FC4 method based on AlexNet (FC4-A) and SqueezeNet (FC4-S) [15], and fast Fourier color constancy (FFCC) [16]. The proposed projective bias correction is applied on the statistical-based methods using a down-sampled version of the images (384×256 pixels). The term (APAP) denotes that the as-projective-as-possible transformation is applied. The term (APAP-LUT) refers to the APAP using a $16^2$-bins lookup table. The bold numbers refer to the state-of-the-art results reported on the dataset.

| | Method | Original methods | | | | Global projective | | | | APAP | | | | APAP-LUT | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Mean | Median | Best 25% | Worst 25% | Mean | Median | Best 25% | Worst 25% | Mean | Median | Best 25% | Worst 25% | Mean | Median | Best 25% | Worst 25% |
| Statistical-based | GW [2] | 4.17 | 3.15 | 0.87 | 9.23 | 2.64 | 1.93 | 0.64 | 5.81 | 2.40 | 1.76 | 0.55 | 5.42 | 2.52 | 1.83 | 0.60 | 5.62 |
| | SoG [5] | 3.28 | 2.46 | 0.76 | 7.24 | 2.75 | 2.04 | 0.64 | 6.02 | 2.43 | 1.69 | 0.53 | 5.56 | 2.55 | 1.90 | 0.57 - | 5.65 |
| | GE-1 [4] | 5.08 | 3.38 | 0.90 | 12.14 | 3.71 | 2.34 | 0.80 | 9.08 | 3.16 | 1.93 | 0.54 | 8.13 | 3.11 | 2.14 | 0.63 | 7.39 |
| | GE-2 [4] | 5.90 | 4.18 | 1.22 | 13.13 | 3.31 | 2.30 | 0.68 | 7.79 | 3.01 | 1.86 | 0.56 | 7.57 | 3.09 | 2.08 | 0.63 | 7.41 |
| | PCA [8] | 4.30 | 2.95 | 0.79 | 10.05 | 2.90 | 2.22 | 0.68 | 6.25 | 2.45 | 1.72 | 0.52 | 5.67 | 2.59 | 1.96 | 0.59 | 5.69 |
| Learning-based | Bayesian [11] | 3.50 | 2.36 | 0.78 | 8.02 | - | - | - | - | - | - | - | - | - | - | - | - |
| | CCC [12] | 2.38 | 1.48 | 0.45 | 5.85 | - | - | - | - | - | - | - | - | - | - | - | - |
| | DS-Net [13] | 2.24 | 1.46 | 0.48 | 6.08 | - | - | - | - | - | - | - | - | - | - | - | - |
| | FC4-A [15] | 2.12 | 1.53 | 0.48 | 4.78 | - | - | - | - | - | - | - | - | - | - | - | - |
| | FC4-S [15] | 2.23 | 1.57 | 0.47 | 5.15 | - | - | - | - | - | - | - | - | - | - | - | - |
| | FFCC [16] | **1.99** | **1.31** | **0.35** | **4.75** | - | - | - | - | - | - | - | - | - | - | - | - |


Fig. 4. Projective bias correction of the estimated illuminant of the gray world algorithm (GW) [2]. (A) A raw Canon EOS-1Ds image from the NUS dataset [8]. (B) The corrected image using the GW algorithm. (C) The corrected image using the global projective transformation (GP). (D) The corrected image using the as-projective-as-possible with the lookup table (APAP-LUT). (E) The ground truth image. The images in (B-E) were rendered to sRGB color space for better visualization using the camera-pipeline software of [28].

percentage rate was 3.5%.

For the reprocessed version [23] of Gehler *et al.*'s dataset [11], we applied our bias correction to three learning-based methods as well: (i) Bayesian-based method [11], (ii) color constancy using natural image statistics and scene semantics (CCNIS) [24], and (iii) exemplar-based color constancy [25]. The initial estimations of the learning-based methods were obtained from http://colorconstancy.com.

We used three-cross-fold validation for each camera in each dataset, where two folds were used for training, and the testing was carried out using the third fold. For the NUS and Gehler-Shi datasets, the color chart was masked out during both training and testing.

For evaluation, we adopt the recovery angular error [22], also known as angular error, that measures the angle between the estimated scene illuminant and the ground truth illuminant.

