Efficient and Robust Color Consistency for Community Photo Collections

Jaesik Park* Intel Labs
Yu-Wing Tai SenseTime
Sudipta N. Sinha Microsoft Research
In So Kweon KAIST

Abstract

We present an efficient technique to optimize color consistency of a collection of images depicting a common scene. Our method first recovers sparse pixel correspondences in the input images and stacks them into a matrix with many missing entries. We show that this matrix satisfies a rank two constraint under a simple color correction model. These parameters can be viewed as pseudo white balance and gamma correction parameters for each input image. We present a robust low-rank matrix factorization method to estimate the unknown parameters of this model. Using them, we improve color consistency of the input images or perform color transfer with any input image as the source. Our approach is insensitive to outliers in the pixel correspondences thereby precluding the need for complex pre-processing steps. We demonstrate high quality color consistency results on large photo collections of popular tourist landmarks and personal photo collections containing images of people.

1. Introduction

Nowadays, the growing popularity of photo sharing and social networks makes it easy to crowdsource photo collections of popular locations and social events. This has led to applications ranging from virtual tourism and navigation [35], image completion [19], colorization [7] and photo uncropping [32]. However, the color statistics of each image in the collection could differ due to different scene illumination at capture time or due to different non linear camera response functions [15, 25]. Such photometric inconsistencies cause visual artifacts in applications that require seamless alignment of multiple overlapping images.

Although modern image editing packages provide some color correction, and tone adjustment functionalities, these techniques usually require indirect user interaction [2, 20], or direct adjustment of color balance or manipulation of the tone curve. Consequently, these interactive techniques are too tedious for large image collections. On the other hand, individual color correction is likely to produce images with inconsistent colors across the whole collection. Recently, HaCohen et al. [17] proposed a method to optimize color consistency across an image collection with respect to a reference image that relies on recovering dense pixel correspondence across multiple images [16]. This method is computationally expensive and not ideal for processing large collections involving thousands of images.

In this paper, we present a new matrix factorization based approach to automatically optimize color consistency for multiple images using sparse correspondence obtained from multi-image sparse local feature matching. For rigid scenes, we leveraging structure from motion (SfM) although it is an optional step. We stack the aligned pixel intensities into a vector whose size equals the number of images. Such vectors are stacked into a matrix, one with many missing entries. This is the observation matrix that will be factorized. Under a simple color correction model, the logarithm of this matrix satisfies a rank two constraint under idealized conditions (perfect correspondences, no noise, constant illumination). The rank two matrix can be expressed as a sum of two rank one matrices – one that depends on the color correction parameters and another that depends on the albedos of the scene points associated with the sparse correspondences. The color correction parameters can be viewed as pseudo white balance and gamma correction parameters of the image. Here, **pseudo** indicates that the estimates do not necessarily coincide with the ground truth values.

Our method is based on the low-rank matrix factorization technique proposed in [4] that is robust to outliers. Robustness is key to the success of our method since in real conditions, several factors – lighting change, shadows, saturated pixels, incorrect feature correspondences, etc. produce outliers that corrupts the low rank structure of the observation matrix. We also analyze ambiguities in the matrix factorization formulation and suggest ways to resolve them practically. The low rank matrix formulation and the application of the L_1-norm based robust factorization technique are the main contributions of our work.

Unlike the previous quadratic optimization problem formulation [17] which relies on dense and accurate correspon-

*Part of this work was done while the first and second author were in KAIST. This work was supported in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No.2010-0028680).

Figure 1. First row: A selection from a large Internet photo collection (1500 images) of the Trevi fountain (images captured with different cameras on different days under different lighting). Second row: Our automatic technique performs consistent color correction on large image sets. Third row: The color of a target image (with a red boundary) can then be efficiently transferred to all the other images.

dence, our technique only requires sparse correspondence. Further more, the inherent robustness of our method makes it less sensitive to outliers and removes the need for complex pre-processing. Our approach is computationally much more efficient than [17] and is practical for large collections exceeding a thousand images. It can also handle smaller sets of images with significant variation in scale, viewpoint, object pose and deformation, where the recovery of outlier-free dense correspondence is challenging [16].

We evaluate our technique on diverse datasets ranging from large image sets of tourist landmarks (Fig. 1 and Fig. 2), Internet images of celebrities as well as personal photo collections. We also demonstrate that color correction and color transfer achieved with our method improves the quality of image stitching, multi-view stereo, and image-based rendering when using crowdsourced photos.

2. Related Work

We now review existing methods for color correction categorized by the need or absence of user interaction.

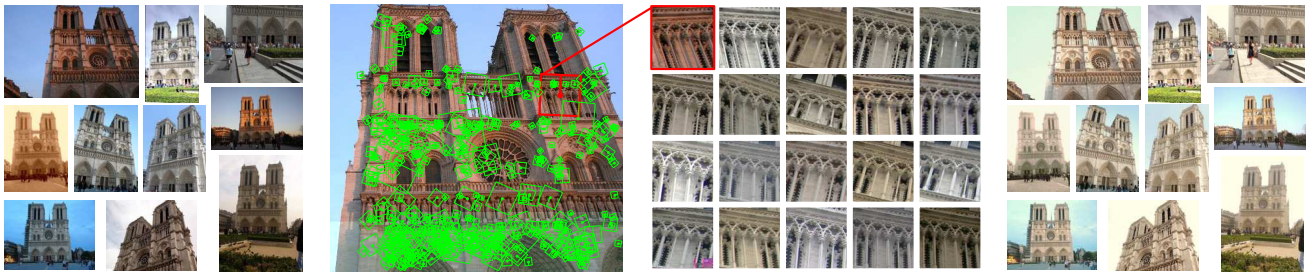
Single image methods. A popular approach for color correction [3] utilizes a reference image of a white object to adjust the white balance of other images captured by the same camera under similar illumination. Modern techniques [10, 11, 13, 14, 30] exploit statistical relationships between light and colors to estimate illumination of a scene for white balance correction. These methods are automatic, but they are designed for single images and cannot enforce consistent corrections on multiple overlapping images.

