

## Efficient global color correction for large-scale multiple-view images in three-dimensional reconstruction

Junxing Yang<sup>a</sup>, Lulu Liu<sup>a</sup>, Jiabin Xu<sup>a</sup>, Yi Wang<sup>b</sup>, Fei Deng<sup>a,\*</sup>

<sup>a</sup> School of Geodesy and Geomatics, Wuhan University, Wuhan, China

<sup>b</sup> China Railway Eryuan Engineering Group CO. LTD, Chengdu, China



### ARTICLE INFO

**Keywords:**

Multi-view images  
Color consistency  
Quadratic splines  
Hierarchical optimization

### ABSTRACT

Consistent global color correction across multiple-view images in three-dimensional (3D) reconstruction is an important and challenging problem. The present work addresses this issue by proposing a novel global color correction method for multi-view images based on a spline curve remapping function. In contrast to existing methods, we obtain a series of optimal functions by minimizing the variance in the color values of all observations of every sparse point generated by the Structure from Motion (SfM) technique. We also find that adding only simple constraints to the spline is required to prevent the loss of image contrast and gradient information. The robustness of the proposed method is ensured by the adoption of strong geometric constraints between multi-view images. Finally, the applicability of the method to large-scale multiple-view images is facilitated by proposing a parallelizable hierarchical image color correction strategy based on a tree structure. The performance of the proposed method is compared with the performances of existing state-of-the-art methods when applied to several challenging datasets. The results indicate that the notable flexibility of the spline curve, along with the proposed optimization process and hierarchical strategy, not only enable the proposed method to perform well with challenging datasets, but also provide high computational efficiency when working with large-scale image sets.

### 1. Introduction

Recent decades have witnessed significant advancement in image-based three-dimensional (3D) reconstruction, where a set of images depicting the same scene are captured from different viewpoints and used to reconstruct 3D models. The images can recover not only the 3D structural information but textural information, from which a realistic and natural model of the real world can be derived. Images captured in the multiple-view image acquisition process may be taken at different times, using different cameras under different weather conditions, which is particularly useful for large-scale scenes. The inevitable variations in the conditions of image capture will result in color discrepancies between images. As such, the original images cannot be used directly to perform texture mapping without generating a noticeable color discrepancy in the textured model, which can greatly affect the visual experience. Therefore, the color discrepancies between multi-view images must be diminished or eliminated to render consistent and harmonious 3D models. Hence, color correction methods play an essential role in the 3D reconstruction of multi-view images.

Numerous color correction algorithms for multi-view images have been proposed (HaCohen et al., 2013; Kuse and Jaiswal, 2015; Park et al., 2016; Shen et al., 2017). These methods use the same idea, which is to find the shared information between images, such as point features and line features, and calculate a global transformation function for each image to minimize the differences in this shared information. The key is to select a flexible enough transformation model and reliable shared information. In addition, the number of aerial images is usually huge, and improving the efficiency of color correction is also important. The linear model (Shen et al., 2017; Eden et al., 2006; Moulon et al., 2013) and gamma model (Park et al., 2016) are often used for color correction because fewer parameters can improve algorithmic efficiency. These two models are not flexible enough for challenging datasets. The authors of Ref. (HaCohen et al., 2013) uses spline function curve as transformation function, which has more flexibility. But at the same time there are more unknowns. And using Non-Rigid Dense Correspondence (NRDC) as the correspondence greatly reduces the efficiency of the algorithm, making it unsuitable for large-scale aerial images. Color transfer (Reinhard et al., 2001; Xiao and Ma, 2009; Su et al., 2014) is

\* Corresponding author.

E-mail address: [fdeng@sgg.whu.edu.cn](mailto:fdeng@sgg.whu.edu.cn) (F. Deng).

another tool for color correction. This method can transfer the color style of a source image to a target image, keeping their colors consistent. A common method is to use color transformation so that the target and source image have the same mean and variance. However, such methods, which only involve two images, can hardly be applied to large scale image sets. Color optimization directly on the mesh texture is an effective method (Kuse and Jaiswal, 2015; Waechter et al., 2014) to eliminate color discrepancies between adjacent texture blocks and improve a model's appearance. However, this requires additional mesh data, and the performance depends on the accuracy of texture mapping in 3D reconstruction.