An example from the NUS dataset is provided in Figure 4. As shown, our correction improves the initial estimate of the simple GW algorithm; the global projection and APAP with the LUT

Table 2. Results on the **Gehler-Shi** [11, 23] dataset. The recovery angular error is reported for the statistical-based methods, learning-based methods, and the proposed projective transformations. The statistical-based methods are: gray world (GW) [2], shades of gray (SoG) [5], the first-order gray edges (GE-1) and the second-order gray edges (GE-2) [4], and the distribution PCA [8]. The learning-based methods are: Bayesian [11], color constancy using natural image statistics and scene semantics (CCNIS) [24], exemplar-based color constancy [25], convolutional color constancy (CCC) [12], deep specialized network (DS-Net) [13], Seoung *et al.*'s method [14], the FC4 method based on AlexNet (FC4-A) and SqueezeNet (FC4-S) [15], and fast Fourier color constancy (FFCC) [16]. The proposed projective bias correction is applied on the statistical-based methods (i.e., GW, SoG, GE [first- and second-orders], and the distribution PCA methods) using a down-sampled version of the images (384×256 pixels). We also applied our transformations on three learning-based methods (i.e., Bayesian [11], CCNIS [24], and Exemplar-based [25]).

| | Method | Original methods | | | | Global projective | | | | APAP | | | | APAP-LUT | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Mean | Median | Best 25% | Worst 25% | Mean | Median | Best 25% | Worst 25% | Mean | Median | Best 25% | Worst 25% | Mean | Median | Best 25% | Worst 25% |
| Statistical-based | GW [2] | 4.90 | 3.74 | 1.04 | 10.83 | 3.13 | 2.40 | 0.64 | 6.89 | 2.76 | 2.02 | 0.53 | 6.21 | 2.96 | 2.22 | 0.59 | 6.58 |
| | SoG [5] | 3.77 | 2.32 | 0.53 | 9.39 | 3.38 | 2.30 | 0.54 | 8.08 | 3.13 | 2.04 | 0.51 | 7.61 | 3.29 | 2.20 | 0.54 | 7.88 |
| | GE-1 [4] | 5.57 | 3.52 | 0.95 | 13.61 | 4.23 | 2.64 | 0.63 | 10.48 | 3.98 | 2.37 | 0.57 | 10.10 | 4.09 | 2.40 | 0.61 | 10.29 |
| | GE-2 [4] | 6.05 | 3.76 | 1.09 | 14.61 | 4.32 | 2.56 | 0.66 | 10.85 | 4.26 | 2.48 | 0.55 | 11.05 | 4.21 | 2.50 | 0.61 | 10.70 |
| | PCA [8] | 4.17 | 2.62 | 0.53 | 10.26 | 3.57 | 2.32 | 0.59 | 8.53 | 3.26 | 2.07 | 0.54 | 8.03 | 3.42 | 2.27 | 0.57 | 8.24 |
| Learning-based | Bayesian [11] | 4.82 | 3.46 | 1.26 | 10.49 | 4.67 | 3.12 | 0.93 | 11.06 | 4.61 | 2.97 | 0.93 | 10.98 | 4.50 | 2.90 | 0.88 | 10.63 |
| | CCNIS [24] | 4.19 | 3.13 | 1.00 | 9.24 | 3.49 | 2.47 | 0.82 | 8.01 | 3.51 | 2.52 | 0.84 | 7.97 | 3.61 | 2.65 | 0.89 | 8.04 |
| | Exemplar-based [25] | 2.89 | 2.27 | 0.82 | 5.98 | 3.01 | 2.38 | 0.87 | 6.32 | 3.02 | 2.43 | 0.92 | 6.27 | 3.09 | 2.39 | 0.98 | 6.36 |
| | CCC [12] | 1.95 | 1.22 | 0.35 | 4.76 | - | - | - | - | - | - | - | - | - | - | - | - |
| | DS-Net [13] | 1.90 | 1.12 | 0.31 | 4.84 | - | - | - | - | - | - | - | - | - | - | - | - |
| | Seoung et al. [14] | 2.16 | 1.47 | 0.37 | - | - | - | - | - | - | - - | - | - | - | - | - | - |
| | FC4-A [15] | 1.77 | 1.11 | 0.34 | 4.29 | - | - | - - | - | - | - | - | - | - | - | - | - |
| | FC4-S [15] | 1.65 | 1.18 | 0.38 | **3.78** | - | - | - | - | - | - | - | - | - | - | - | - |
| | FFCC [16] | **1.61** | **0.86** | **0.23** | 4.27 | - | - | - | - - | - | - | - | - | - | - | - | - |

Table 3. Results on the **INTEL-TUT** [26] dataset. The recovery angular errors are reported for statistical-based methods with and without our proposed projective transformations. The methods are: gray world (GW) [2], shades of gray (SoG) [5], the first-order gray edges (GE-1) and the second-order gray edges (GE-2) [4], and the distribution PCA [8]. The proposed projective bias correction is applied using a down-sampled version of the images (384×256 pixels).