An interactive system used to locally edit tonal values in an image [27] performs local white balance correction for images with mixed illumination. Another approach proposed by Boyadzhiev et al. [2] involves a two light source mixture model that is used to correct spatially varying white balance with minimal user input [20]. However, interactive methods are impractical for very large image collections.

Batch methods. A few automatic color correction methods exist for large photo collections of rigid scenes. Garg et al. [12] observe that scene appearance often has low dimensionality and exploit that fact for color correction. Laffont et al. [24] estimate coherent intrinsic images and transfer localized illumination using the decomposed layers. Díaz et al. [9] performs batch radiometric calibration using the empirical prior on camera response functions [15]. Shi et al. [33] handles the effect of nonlinear camera response using a shape prior. Kim and Pollefeys [22] introduce a decoupled scheme for radiometric calibration and the vignetting correction. In contrast to [9, 12, 22, 24, 33], our method only requires sparse correspondence. Moreover, images of non-rigid scenes can be handled. For rigid scenes, we optionally use SfM for more accurate correspondence but neither surface normals nor dense 3D models are needed.

For more general scenes, the non-rigid dense correspondence (NRDC) algorithm proposed by HaCohen et al. [16] was used to optimize color consistency for image pairs using a linear color transform. Their work in [17] then targets photo collections by propagating colors of a reference image to other images. This involves estimating optimal parameters of three piecewise-quadratic splines which minimize color differences between all correspondences. However, their quadratic energy formulation is sensitive to errors in matching, and therefore rely on accurate and dense correspondences obtained using a computationally expensive algorithm such as NRDC. This makes their approach less suited for very large photo collections [9].

Instead of using high quality correspondences recovered by NRDC [17], our technique uses sparse correspondences and it is also less sensitive to erroneous correspondences due to the underlying robust optimization framework. According to [17], an accelerated implementation of NRDC took more than 6 hours to construct the underlying match graph for a set of 865 images on a MacBook Pro (2.3 Ghz Core i7 CPU and 8GM RAM). In contrast, for our TREVI FOUNTAIN (1500 images) dataset, feature matching and SfM in our implementation together took 50 minutes on a desktop PC with about 30 minutes for feature matching. For non-rigid scenes, the SfM stage is replaced by a much faster



(a) (b) (c) (d)

Figure 2. Overview: (a) Selected input images. (b) Keypoints extracted in an image. (c) Local feature descriptors are matched to obtain sets of aligned image patches from which the low-rank matrix is constructed. (d) We factorize this matrix using our robust technique and jointly estimate color correction parameters for each image. (d) The same set of images after color correction.

graph algorithm described in the paper.

Applications. Both geometric alignment of overlapping images as well as color and gamma correction is crucial for visual aesthetics in applications such as virtual tourism and navigation [1, 23, 34, 35], photo-realistic rendering [31], scene completion [19], image colorization [7, 28], image restoration [8], image montage [6], photobio [21] and photo uncrop [32]. Color correction prior to image alignment can further improve the alignment accuracy in these methods.

3. Color Correction Model

We adopt a global color correction model for reasons discussed in [17], namely robustness to alignment errors, ease of regularization and higher efficiency due to fewer unknown parameters. Our simple model is as follows:

$$I' = (cI)^{\gamma} \tag{1}$$

where I' is the input image, I is the desired image, c is a scale factor equivalent to the white balance function [20] and $(.)^{\gamma}$ is the non-linear gamma mapping. Equation (1) is independently solved for each color channel.

We assume that the surface reflectance of a scene point is constant across the images. Given m input images $\{I_i\}_{i=1}^{m}$, n 3D points $\{p_j\}_{j=1}^{n}$ and their 2D image projections $\{x_{ij}\}$, the intensity at a particular pixel x_{ij} in image I_i, is

$$I_i(x_{ij}) = (c_i a_j e_{ij})^{\gamma_i}, \tag{2}$$

where a_j is the constant albedo of the j-th 3D point and c_i and γ_i are the unknown global parameters for the i-th image. The per-pixel error term denoted as e_{ij} captures unmodeled color variation due to factors such as lighting and shading change that cannot be modeled with Eq. (1).

Taking logarithms on both sides of Eq. (2), we get:

$$\log(I_i(x_{ij})) = \gamma_i \log(c_i) + \gamma_i \log(a_j) + \gamma_i \log(e_{ij}). \tag{3}$$

Rewriting Eq. (3) in matrix form, by grouping image intensities by scene point into sparse column vectors of length m and stacking the n columns side by side, we get:

$$\mathbf{I} = \mathbf{C} + \mathbf{A} + \mathbf{E}. \tag{4}$$

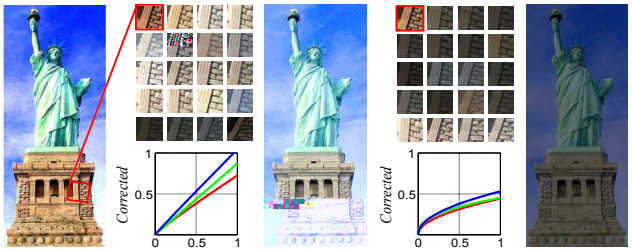
Here, n denotes the number of 3D points or equivalently the number of correspondence sets. $\mathbf{I} \in \mathbb{R}^{m \times n}$ is the observation matrix, where each entry $\mathbf{I}_{ij} = \log(I_i(x_{ij}))$. $\mathbf{C} \in \mathbb{R}^{m \times n}$ is the color coefficient matrix where $\mathbf{C}_{ij} = \gamma_{ij} \log c_{ij}$. $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the albedo matrix where $\mathbf{A}_{ij} = \gamma_{ij} \log a_{ij}$. Finally, $\mathbf{E} \in \mathbb{R}^{m \times n}$ is the residual matrix where $\mathbf{E}_{ij} = \gamma_{ij} \log e_{ij}$. Here, the row index i denotes the i-th image, and the column index j denotes the j-th 3D point.

Lemma 1. Rank of $\mathbf{C} + \mathbf{A} = 2$.