Current color-correction methods calculate the energy cost with a pair of images as a unit and a graphical style, are computationally expensive, and are unsuitable for challenging and large-scale datasets. Inspired by previous work (HaCohen et al., 2013) and bundle adjustment (BA), we propose a method to improve the efficiency and effectiveness of correcting the color consistency of large-scale multi-view images. We apply SfM to obtain sparse points as the shared content, and minimize the variance of color values of all observations of every sparse point with several simple but powerful constraints to obtain a series of global quadratic spline transformation functions. The proposed approach is more efficient and elegant than methods that take each pair of overlapped images as a unit (HaCohen et al., 2013; Moulou et al., 2013). The strong constraint relationship between multi-view images makes the matching process reliable, and it guarantees the robustness of our algorithm even without removing noise. The quadratic spline is sufficiently flexible to compensate for a variety of common appearance differences, such as linear curves, gamma curves, and other standard global functions. To preserve contrast and gradient information, it suffices to add some simple constraints to the gradient of the spline curve. Finally, we propose a hierarchical parallel color-correction strategy based on the m-ray tree structure to process large-scale images. Our algorithm has better efficiency and effectiveness compared to current representative algorithms when applied to various datasets.

The rest of the article is structured as follows. An overview of related work is provided in Section 2. Section 3 outlines our proposed color-correction algorithm. Section 4 compares the performance of the proposed method and existing methods when applied to several challenging datasets. We present a summary and draw conclusions in Section 5.

## 2. Related work

We discuss three types of color correction methods: color transfer between two images, texture optimization, and color correction across images.

### 2.1. Color transfer

Ensuring color consistency between multi-view images is similar to color transfer, and can be summarized as the same problem of adjusting the tone of two or more images to obtain consistency. Numerous color-transfer techniques have been developed, which can be classified as parametric or nonparametric, and global or local. The authors of Ref. (Reinhard et al., 2001) proposed a technique to convert the color style of a target image to that of a reference image based on mean values and variances. Images are adjusted in the  $l\alpha\beta$  color space to eliminate the correlation between the red, green, and blue (RGB) channels. The authors of Ref. (Reinhard et al., 2001) solved the problem of regional color transfer between two natural images through probabilistic segmentation with an expectation maximization (EM) scheme to apply spatial and color smoothness to infer natural connectivity between pixels. The authors of Ref. (Xiao and Ma, 2009) proposed a color transfer algorithm based on two scene aspects, including details and color. The color-transfer problem was treated as an optimization problem and solved using a two-step method of histogram matching and gradient preserving optimization. Various metrics were proposed to objectively evaluate the

performance of example-based color-transfer algorithms. The authors of Ref. (Su et al., 2014) proposed a color-transfer method that obtained relatively competitive grain-free and detail-preservation effects using gradient perception decomposition, color mapping, and detail enhancement. The authors of Ref. (Su et al., 2014) followed the same strategy to suppress corruption artifacts. However, the retained structure cannot be guaranteed to be consistent in quantity with the original structure due to the use of gradient enhancement. The authors of Ref. (Vallet and Lelégard, 2013) performed partial iterations to symmetrize the nonparametric color correction to simultaneously adjust two images without preserving one as a reference image. Because it only symmetrizes the color correction of an image pair, it must iterate the process for multi-view image color correction. The authors of Ref. (Hwang et al., 2014) proposed a scattered point interpolation scheme based on a moving least squares method, enhanced by a probability model of color transfer in a 3D color space to address misalignment and noise. The authors of Ref. (Nguyen et al., 2014) eliminated the color bias caused by different lighting in source and target images by applying a white balance step on both images, thereby transforming the luminance values of the target image using gradient preserving matching (Xiao and Ma, 2009). Color transfer methods based on machine learning have obtained excellent results in conjunction with prior knowledge (Wang et al., 2011; He et al., 2017; Luan et al., 2017). However, color transfer techniques are inappropriate for the problem of multi-view color correction in 3D reconstruction because they involve only two images. However, these methods have great reference value for processing multi-view images. Additional details regarding color transfer techniques and comparisons of methods can be found elsewhere (Xu and Mulligan, 2010; Faridul et al., 2014).