| Method | Original methods | | | | Global projective | | | | APAP | | | | APAP-LUT | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean | Median | Best 25% | Worst 25% | Mean | Median | Best 25% | Worst 25% | Mean | Median | Best 25% | Worst 25% | Mean | Median | Best 25% | Worst 25% |
| GW [2] | 4.77 | 3.75 | 0.99 | 10.29 | 4.68 | 3.12 | 1.15 | 11.11 | 4.30 | 2.44 | 0.69 | 11.30 | 4.46 | 2.85 | 0.94 | 11.01 |
| SoG [5] | 4.99 | 3.63 | 1.08 | 11.20 | 4.37 | 2.87 | 1.04 | 10.49 | 3.99 | 2.14 | 0.60 | 10.82 | 4.11 | 2.62 | 0.84 | 10.17 |
| GE-1 [4] | 4.62 | 2.84 | 0.94 | 11.46 | 3.71 | 2.34 | 0.80 | 9.08 | 3.26 | 1.54 | 0.44 | 9.15 | 3.38 | 2.08 | 0.64 | 8.52 |
| GE-2 [4] | 4.82 | 2.97 | 1.03 | 11.96 | 3.10 | 2.11 | 0.74 | 7.20 | **2.73** | **1.47** | **0.41** | **7.34** | 2.87 | 1.89 | 0.59 | 6.90 |
| PCA [8] | 4.65 | 3.39 | 0.87 | 10.75 | 3.65 | 2.67 | 0.98 | 8.12 | 3.16 | 1.89 | 0.52 | 8.15 | 3.30 | 2.41 | 0.76 | 7.48 |

obtained 5.0 and 2.6 recovery angular errors, respectively, compared to 11.6 of the original GW estimation.

We reported the mean, median, best 25%, and worst 25% performance for each method before and after our correction, where the best 25% and worst 25% are the mean of the smallest 25% error values and the mean of the highest 25% error values, respectively.

Tables 1, 2, and 3 show the results on the NUS, Gehler-Shi, and INTEL-TUT datasets, respectively, using our proposed bias correction framework. In addition to the aforementioned methods, for comparison's sake, we reported the results of other learning-based methods [12–16] on the Gehler-Shi and NUS datasets. It is worth mentioning that the results of learning-based

methods were taken from past papers. As the INTEL-TUT dataset is recently published, the results of the learning-based methods are not available. Therefore, we reported only the results obtained using GW, GE, SoG, and distribution PCA methods with and without the proposed bias correction methods.

We would like to point out that the results of GW, GE, SoG, and distribution PCA methods may differ slightly from other reported results in the literature, because we used fixed parameters in all experiments instead of fine-tuning the parameters per camera.

The results using three different datasets show that the proposed projective bias correction consistently reduces the angular error. As expected, the APAP correction achieves better improvement than the global projective transformation, as the former applies a local correction generated in reliance on the nearest training samples, while the latter corrects the bias globally. Our proposed projective bias correction improves the simple statistical-based methods to yield results close to sophisticated learning-based methods – for example, our APAP correction using GW produces results that match some of the recent learning-based methods (e.g., CCC [12] on the NUS dataset [8]; see Table 1).

Because of the systematic failures [29] of the statistical-based methods, the amount of angular error reduction of the statistical-based methods is higher than that obtained using the learning-based methods. The error of the statistical-based methods is reduced by 24.9% (32.6% on NUS, 23.8% on Gehler-Shi, and 18.2% on INTEL-TUT datasets), 32.2% (40.8% on NUS, 28.9% on Gehler-Shi, and 26.87% on INTEL-TUT datasets), and 29.8% (39.0% on NUS, 26.5% on Gehler-Shi, and 24.0% on INTEL-TUT datasets) using the global projection, the APAP correction, and the APAP with the LUT, respectively, while the error reduction obtained for the learning-based methods is only 5.4%, 5.6%, and 5.1% on average using the global, APAP, and APAP with the LUT, respectively.

The proposed method not only achieves good accuracy but also runs faster than the state-of-the-art methods. Our Matlab implementation takes 1.435, 630, and 15.891 milliseconds on Intel Core CPU i7-6700 for the global projection, APAP, and APAP with the $16^2$-bins LUT, respectively. Note that the reported time is only for bias correction and does not include the time required for illuminant estimation. Compared to learning-based methods, such as the deep specialized network (DS-Net) [13], which takes 3 seconds per image on GPU, CCC [12], which requires 0.52 seconds per image, or the fast Fourier color constancy (FFCC) [16], which needs ~0.03 seconds, the proposed correction method requires only 0.011 seconds on average using the GW for the initial estimation. Our algorithm's simplicity and speed make it ideal for real-time scenarios.

## 5. Conclusions

This paper has proposed a method to improve the accuracy of statistical-based illumination estimation methods by applying an as-projective-as-possible bias-correction function. This work extends the finding in [18] that demonstrated a global projective transform could be used as an effective bias-correction function. The proposed APAP function adapts the projective transform locally to the training data to obtain further improvements in the bias-correction. To provide computational efficiency, we also described how to use a 16×16 lookup table to quickly approximate an APAP bias-correction based on a new input. We demonstrated our approach on several white-balance datasets and show consistent improvements between 24%-30% over the global projective bias correction on several statistical-based illumination estimation approaches.