Proof. Since c and γ are global parameters for an image, $c_{i1} = c_{i2} = \cdots = c_{in}$, and $\gamma_{i1} = \gamma_{i2} = \cdots = \gamma_{in}$. Thus, each row of $\mathbf{C}$ is identical. Hence, $\mathbf{C}$ is a rank-1 matrix. Similarly, a_{ij} represents surface albedo of a scene point and $a_{1j} = a_{2j} = \cdots = a_{mj}$. Since each row of $\mathbf{A}$ is multiplied by the same value, γ_i, the rows of $\mathbf{A}$ are linearly dependent but have different linear coefficients γ_1/γ_i. Thus, $\mathbf{A}$ is also a rank-1 matrix. Further, since the linear dependence of $\mathbf{C}$ and $\mathbf{A}$ are independent of each other, the rank of $\mathbf{C} + \mathbf{A}$ is equal to 2. This concludes the proof. □

Assumptions. The matrix $\mathbf{C}$ should be viewed as a set of global image parameters for optimizing color consistency across the input images. In general, our estimate of $\mathbf{C}$ will not match the camera's true white balance and gamma settings. Similarly, our estimate of $\mathbf{A}$ is unlikely to match the true albedos. We now discuss our assumptions regarding unmodeled color variation caused by lighting change, etc.

Consider the situation where the images of a scene point are mostly captured in bright lighting. In this case, its albedo estimate is likely to be greater than the true value. In fact, the dominant bright illumination would be absorbed into the albedo term. The implicit assumption we make is that most 3D scene points have a somewhat dominant mode in their color distribution and that for most points, the estimated albedos will be consistent with the dominant color modes. This is true for example if most of the images were captured during daytime in bright lighting. Fig. 3 shows an example where the input images have two very different dominant color modes. In this case, the outputs for a specific image (see Fig. 3(a)) present in both sets will be differ-



(a) (b) (c)

Figure 3. Our correction result depends on the color distribution of the input image set. (a) One of the input images I. (b) The result when I is processed along with a specific set of 16 bright and 4 dark images. (c) A different result is produced when a different set of 19 image (16 dark and 4 bright images) are used. A few aligned patches and the estimated color correction curves are shown.

ent. This is because we aim at optimizing color consistency across the majority of images in the input set.

According to Lemma 1, if matrix $\mathbf{I}$ has rank greater than 2, then the matrix $\mathbf{E}$ must encode the per-element deviations from the rank 2 structure. Based on the assumption stated earlier, we expect most entries in $\mathbf{E}$ to be close to zero since most observations will have relatively small deviations from the dominant color. Large, non-zero entries in $\mathbf{E}$ are caused by outliers due to shadows, saturation, pixel mismatches etc. The occurrence of such entries will be sparse. Our experiments on diverse Internet image datasets reported later on in Sec. 6.2 validates these hypotheses and our assumptions about scene illumination changes.

Ambiguities in the Solution. The solution $\{\mathbf{C}, \mathbf{A}\}$ has a multiplicative ambiguity. If γ^* and c^* are the true values (assuming that we have the correct estimate of $\mathbf{C}$), then $\kappa\gamma^*$ and $\frac{1}{\kappa}\log(c^*)$ are also a correct decomposition of $\mathbf{C}$ for any arbitrary positive scale factor κ. Note that the multiplication by κ does not increase the rank of $\mathbf{C}$. Thus, it is impossible to recover γ^* and c^* simultaneously, without making assumptions on γ^* or c^*. A similar multiplicative ambiguity also exists for $\mathbf{A}$.

4. Matrix Factorization-based Formulation

In order to estimate γ, c, and a, we define two augmented matrices – $\mathbf{P} := [\mathbf{c}\odot\mathbf{g}, \mathbf{g}] \in \mathbb{R}^{m\times 2}$ is a $m\times 2$ matrix which concatenates two column vectors $\mathbf{c}\odot\mathbf{g}$ and $\mathbf{g}$, $\mathbf{c}\in\mathbb{R}^{m\times 1}$, $\mathbf{c}_i = \log c_i$, $\mathbf{g}\in\mathbb{R}^{m\times 1}$, $\mathbf{g}_i = \gamma_i$, $\odot$ denotes an element-wise multiplication operator, and $\mathbf{Q} := [\mathbf{1}, \mathbf{a}] \in \mathbb{R}^{n\times 2}$ is a $n\times 2$ matrix which concatenates two column vectors, $\mathbf{1}\in\mathbb{R}^{n\times 1}$ is a vector filled with 1, and $\mathbf{a}\in\mathbb{R}^{n\times 1}$, $\mathbf{a}_j = \log a_j$. By this definition, the augmented matrices satisfy $\mathbf{PQ}^\mathsf{T} = \mathbf{C}+\mathbf{A}$. We can solve $\mathbf{P}$ and $\mathbf{Q}$ by applying the factorization based low-rank matrix completion [4] method.

$$\mathbf{P}^*, \mathbf{Q}^* = \underset{\mathbf{P},\mathbf{Q}}{\operatorname{argmin}} \|\mathbf{W}\odot(\mathbf{I}-\mathbf{PQ}^\mathsf{T})\|_p + \frac{\lambda}{2}(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2), \tag{5}$$

where $\mathbf{W}$ is a binary indicator matrix of $\mathbf{E} = \mathbf{I}-\mathbf{PQ}^\mathsf{T}$, $\|\cdot\|_p$ denotes the L_p-norm, $\|\cdot\|_F^2$ denotes the Frobenius norm of a matrix, and λ is a parameter which controls the sparsity of the solution. We use the L_1-norm here (**i.e.** $p = 1$) to deal with outliers in $\mathbf{E}$. $\mathbf{W}_{ij} = 1$ if the j-th correspondence appears in the i-th image, and $\mathbf{W}_{ij} = 0$ otherwise.