### 2.2. Texture optimization

The purpose of color correction for multi-view images is to ensure that the texture of the model has color consistency and to improve the model's appearance. Therefore, it is a good idea to directly perform color correction on textures of the model (Kuse and Jaiswal, 2015; Waechter et al., 2014; Lempitsky and Ivanov, 2007; Pan and Taubin, 2015). The authors of Ref. (Kuse and Jaiswal, 2015) proposed a 3D reconstruction method for multi-view images by constructing a graph model of 3D points generated by dense matching, optimizing the color of the 3D points, and re-colorizing the images at the re-projected coordinates of the points using the obtained optimal colors. The authors of Ref. (Waechter et al., 2014; Lempitsky and Ivanov, 2007) proposed an algorithm to globally minimize the color consistency on the textures. Local poisson blending was applied to further adjust the seam visibility. The algorithms have been integrated into the mvs-texturing<sup>1</sup> open-source library. This method relies on the accuracy of texture mapping because it employs the average of the neighboring pixels of two images as the boundary value to optimize the energy function. However, it can be counterproductive in the absence of accurate texture mapping and when using this algorithm to process multiple models in parallel, the color inconsistency between the models will be magnified. To address this problem, The authors of Ref. (Fu et al., 2020) proposed to jointly and iteratively optimize the camera poses, texture, and geometry of the reconstructed model, and the color consistency between the key frames in RGB-D reconstruction, which can recover not only fine-scale geometry but high-fidelity texture. A large number of parameters must be optimized, and mesh data are required, which leads to low efficiency, so it is not suitable for large-scale aerial imagery. Color correction across multi-view images can be used as a preliminary step of these methods. Combining these two type of algorithms can obtain better color consistency.

<sup>1</sup> <https://github.com/nmoehrle/mvs-texturing>.

### 2.3. Color correction across image sets

Many color correction algorithms are used in image stitching and 3D reconstruction. Most of these minimize the color difference of the shared content so as to obtain a set of transformation parameters. We review some methods in image stitching for reference. The authors of Ref. (Brown and Lowe, 2007) proposed image stitching based on invariant features, and addressed the problem of color differences between images by globally applying gain compensation (a linear model) to minimize the difference of the shared content. This method has been implemented in the OpenCV stitching model. The authors of Ref. (Xiong and Pulli, 2010) proposed a gamma model for color correction rather than a linear model for the luminance channel in the YCrCb color space to avoid the overflow of luminance. The linear and gamma models are insufficiently flexible to accommodate some challenging data. The authors of Ref. (HaCohen et al., 2013) modeled the color mapping function as a quadratic spline curve, which was flexible enough to correct significant color disparities. However, the non-rigid dense correspondence (NRDC) method used by this approach to obtain dense correspondence is time-consuming, and therefore not suitable for large-scale image sets. The authors of Ref. (Xia et al., 2019) formulated gradient preservation or contrast enhancement with model parameters based on a quadratic spline curve. However, the single-channel optimization strategy was unable to solve the white balance problem.

Parametric and global methods are similarly preferred for 3D reconstruction with large-scale multi-view image sets to limit the computational burden. The authors of Ref. (Moulon et al., 2013) proposed a global multi-view color consistency algorithm based on a linear model to optimize the color consistency of multi-view images via sparse points generated using the structure from motion (SfM) technique. The algorithm has been implemented in openMVG (Moulon et al., 2016). Common color information between images was selected robustly for histogram matching. Three global transformation models were estimated for each image by minimizing the difference between their common color information through global optimization. This method introduces very few parameters, which shows that the algorithm is efficient, but at the same time the linear model is not flexible enough. The authors of Ref. (Shen et al., 2017) minimized the color discrepancy of multi-view images to obtain a series of global linear functions, which used triangulation mesh to compute correspondence and construct a more complex energy function than (Moulon et al., 2013). This method can obtain reliable color matching information, particularly for low-texture areas such as roads and walls. However, additional data are usually required, and it is time-consuming to calculate the correspondence using mesh. The authors of Ref. (Park et al., 2016) addressed this problem by performing SIFT (Lowe, 2004) feature matching and estimating the gamma model parameters through robust and efficient low-rank matrix factorization. However, due to its inflexible model, it was not as good as that of HaCohen et al. (2013), and the estimated albedo tended to be overly dark for rigid scenes and under shadow for surfaces because the estimated parameters could not be guaranteed to be accurate with respect to ground-truth camera parameters.

Most of these methods tend to construct a graph of multi-view images and calculate cost from a pair of images, which forms an edge in the graph. To improve efficiency, thresholds must be used to filter out unfavorable edges. However, the complex camera motions still make this graph complicated, and an efficient method has yet to be developed. In addition, the transformation model of current algorithms for multi-view images is still not flexible enough to process more challenging data. Inspired by HaCohen et al. (2013) and the idea of hierarchical optimization, we adopt flexible enough quadratic splines as the global transformation function to deal with challenging datasets, and construct a quality-aware energy function with tie points as the unit to avoid building a complicated graph, which is different from existing methods. In addition, a parallelizable hierarchical strategy is proposed to improve the efficiency when facing large scale image sets.