## Acknowledgments

This study was funded in part by the Canada First Research Excellence Fund for the Vision: Science to Applications (VISTA) programme and an NSERC Discovery Grant.

## References


1. A. Gilchrist, *Seeing Black and White*, 40 (OUP USA, 2006).
2. G. Buchsbaum, "A spatial processor model for object colour perception," J. Frankl. institute **310**, 1–26 (1980).
3. K. Barnard, V. Cardei, and B. Funt, "A comparison of computational color constancy algorithms. i: Methodology and experiments with synthesized data," IEEE Transactions on Image Process. **11**, 972–984 (2002).
4. J. Van De Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," IEEE Transactions on Image Process. **16**, 2207–2214 (2007).
5. G. D. Finlayson and E. Trezzi, "Shades of gray and colour constancy," in *Color and Imaging Conference,* (2004).
6. D. H. Brainard and B. A. Wandell, "Analysis of the retinex theory of color vision," J. Opt. Soc. Am. A **3**, 1651–1661 (1986).
7. H. R. V. Joze, M. S. Drew, G. D. Finlayson, and P. A. T. Rey, "The role of bright pixels in illumination estimation," in *Color and Imaging Conference,* (2012).
8. D. Cheng, D. K. Prasad, and M. S. Brown, "Illuminant estimation for color constancy: Why spatial-domain methods work and the role of the color distribution," J. Opt. Soc. Am. A **31**, 1049–1058 (2014).
9. Z. Lou, T. Gevers, N. Hu, M. P. Lucassen *et al.*, "Color constancy by deep learning," in *BMVC,* (2015).
10. D. Cheng, B. Price, S. Cohen, and M. S. Brown, "Effective learning-based illuminant estimation using simple features," in *CVPR,* (2015).
11. P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian color constancy revisited," in *CVPR,* (2008).
12. J. T. Barron, "Convolutional color constancy," in *ICCV,* (2015).
13. W. Shi, C. C. Loy, and X. Tang, "Deep specialized network for illuminant estimation," in *ECCV,* (2016).
14. S. W. Oh and S. J. Kim, "Approaching the computational color constancy as a classification problem through deep learning," Pattern Recognit. **61**, 405–416 (2017).
15. Y. Hu, B. Wang, and S. Lin, "Fc4: Fully convolutional color constancy with confidence-weighted pooling," in *CVPR,* (2017).
16. J. T. Barron and Y.-T. Tsai, "Fast Fourier color constancy," in *CVPR,* (2017).
17. G. D. Finlayson, "Corrected-moment illuminant estimation," in *ICCV,* (2013).
18. G. D. Finlayson, "Colour and illumination in computer vision," Interface Focus. **8** (2018).
19. G. D. Finlayson, H. Gong, and R. Fisher, "Color homography: theory and applications," IEEE Transactions on Pattern Analysis Mach. Intell. pp. 1–1 (2018).
20. F. G. D., M. D. Maryam, and M. Michal, "The alternating least squares technique for nonuniform intensity color correction," Color. Res. & Appl. **40**, 232–242 (2015).
21. J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving DLT," in *CVPR,* (2013).
22. G. D. Finlayson, B. V. Funt, and K. Barnard, "Color constancy under varying illumination," in *ICCV,* (1995).
23. L. Shi, "Re-processed version of the Gehler color constancy dataset of 568 images," http://www.cs.sfu.ca/ color/data/.
24. A. Gijsenij and T. Gevers, "Color constancy using natural image statistics and scene semantics," IEEE Transactions on Pattern Analysis Mach. Intell. **33**, 687–698 (2011).
25. H. R. V. Joze and M. S. Drew, "Exemplar-based color constancy and multiple illumination," IEEE Transactions on Pattern Analysis Mach. Intell. **36**, 860–873 (2014).
26. Ç. Aytekin, J. Nikkanen, and M. Gabbouj, "A data set for camera-independent color constancy," IEEE Transactions on Image Process. **27**, 530–544 (2018).
27. G. D. Finlayson, G. Hemrit, A. Gijsenij, and P. Gehler, "A curious problem with using the colour checker dataset for illuminant estimation," in *Color and Imaging Conference,* (2017).
28. H. C. Karaimer and M. S. Brown, "A software platform for manipulating the camera imaging pipeline," in *ECCV,* (2016).
29. R. Zakizadeh, M. S. Brown, and G. D. Finlayson, "A hybrid strategy for illuminant estimation targeting hard images," in *IEEE International Conference on Computer Vision Workshops,* (2015).