The optimal solution of Eq. (5) still contains the multiplicative ambiguity. In order to obtain the correct solution, we introduce a new constraint on $\mathbf{Q}$:

$$\mathbf{P}^*, \mathbf{Q}^* = \underset{\mathbf{P},\mathbf{Q}}{\operatorname{argmin}} \|\mathbf{W}\odot(\mathbf{I}-\mathbf{PQ}^\mathsf{T})\|_1 + \frac{\lambda_1}{2}(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) + \frac{\lambda_2}{2}(\|\mathbf{Q}-\mathbf{Q}'\|_F^2), \tag{6}$$

where $\mathbf{Q}' := [\mathbf{1}, \mathbf{a}']$ and $\mathbf{a}'$ is an approximate solution of surface albedo which imposes regularization on $\mathbf{a}$. The multiplicative ambiguity in $\mathbf{g}$ and $\mathbf{c}$ is then resolved sequentially after obtaining the correct solution of $\mathbf{a}$.

Approximate Solution of $\mathbf{a}$. Without any prior information on $\mathbf{a}$, we use the same assumption as [28], that median intensities provide an approximate estimate of surface albedo if a scene is observed from multiple viewpoints under changing illuminations. We also encourage the albedo estimates to be spatially smooth. Thus, we have

$$\mathbf{a}' = \underset{a}{\operatorname{argmin}} \sum_i (a_i - a_{i,med})^2 + \sum_i \sum_{j\in\mathcal{N}_i} w_{i,j}(a_i - a_j)^2 \tag{7}$$

where $a_{i,med}$ is the median value of pixel intensities in logarithm domain across the images, the spatial weight $w_{i,j}$ is inversely proportional to the 2D Euclidean distance between i-th and j-th points in an image (if the correspondences are estimated from SfM, the 3D distance is used instead), and $\mathcal{N}_i$ is a set of local neighbor of i-th point.

Optimization Procedures. In order to solve Eq. (6) efficiently, we utilize the Augmented Lagrange Multiplier (ALM) method [26] and rewrite Eq. (6) as:

$$\underset{\mathbf{Z},\mathbf{P},\mathbf{Q},\mathbf{Y},\alpha}{\operatorname{argmin}} \|\mathbf{W}\odot(\mathbf{I}-\mathbf{Z})\|_1 + \frac{\lambda_1}{2}\left(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2\right) + \frac{\lambda_2}{2}\left(\|\mathbf{Q}-\mathbf{Q}'\|_F^2\right) + \langle\mathbf{Y}, \mathbf{Z}-\mathbf{PQ}^\mathsf{T}\rangle + \frac{\alpha}{2}\|\mathbf{Z}-\mathbf{PQ}^\mathsf{T}\|_F^2, \tag{8}$$

where $\mathbf{Z}$, $\mathbf{Y}$, and α are auxiliary variables. The optimization procedure of the ALM method [26] is summarized in Algorithm 1. First, it involves decomposing Eq. (8) into separate subproblems for $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{Z}$ which are solved iteratively. This is called the inner-loop. Next, using the estimates of $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{Z}$ in the current iteration, the values of $\mathbf{Y}$ and α are updated. Finally, the inner-loop repeats with updated values of $\mathbf{Y}$ and α until convergence. This is called the outer-loop. We now derive the solutions for each of the independent subproblems.

The sub-problems involving finding optimal values of $\mathbf{P}$ and $\mathbf{Q}$ (Eq. (8)) can be solved in closed form by setting the

(a) (b) (c) (d) (e) (f) (g)

Figure 4. Color transfer example: (a–b) Image pair taken from [16]. (c) Our color transfer result using nine SIFT correspondences. (d) the nine associated patches. (e) The yellow dot indicates the center pixel whereas orange dots indicate additional pixels sampled using augmentation. (f) Our result with augmented correspondences. Note the significant improvement over our result shown in (c). (g) The result from [16] is similar to our result shown in (f) but was obtained using dense correspondences.

Algorithm 1 Factorization based low-rank matrix completion

Input : $\mathbf{I} \in \mathbb{R}^{m\times n}$, $\lambda_1 = 1/\sqrt{\min(m,n)}$, and $\lambda_2 = 10\lambda_1$. Initialize $\mathbf{P}^0 = [\mathbf{0}, \mathbf{1}]$, $\mathbf{Q}^0 = [\mathbf{1}, \mathbf{a}']$, and $\mathbf{Z}^0$ as a random matrix sampled from a unit normal distribution, $\mathbf{Y} = 0$, $\alpha^0 = 10^{-3}$.
while not converged **do**
  while not converged **do**
    Update $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{Z}$ via Eq. (9, 10, and 12).
  end while
  Update $\mathbf{Y}$ via Eq. (13).
  $\alpha = \min(1.5\alpha, 10^{20})$.
end while
Output : $(\mathbf{P}^*, \mathbf{Q}^*, \mathbf{E}^* = \mathbf{I} - \mathbf{P}^*\mathbf{Q}^*)$.

first order derivatives of Eq. (8) with respect to $\mathbf{P}$ and $\mathbf{Q}$ respectively to zero. We obtain the following expressions.

$$\mathbf{P} = (\alpha\mathbf{Z} + \mathbf{Y})\mathbf{Q}(\alpha\mathbf{Q}^\mathsf{T}\mathbf{Q} + \lambda_1\mathbb{I}_{2\times 2})^{-1}, \quad (9)$$

$$\mathbf{Q} = ((\alpha\mathbf{Z} + \mathbf{Y})^\mathsf{T}\mathbf{P} + \lambda_2\mathbf{Q}')(\alpha\mathbf{P}^\mathsf{T}\mathbf{P} + (\lambda_1 + \lambda_2)\mathbb{I}_{2\times 2})^{-1}; \quad (10)$$

where $\mathbb{I}_{2\times 2}$ is a 2×2 identity matrix. This derivation is possible because the expression in Eq. (8) is quadratic in $\mathbf{P}$ and $\mathbf{Q}$. In our implementation, we follow the suggested parameter value by [38] and set $\lambda_1 = 1/\sqrt{\min(m,n)}$, and we set $\lambda_2 = 10\lambda_1$. Also, since the first column of $\mathbf{Q}$ should be equal to $\mathbf{1}$, we substitute the first column of $\mathbf{Q}$ by $\mathbf{1}$ after each iteration. This makes the inner loop converge faster.