## 3. Proposed approach

### 3.1. Overview

Given a set of multi-view images for a large-scale scene, we seek to weaken the color discrepancy between images based on shared content. The color consistency between images can be corrected according to a user-specified reference image. On the basis of HaCohen et al. (2013), we modify the contradiction term of the loss function (describe the difference in shared information between images) and add some constraints to better apply to multi-view images, and retain some constraints of HaCohen et al. (2013), such as the second derivative constraint. We first apply SfM to generate sparse points as shared content of images and then construct a symmetric (Vallet and Lelégard, 2013) lost function to minimize the variance of color values over all observations of every sparse point instead of difference between a pair of images used by HaCohen et al. (2013), to obtain a set of the flexible enough quadratic spline functions. In addition, in order to preserve the gradient information and contrast of the image, we add a simple constraint to the gradient of the curve which differs from the two step method proposed by HaCohen et al. (2013). Finally, We propose a hierarchical color correction strategy based on the m-ray tree structure to process large-scale multi-view images.

### 3.2. Sparse correspondence

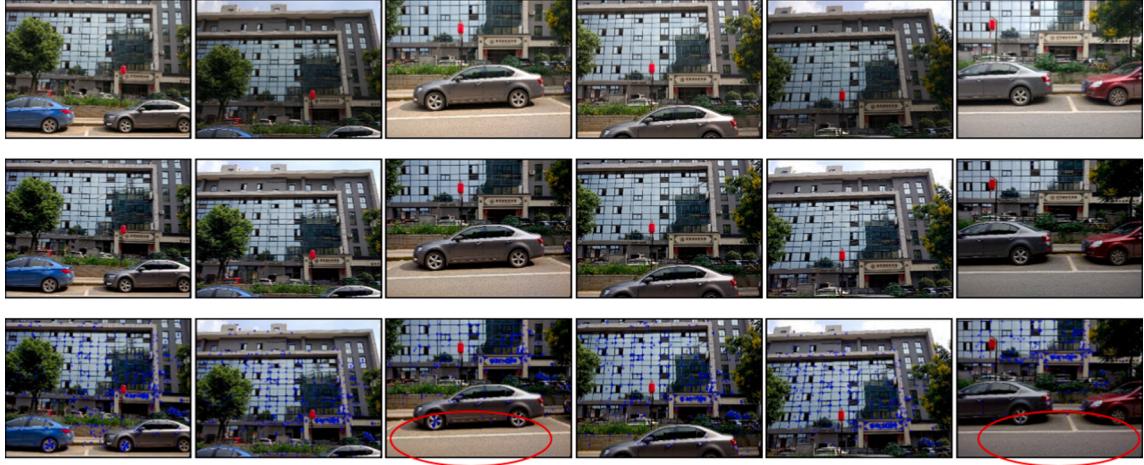
In 3D reconstruction, to make an image entirely cover a specific scene, the multi-view image has a high degree of overlap. Hence, there is a strong constraint relationship between images, which enables us to obtain a reliable SfM geometry structure to guarantee the algorithm's robustness. Experiments show that the baseline algorithm that generates sparse points can meet the needs of our algorithm. An advantage of sparse points is a low computing cost. The desired effect can be achieved even with few control points. Of course, we can also use dense points for optimization to increase the accuracy, but this will increase the computational burden. The disadvantage is that the number of feature points for some weakly textured areas will be small, which may cause color discrepancies in these areas to be ignored. If these areas have a large color discrepancy between images, it may fail in these regions, as Fig. 1 shows. A number of open-source libraries, such as openMVG and colmap (Schonberger and Frahm, 2016), or software such as ContextCapture (ContextCapture, 2018) or PhotoScan (Agisoft photoscan professional, 2018), can obtain the robust sparse points of images by SfM. We use the SfM toolchain of openMVG and ContextCapture to export sparse points and matching information between images. Information on SfM theory is available elsewhere (Schonberger and Frahm, 2016; Gherardi et al., 2010; Agarwal et al., 2011; Crandall et al., 2011). The input can be represented by  $O = O_i$  and  $I = I_i$ , where  $O_i$  is the set of RGB values for all observations of the  $i$ -th sparse point, and  $I$  is the set of multi-view images.