After substituting the values of $\mathbf{P}$ and $\mathbf{Q}$, we can rewrite Eq. (8) as a subproblem of $\mathbf{Z}$, which is as follows.

$$\underset{\mathbf{Z}}{\operatorname{argmin}} \|\mathbf{W} \odot (\mathbf{I} - \mathbf{Z})\|_1 + \frac{\alpha}{2}\|\mathbf{Z} - \left(\mathbf{P}\mathbf{Q}^\mathsf{T} - \frac{\mathbf{Y}}{\alpha}\right)\|_F^2. \quad (11)$$

Eq. (11) has the following closed form solution [4].

$$\mathbf{Z} = \mathbf{W} \odot \left(\mathbf{O} - \mathcal{S}_{\frac{1}{\alpha}}(\mathbf{I} - \mathbf{P}\mathbf{Q}^\mathsf{T} + \frac{\mathbf{Y}}{\alpha})\right) + \overline{\mathbf{W}} \odot (\mathbf{P}\mathbf{Q}^\mathsf{T} - \frac{\mathbf{Y}}{\alpha}), \quad (12)$$

where $\mathcal{S}_d(b) = \max(0, b - d)$ denotes an element-wise shrinkage operator, and $\overline{\mathbf{W}}$ denotes the complement of $\mathbf{W}$. We repeat the inner-loop, which solves Eq. (9), Eq. (10) and Eq. (12) sequentially, until the decrease in residual error e of Eq. (8) is very small. We stop iterating when $|e_t - e_{t-1}| < 10^{-12} \times e_{t-1}$, where e_t and e_{t-1} are the residuals after the t-th and $(t-1)$-th iterations, respectively. After optimizing $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{Z}$, $\mathbf{Y}$ is updated as follows.

$$\mathbf{Y} = \mathbf{Y} + \alpha(\mathbf{Z} - \mathbf{P}\mathbf{Q}^\mathsf{T}), \quad (13)$$

where α is reset to $\min(1.5\alpha, 10^{20})$. Using the updated value of $\mathbf{Y}$ and α, we repeat the inner-loop if $\|\mathbf{I} - \mathbf{P}\mathbf{Q}\|_F^2 > 10^{-9} \times \|\mathbf{I}\|_F$.

After the optimization procedure converges, we can retrieve estimates, $\mathbf{a}^*$ from $\mathbf{Q}^* := [\mathbf{1}, \mathbf{a}^*]$ and $\mathbf{g}^*$ from $\mathbf{P} := [\mathbf{c}^* \odot \mathbf{g}^*, \mathbf{g}^*]$. Then $\mathbf{c}^*$ is also obtained from $\mathbf{c}^* \odot \mathbf{g}^*$ by dividing by $\mathbf{g}^*$ obtained in the previous step. By applying the inverse functions associated with the estimated $\mathbf{g}^*$ and $\mathbf{c}^*$ on the input images, we can achieve color consistency across the entire set of images.

Outlier detection. The residual errors in $\mathbf{E}$ in Eq. (4) can be used to detect outlier observations that do not follow our model. This can be due to shadows, saturated pixels or erroneous correspondences. For such observations, applying our color correction leaves a large residual. The top 10% pixels in terms of larger residual error are classified as outliers. Figure 5 shows an example of outliers in an image from the TREVI FOUNTAIN dataset.

(a) (b) (c)

Figure 5. (a) An image from TREVI FOUNTAIN. (b) Color and gamma corrected result. (c) Inliers to our model are marked in blue whereas outlier pixels (mostly in shadows) are marked in red.

5. Implementation Details

We now describe the steps for recovering sparse correspondence and construction of the observation matrix $\mathbf{I}$.

SfM pre-processing. For landmarks or in general rigid scenes, we use scale invariant feature matching [29, 37] and structure from motion (SfM) [35] to obtain multi-image correspondences. Although the estimated sparse 3D reconstruction is not used, the SfM pipeline filters outliers effectively and retains globally optimized correspondences geometrically consistent over many images.

Non-rigid scenes. For such input images, we extract SIFT features [29] and run nearest neighbor (NN) descriptor matching on image pairs. For each pair, we do the matching both ways and retain the matches for the reciprocal nearest neighbors[1]. Next, we construct an undirected match graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents all the SIFT features, and $\mathcal{E}$ denotes all the pairwise matches. Within this graph, we then find the maximal cliques of size three and above. The maximal cliques are computed using a variant of the Bron-Kerbosch algorithm [36] and these provide the correspondences used to construct the observation matrix.

Building the matrix I. Using the scale and orientation of matched keypoints, we resample patches 30×30 pixels wide from the associated input images which are often well aligned (see Figures 2(b) and 2(c)). We sample intensities from these patches to construct $\mathbf{I}$. For reliable correspondences such as those obtained from SfM, we select one observation from each patch in a feature track to construct a new row for the matrix $\mathbf{I}$. Specifically, we use the median intensity of each patch; this provides some robustness to misalignments, occlusion, shadows and JPEG artifacts.

Data Augmentation. For scenes with sparser feature matches there are too few observations to estimate all the unknowns, as noted in [16]. Fig. 4 shows a two image example with only nine feature matches[2]. Here, single pixel sampling produces unsatisfactory results (Fig. 4(c)). We address this issue by augmenting the observations using additional pixels sampled from the aligned patches using a predefined sampling pattern (Fig. 4(e)). This assumes reasonable patch alignment and local surface smoothness but is quite effective since the subsequent optimization step is robust to outliers. The improved result is shown in Fig. 4(f).

6. Experimental Results

We first analyze the robustness of our method on synthetic data. Next, selected color correction results are presented. A detailed comparison with [17] is reported highlighting the higher efficiency and robustness of our method followed by an analysis of running times. The supplementary material has additional results and shows improved results for image stitching, multi-view stereo and image-based rendering made possible by our method.

6.1. Robustness Analysis

We conducted synthetic experiments with sparse observation matrices generated according to our model (Eq. (2)) with image intensities scaled to the range $[0, 1]$. We sample $a_j \sim \mathcal{U}(0, 1)$, $c_i \sim \mathcal{U}(0.5, 1.5)$, and $\gamma_i \sim \mathcal{U}(0.5, 4)$, where $\mathcal{U}(a, b)$ denotes an uniform distribution over $[a, b]$. These matrices have size $\mathbb{R}^{m \times n}$ with the number of points n fixed (=1000), the number of images m, varying from 50 to 500 and the fraction of missing entries fixed (=95%). To simulate per-element deviations from the rank 2 structure, we sample $e_{ij} \sim \mathcal{N}(1, \sigma^2)$ with varying σ where $\mathcal{N}(\mu, \sigma^2)$ denotes normal distribution with mean μ and variance σ^2. Finally, we sample random outliers in e_{ij} from $\mathcal{U}(0.5, 1.5)$ given a target outlier ratio. To evaluate the importance of the robust L_1-norm in Eq. (6), we also test a variant of our algorithm that instead uses the L_2-norm in Eq. (6).

		σ used for $e_{ij} \sim \mathcal{N}(1, \sigma^2)$ and outlier percentage of e_{ij}					
		$\sigma = 0.01$		$\sigma = 0.03$		$\sigma = 0.05$	
		10%	20%	10%	20%	10%	20%
# of Imgs	50	2.17 / 4.73	3.90 / 6.81	3.51 / 5.65	3.79 / 5.81	4.42 / 6.17	4.63 / 5.65
	100	0.93 / 2.91	1.55 / 4.46	2.02 / 3.41	2.40 / 4.32	2.39 / 3.44	2.93 / 4.45
	300	0.33 / 2.19	0.34 / 3.00	0.96 / 2.38	1.09 / 3.02	1.34 / 2.31	1.52 / 2.95
	500	0.26 / 1.82	0.29 / 2.56	0.71 / 1.82	0.89 / 2.78	1.26 / 2.00	1.30 / 2.58

*Residual errors when applying L_1 / L_2-norm based approaches. Unit: $(\times 10^{-2})$

Table 1. Comparisons between L_1 and L_2-norm in the first term of Eq. (6). Using L_1-norm consistently gives more accurate results than using L_2-norm. The image intensities are normalized to [0,1].

Table 1 summarizes the mean residual errors ($\|\mathbf{W} \odot (\mathbf{G} - \mathbf{P}^*\mathbf{Q}^{*\mathsf{T}})\|_1$) for various runs where the outlier fraction was set to 10% and 20% for three different settings of σ. Here $\mathbf{G}$ denotes ground truth. The mean residual errors are always less than 0.05 across all runs. The L_1-norm is consistently more robust than the L_2-norm especially when σ is small and the improvement margin increases when the outlier fraction increases. In general, accuracy increases when more images are used. Larger values of σ as expected causes moderate increase in error but the L_1-norm still performs the best.

6.2. Batch Color Correction

We first show selected color correction results on subsets of photos from five collections of tourist landmarks in Figures 1 and 6 – NOTRE DAME (715 images) [35], TREVI FOUNTAIN (1500 images), ST. BASIL CATHEDRAL (1700 images), STATUE OF LIBERTY (2362 images), and DRESDEN FRAUENKIRCHE (2025 images)[3]. The latter four also contain Flickr images downloaded as part of the public dataset (`landmark3d.codeplex.com`) from [18]. We used our own state of the art SfM implementation.

We have confirmed the assumptions made in our low-rank model by analyzing $\mathbf{E}$, the residual matrix for the five datasets. Figure 7 shows the error distributions on these datasets. Only 0.75%, 0.96%, 2.88%, 0.31% and 1.89% of observations have residual error greater than 0.2 respectively (where image intensities lie in $[0, 1]$).

Figures 8 and 9 show results on the WEDDING [17], and ICE SKATER datasets. The ICE SKATER dataset contains 36 images obtained using Google Image Search and

[1] a, b are reciprocal nearest neighbors, when a's NN is b and vice versa.

[2] In this case, we use the intensity of the image in (a) as the approximate solution of $\mathbf{a}'$. Hence, the correction is with respect to the image in (a).

[3] The numbers in brackets are the input sizes for SfM.

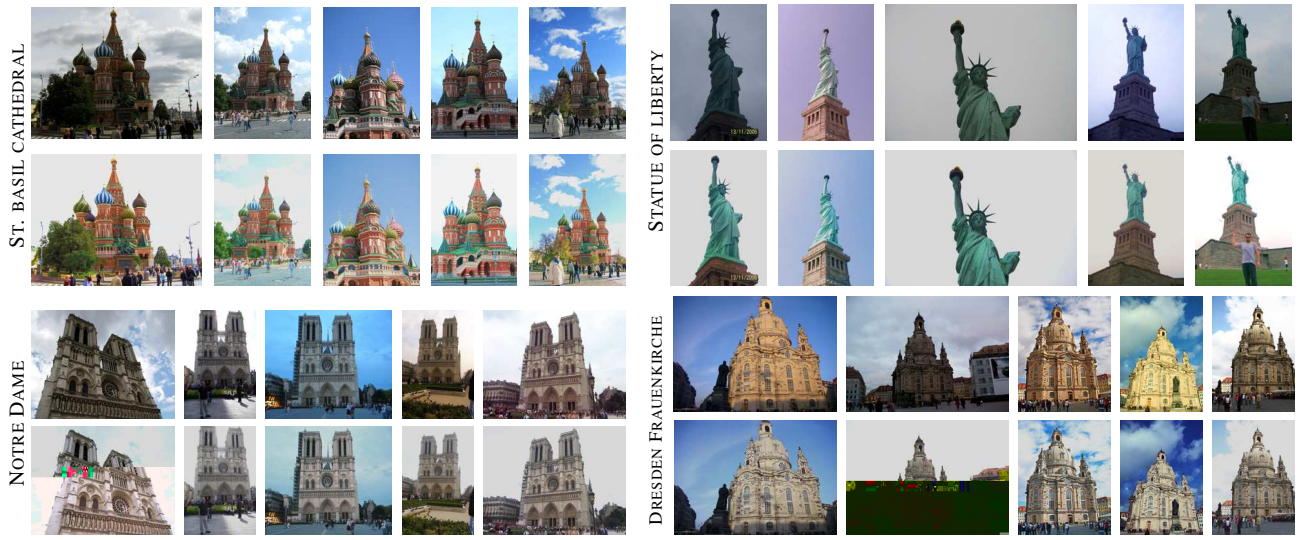


Figure 6. Results on four other landmark photo collections. In each case, five images are shown. The upper row shows the original images, the lower row shows our auto correction results. More results are shown in the supplementary material.