### 3.3. Quadratic splines

Based on the work of HaCohen et al. (2013), a global quadratic spline curve is chosen for every color channel of the image to fit various transformation models. Each curve includes six nodes, with a total of 15 unknowns; hence, a three-channel image has 45 unknowns. Experiments showed that a six-node quadratic spline is sufficiently flexible for challenging datasets. Of course, greater flexibility can be obtained using more nodes, but this will increase the computational burden. A quadratic spline is a piecewise-polynomial of the form

$$f_i^k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2, \quad k = 0, 1, \dots, n-1, \\ x \in [0, 1], x_k \in \{step^*k\}_{k=0}^{n-1} \quad (1)$$

where  $x$  are input intensities, which are remapped to the interval  $[0, 1]$ ,



**Fig. 1.** Results of our approach on GROUND. Row 1: original images; Row 2: corrected images; Row 3: corrected images with features. Blue points represent features. Red circles represent weakly textured areas where color cannot be corrected because there is no correspondence. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$x_k$  is the  $k$ -th of  $n$  nodes, and  $\text{step} = x_{k+1} - x_k$ .

We adopted a global parameter color model rather than a local model for three reasons. The global model is more robust to errors in correspondences. Like the constraints we used, the global model is easily normalized. Fewer unknowns in the global parameter model ensure more efficient optimization. This is particularly important for large-scale image-based 3D reconstruction.

#### 3.4. Color consistency optimization

To achieve color consistency between multi-view image sets, the results should adhere to three criteria (HaCohen et al., 2013):

- (1) Differences between pixels depicting equivalent content should be minimized.
- (2) Images after color adjustment should be as similar as possible to the original image.
- (3) Unsightly visual artifacts, such as the loss of gradient information or contrast, must be avoided.

These criteria are met by finding a set of quadratic spline functions that globally transform the multi-view images to obtain color consistency. We independently optimize the energy function in each channel of the RGB color space. The energy function of any color channel is

$$\{f_i\} = \operatorname{argmin}(C_\sigma + \lambda_1 C_\varphi + \lambda_2 C_\omega), \quad (2)$$

subject to :  $C_{\text{hard}}(f_i), \forall i \in \{1, \dots, n\}$ ,

where  $C_\omega$  is the degree of dispersion of all pixels depicting the same content,  $C_\varphi$  balances the similarity between the adjusted and original images,  $C_\omega$  penalizes the second derivative to ensure a smooth quadratic spline curve, and  $C_{\text{hard}}$  represents three types of hard constraints. These terms are defined as follows.

For  $C_\omega$ , we use the variance of the RGB values of all observations for every sparse point to represent the degree of dispersion,

$$C_\sigma = \sum_{P_i \in P} \frac{\sum_{p_j \in P_i} (f_j(p_j) - \text{mean}_i)^2}{m_i}, \quad (3)$$

where  $P_i$  is the set of RGB values of all observations of the  $i$ -th sparse point,  $\text{mean}_i = \frac{\sum_{p_j \in P_i} f_j(p_j)}{m_i}$  is the average value of the adjusted  $P_i$ , and  $m_i$  is the number of observations of the  $i$ -th sparse point. This is more efficient and elegant than calculating the cost of every pair of images as a unit, and no complex graph structure must be calculated, as is the case with

most conventional methods (HaCohen et al., 2013; Shen et al., 2017; Xia et al., 2019).

$C_\varphi$  is defined as

$$C_\varphi = \sum_{x \in \Omega_N} (f_i(x) - x)^2, \quad (4)$$

where  $\Omega_N$  is the set of some uniformly distributed discrete points on the  $x$ -axis. Based on experiments, we set the soft constraint  $\lambda_1$  to 0.001 in all of our experiments, and expected results were obtained from all our datasets. A smaller  $\lambda_1$  means that we give more weight because we want better color consistency between images. We define  $C_\omega$  as

$$C_\omega = \sum_{x \in \Omega_N} (f''_i(x))^2. \quad (5)$$

Experiments indicated that the proposed algorithm was not sensitive to this term, but it is generally necessary to avoid significant errors.

The first type of hard constraint  $C_{\text{hard}}$  is inherent to splines; it ensures a continuous and derivable function, and is given as

$$\begin{aligned} f_i(x_i) &= f_i(x_{i+1}), i = 0, 1, \dots, n-2 \\ f'_i(x_i) &= f'_i(x_{i+1}), i = 0, 1, \dots, n-2. \end{aligned} \quad (6)$$

The first constraint includes  $n-1$  equations that ensure continuity at interior points. The second constraint consists of  $n-1$  equations to ensure a continuous slope at interior points.

The second type of  $C_{\text{hard}}$  constraint, a reference image constraint, is applied because the color of a reconstructed image may appear unnatural even with optimal color correction. This constraint is applied when a user selects one or more reference images and transfers their color style to other images in an image set. Fig. 2 compares the color consistency correction applied to images in the top row with and without using the reference image in the red box. These are respectively given in the second and third rows. This constraint can be given as

$$f_i(x) - x = 0, \forall i \in \Omega_R, \quad (7)$$

where  $\Omega_R$  is the set of reference image indices.

The third type of  $C_{\text{hard}}$  constraint is designed to preserve the gradient and contrast of the images. Unlike previous work (Xia et al., 2019), we add no gradient term to the energy function  $C$ , and directly constrain the gradient values of some nodes of the quadratic spline curve. In fact, the loss of gradient information is due to those portions of the quadratic spline curve with gentle slopes. This problem is easily addressed with the following constraints to the first derivative of the quadratic spline curve:



**Fig. 2.** Color consistency results obtained with and without applying a reference image constraint. The red box represents the reference image, row 1 presents the input images, row 2 presents the results without reference images, and row 3 represents the results with one reference image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\begin{aligned} \delta_{\min} &\leq f_i'(x_j) \leq \delta_{\max}, \forall x_j \in \Omega_N, \\ f_i(0) &\geq \varepsilon_{\min} f_i(1) \leq \varepsilon_{\max} \end{aligned} \quad (8)$$

where  $\delta_{\min}$  and  $\delta_{\max}$  are the maximum and minimum slope values, respectively, and  $\varepsilon_{\min}$  and  $\varepsilon_{\max}$  define the gamut in every color channel. The color dynamic range must be preserved to ensure that color values do not overflow and detract from the image contrast. This is achieved through appropriate values of  $\varepsilon_{\min}$  and  $\varepsilon_{\max}$ . However, the use of a reference image provided positive results during testing even without adding the constraints of Eq. 8. All global parameters of  $f_i$  can be solved for by convex quadratic programming (Goldfarb and Idnani, 1983) using an open-source library QuadProg++.<sup>2</sup> The final energy function is

$$C = x^T(A + I + G)x + (g_i + g_g)x + g_{i0} + g_{g0} = x^TGx + gx + g0, \quad (9)$$

where  $x$  is the vector of unknown parameters,  $A$  is a matrix of the variances of all observations of each sparse point,  $I$  is a matrix of the degrees to which new pixel values deviate from the original values,  $G$  is a matrix that limits the size of the second derivative,  $g_i$  and  $g_g$  are corresponding first-order matrices, and  $g_0$  is a constant matrix. Because the  $G$  matrix is positive definite, the global optimal solution can be obtained.

### 3.5. Hierarchical optimization

There are lots of images in 3D reconstruction involving large scenes, in which case the dimensions of  $x$  and  $G$  in Eq. 9 will be very large. This greatly increases memory consumption and reduces the efficiency of the color-correction algorithm owing to the large number of matrix operations, particularly multiplication, required to calculate  $G$  and  $g$ . We address this through a hierarchical optimization strategy, which is easily accelerated using parallel operations such as multi-threading or cluster computing.

The set of images is first divided into  $n$  groups, then we construct an image tree as shown in Fig. 3. The images of a group form a virtual image used in the color correction of that tree level. Then, we perform color correction on each node (group) at that tree level from level 1. This is hierarchically repeated until reaching the root node of the image tree. For a tree depth of  $h$ , color correction is performed for nodes with nonzero degrees enclosed in the figure from bottom to top, beginning with the 1-th level. All the children of one node use the same set of parameters for color correction. The process is easily accelerated via parallel operations because each node with the same level is completely independent. Moreover, the number of child nodes of each node is relatively small, so the scale of  $x$  is also small, which can greatly decrease memory requirements and improve the effectiveness of color correction.

For general scenes, the photography centers of all images can be grouped using k-means clustering, graph cuts, or other methods (Zhu

et al., 2018) to ensure that the images of each group are connected through sparse points. We group images using the openMVG toolkit, which implements augmented discriminative clustering.