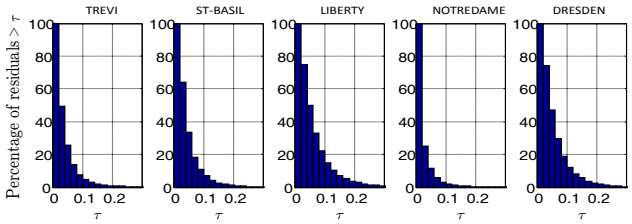


Figure 7. Residual error ($\|\mathbf{W} \odot (\mathbf{I} - \mathbf{P}^*\mathbf{Q}^{*\mathsf{T}})\|_1$) distribution on five real datasets. The percentages of elements in the residual vector with magnitude greater than the threshold τ are plotted.

contain noticeable variations in colors, contrast, human pose and appearance. On these two datasets, we used the implementation of SIFT features in VLFeat (www.vlfeat.org). An additional example, the BUSH sequence can be found in our supplementary material.

Inspecting the results qualitatively, we see that despite the huge input variability, darker images are brightened and the images with unusual colors are successfully corrected to be consistent with the other images. Once each image in the input set is corrected, we can treat those images as if they were taken from a single virtual camera. Figure 1 shows a color transfer example on the TREVI FOUNTAIN dataset. The corrected images were transformed with the inverse camera function of the selected reference image.

Comparison with [17]. We compare our method with that of HaCohen **et al**. [17] on a small subset of the WEDDING dataset (9 images) published on their project website and the ICE SKATER dataset (36 images). Here, we use our own re-implementation of [17] based on **quadprog**, a Matlab quadratic programming package. We test two baselines by running their method using both NRDC [16] correspondences as well as our sparse correspondences as input. Both WEDDING and ICE SKATER are small datasets and the huge variation in image scale, composition and the subject's pose makes it difficult to obtain high quality results using NRDC [16]. Since the sparse SIFT correspondences are fewer, we perform data augmentation on it (30× samples) as described earlier in the paper.

The results shown in Fig. 8 and Fig. 9 demonstrate that our approach is quite effective and performs better than both baselines. In contrast, the baseline method [17] performs moderately with NRDC correspondences (Fig. 8 2nd row) but shows a lack of color consistency when used with sparse correspondences (Fig. 8 3rd row). The cost function based on the L_2 norm used in [17] appears to be sensitive to outliers producing inconsistent colors even for the most similar images (**e.g**. the highlighted columns in Fig. 8 and Fig. 9). Increasing the weight of the regularization term in their method [17] also tends to produce darker images as their energy function penalizes pairwise intensity differences and hence can favor a color transform function that makes the image intensities darker.

Running Times. For TREVI FOUNTAIN (1500 images), SfM reconstructs 1467 cameras. When we test our factorization method on these images, the observation matrix has size 1467×52092. Our Matlab implementation took 56 minutes on a desktop PC with Intel i7-4790 CPU @ 3.6 GHz and 16 GB RAM. For an image-based rendering application described in the supplementary material, we selected a subset of 390 images. This time the 390×44827 matrix was factorized in about 18 minutes. The results on the common 390 images are almost identical. The timings for ICE SKATER (36 images) were 47 seconds for feature matching, 0.4 seconds for finding maximal cliques and 153 seconds for factorizing the matrix built from 146K correspondences (with 30× augmented samples). The SVD inside each inner

Figure 8. Comparison with [17] on the WEDDING dataset (9 images). (Row 1): Input images. (Row 2): Results from HaCohen **et al**.'s technique [17] using NRDC [16] and (Row 3) using sparse correspondences (same as the input to our method). (Row 4): Our results. The overall color consistency of our results are noticeably higher even though sparse correspondences were used.

Figure 9. ICE SKATER dataset (36 images): (Row 1) Selection of nine input images. (Row 2) Results of HaCohen **et al**. [17] using dense correspondences [16]. (Row 3): Results obtained using our method where significant color variations are consistently corrected.

loop iteration is the main computational bottleneck and can be sped up using fast singular value thresholding [5].

Limitations. Our approach has the same limitations as that of HaCohen et al. [17]. It may be ineffective when the input photos have low overlap. Also, our method may be less effective for certain input photos which have drastic lighting changes such as with daytime and night photos. Nevertheless the problem is somewhat mitigated by our ability to handle large photo collections. For rigid scenes, for surfaces always under shadow, the estimated albedo tends to be darker. This is because our estimation of white balance and gamma coefficients is biased by the initial albedo estimates and we cannot guarantee that the estimated parameters will always be accurate with respect to ground truth camera parameters. Finally, when brightening a dark image using the estimated values, brighter pixels may become saturated.

7. Conclusion

We have presented a novel and practical approach to optimize color consistency of a photo collection. Our key contribution is the novel rank-2 formulation of the problem and the proposed robust matrix factorization-based technique. Our robust formulation alleviates the need for dense correspondences as required by [17]. It is shown to be effective on Internet photo collections of tourist landmarks, celebrities as well as personal photo collections. In the future, we plan to conduct a user study to evaluate our method on more diverse photo collections.

References

[1] A. Arpa, L. Ballan, R. Sukthankar, G. Taubin, M. Pollefeys, and R. Raskar. Crowdcam: Instantaneous navigation of crowd images using angled graph. In **3DTV-Conference, 2013 International Conference on**, pages 422–429. IEEE, 2013.