We analyze the computation time required for color correction using the proposed hierarchical color consistency correction algorithm by applying an m-ary tree for image grouping as an example. We assume that the number of images is  $n$ , the number of nodes with a degree of 0 is  $L$ , and the number of other nodes is  $l$ . The computation time required to correct the color of a group of images is  $C(m)$ , where  $m$  is the maximum number of images in a group. The properties of the m-ary tree indicate that  $L = (m-1)*l + 1$ . Any node with a degree of 0 represents a single image, so  $n = L$  and  $l = (n-1)/(m-1)$ . Therefore, the total computation time is  $T = C(m) \times ((n-1)/(m-1))$ . Assuming  $C$  is constant, the algorithmic complexity,  $O(T(n)) = O(n)$ , is linear.

$C(m)$  is generally related to the numbers of images and sparse points, and it is the smallest time unit in our algorithm. If  $m = 2$ , then the overall tree structure is that of a binary tree, which is most common. However, the tree depth increases with decreasing  $m$ , and this increases the number of color transformations that must be applied to the images. Accordingly, multi-level correction may result in the accumulation of errors and greater color distortion. Therefore, the depth of the tree must be reduced to the extent possible. We set the maximum number of images in a group,  $m$ , to 40. As such, an m-ary tree is constructed with 1600 images, with a maximum depth of 2. Accordingly, images require only two transformations.

Calculating the matrix  $A$  in Eq. 9 is time-consuming operation in our algorithm, whose time consumption is directly proportional to the number of sparse points. Hence, the computational efficiency of the algorithm can be controlled by the number of sparse points. We employ all sparse points generated by SfM in the color-correction process to achieve the best results.

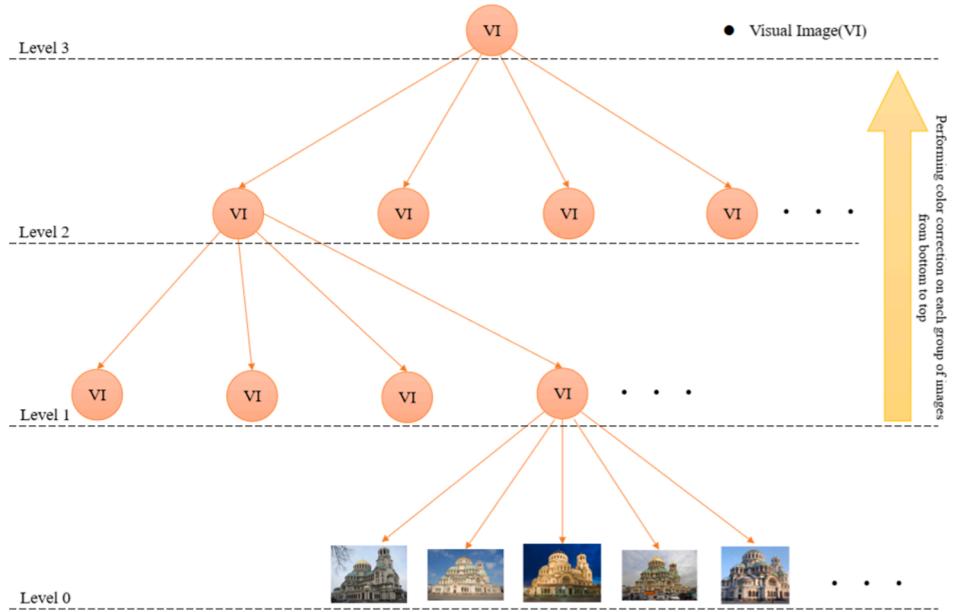
## 4. Results and discussion

### 4.1. Datasets

The proposed 3D reconstruction approach was evaluated using datasets of various types. Table 1 lists their details. In addition, we conduct experiments on five challenging internet datasets to evaluate the effectiveness of proposed method, which will be introduced in Section 4.2. ZIEGELEIPARK is from ODMdata.<sup>3</sup> Navona (Farenzena et al., 2009), HOUSE2 and GROUND are captured by a handheld camera. The other datasets are captured by a camera equipped on UAV. HOUSE, HOUSE2 and SCHOOL were made more challenging by using Photoshop to exaggerate color differences between images. We compare the proposed algorithm with the existing two algorithms on HOUSE and SCHOOL. HK-0043HY, HOUSE2, and Navona were used to evaluate the

<sup>2</sup> <https://github.com/liuq/QuadProgpp>.

<sup>3</sup> <https://github.com/OpenDroneMap/ODMdata>.



**Fig. 3.** Hierarchical color correction algorithm based on m-tree structure for large-scale image sets. The node in the current layer ( $\text{level} \geq 1$ ) is treated as a virtual image to participate in color correction, and its child nodes share the same parameters of the visual image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 1**

Details of datasets employed in experiments.