[2] I. Boyadzhiev, K. Bala, S. Paris, and F. Durand. User-guided white balance for mixed lighting conditions. **ACM Transactions on Graphics (TOG)**, 31(6), 2012.

[3] G. Buchsbaum. A spatial processor model for object colour perception. **Journal of The Franklin Institute**, 310(1):1–26, 1980.

[4] R. Cabral, F. D. la Torre, J. P. Costeira, and A. Bernardino. Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. In **Proceedings of International Conference on Computer Vision (ICCV)**, 2013.

[5] J.-F. Cai and S. Osher. Fast singular value thresholding without singular value decomposition. **Methods and Applications of Analysis**, 20(4):335–352, 2013.

[6] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: internet image montage. **ACM Transactions on Graphics (TOG)**, 28(5), 2009.

[7] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin. Semantic colorization with internet images. **ACM Transactions on Graphics (TOG)**, 30(6), 2011.

[8] K. Dale, M. K. Johnson, K. Sunkavalli, W. Matusik, and H. Pfister. Image restoration using online photo collections. In **Proceedings of International Conference on Computer Vision (ICCV)**, pages 2217–2224, 2009.

[9] M. Díaz and P. Sturm. Radiometric calibration using photo collections. In **Proceedings of IEEE International Conference on Computational Photography (ICCP)**, 2011.

[10] G. D. Finlayson, S. D. Hordley, and P. M. Hubel. Color by correlation: A simple, unifying framework for color constancy. 23(11):1209–1221, 2001.

[11] D. Forsyth. A novel algorithm for color constancy. **International Journal on Computer Vision (IJCV)**, 5:5–36, 1990.

[12] R. Garg, H. Du, S. Seitz, and N. Snavely. The dimensionality of scene appearance. In **Proceedings of International Conference on Computer Vision (ICCV)**, 2009.

[13] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In **Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 2008.

[14] A. Gijsenij and T. Gevers. Color constancy using natural image statistics. In **Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 2007.

[15] M. Grossberg and S. Nayar. Modeling the space of camera response functions. **IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)**, 26(10), 2004.

[16] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. **ACM Transactions on Graphics (TOG)**, 30(4):70:1–9, 2011.

[17] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Optimizing color consistency in photo collections. **ACM Transactions on Graphics (TOG)**, 32(4), 2013.

[18] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, and F. Wu. 3d visual phrases for landmark recognition. In **CVPR**, pages 3594–3601, June 2012.

[19] J. Hays and A. A. Efros. Scene completion using millions of photographs. **ACM Transactions on Graphics (TOG)**, 26(3), 2007.

[20] E. Hsu, T. Mertens, S. Paris, S. Avidan, and F. Durand. Light mixture estimation for spatially varying white balance. **ACM Transactions on Graphics (TOG)**, 27(3), 2008.

[21] I. Kemelmacher-Shlizerman, E. Shechtman, R. Garg, and S. Seitz. Exploring photobios. **ACM Transactions on Graphics (TOG)**, 30(4), 2011.

[22] S. J. Kim and M. Pollefeys. Robust radiometric calibration and vignetting correction. **IEEE Trans. Pattern Anal. Mach. Intell.**, 30(4):562–576, Apr. 2008.

[23] A. Kushal, B. Self, Y. Furukawa, D. Gallup, C. Hernandez, B. Curless, and S. M. Seitz. Photo tours. In **3DImPVT**, 2012.

[24] P.-Y. Laffont, A. Bousseau, S. Paris, F. Durand, and G. Drettakis. Coherent intrinsic images from photo collections. **ACM Transactions on Graphics (SIGGRAPH Asia)**, 31, 2012.

[25] H. Lin, S. Kim, S. Susstrunk, and M. S. Brown. Revisiting Radiometric Calibration for Color Computer Vision. In **Proceedings of International Conference on Computer Vision (ICCV)**. IEEE, 2011.

[26] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. In **Mathematical Programming**, 2010.

[27] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski. Interactive local adjustment of tonal values. **ACM Transactions on Graphics (TOG)**, 25(3):646–653, 2006.

[28] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng. Intrinsic colorization. **ACM Transactions on Graphics (SIGGRAPH Asia 2008 issue)**, 27(5):152:1–152:9, December 2008.

[29] D. G. Lowe. Distinctive image features from scale-invariant keypoints. **International Journal on Computer Vision (IJCV)**, 60(2):91–110, 2004.

[30] C. Rosenberg, T. Minka, and A. Ladsariya. Bayesian color constancy with non-gaussian models. In **Annual Conference on Neural Information Processing Systems (NIPS)**, 2004.

[31] Q. Shan, R. Adams, B. Curless, Y. Furukawa, and S. M. Seitz. The visual turing test for scene reconstruction. In **Proceedings of International Conference on 3D Vision (3DV)**, 2013.

[32] Q. Shan, B. Curless, Y. Furukawa, C. Hernandex, and S. M. Seitz. Photo uncrop. In **Proceedings of European Conference on Computer Vision (ECCV)**, 2014.

[33] B. Shi, K. Inose, Y. Matsushita, P. Tan, S.-K. Yeung, and K. Ikeuchi. Photometric stereo using internet images. In **Proceedings of International Conference on 3D Vision (3DV)**, 2014.

[34] N. Snavely, R. Garg, S. M. Seitz, and R. Szeliski. Finding paths through the world's photos. **ACM Transactions on Graphics (TOG)**, 27(3):11–21, 2008.

[35] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In **Proceedings of ACM SIGGRAPH**, pages 835–846. ACM Press, 2006.

[36] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. A new algorithm for generating all the maximal independent sets. **SIAM Journal of Computing**, 6:505–517, 1977.

[37] S. Winder, G. Hua, and M. Brown. Picking the best daisy. **CVPR**, 0:178–185, 2009.

[38] J. Wright, A. Ganesh, S. Rao, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In **Annual Conference on Neural Information Processing Systems (NIPS)**, 2009.