	HOUSE	SCHOOL	HOUSE2	Navona	GROUND	TJH6	HK-0012GZHP	HK-0043HY	HK-0003FH	ZIEGELEIPARK
No. of images	36	55	13	92	50	55	1758	1424	1320	6796
Size	4592*3058	4864*3648	3872*2592	4000*3000	4608*3456	4864*3648	7360*4912	5472*3648	6000*4000	4000*3000
Platform	UAV	UAV	Handheld	Handheld	Tbox	UAV	UAV	UAV	UAV	UAV
Sparse points	12.7k	5.4k	5.3k	24.3k	1.9k	89.5k	504k	251k	459.30k	437.32k
No. of groups	1	1	1	1	1	1	40	42	38	102

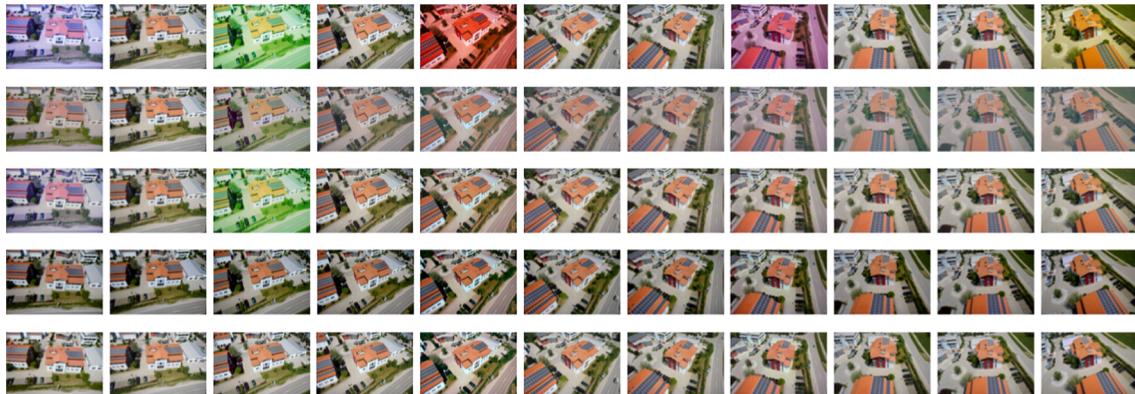
performance of our method on textured models. HK-0012GZHP, HK-0043HY, HK-0003FH, and ZIEGELEIPARK have large-scale images that were used to evaluate the efficiency of the three methods.

#### 4.2. Comparison with other methods

We compare the performance of the proposed method to that of two other methods (Park et al., 2016; Moulon et al., 2013) to qualitatively and quantitatively evaluate its effectiveness.

#### 4.2.1. Qualitative evaluation

We qualitatively evaluate the three methods on two typical datasets, HOUSE and SCHOOL. There is a large color inconsistency between the images of HOUSE. The images of SCHOOL have large exposure inconsistencies. Fig. 4 presents the experimental results obtained by the three color-correction approaches for the HOUSE dataset, from which we can see that the results of Moulon et al. (2013) exhibit worse color consistency than the other methods, and the loss of gradient information is severe, causing images to appear hazy. This occurs because the linear model is not flexible enough. When the color inconsistency between



**Fig. 4.** Color correction results for HOUSE dataset. Due to space limitations, only some images are listed. All results can be found on the website (Link: <https://drive.google.com/drive/folders/1LjQ1cJbnLlyqPKZiBw7JH2w7XjBXXAvh?usp=sharing>). (Row 1) input images; (Row 2) results of Moulon et al. (2013); (Row 3) results of Park et al. (2016); (Row 4) results of proposed approach without reference image; (Row 5) results of proposed approach with reference image.

images is large, the slope of the linear model is relatively small, which will cause the loss of image contrast. The results of Park et al. (2016) exhibit better gradient information retention than those of Moulon et al. (2013), owing to its albedo-based correction model. However, the color is not as consistent as ours, especially in the first and third images, which is related to the inflexible model. We note from rows (d) and (e) of Fig. 4 that the color consistency obtained by the proposed approach is the best of the three methods, and the image structure, such as gradient and contrast, is well preserved, regardless of whether a reference image is added. Fig. 5 presents the experimental results obtained by the three color-correction approaches for the SCHOOL dataset, from which it is seen that the results of Moulon et al. (2013) are still inferior to those of the other two methods in terms of color consistency and contrast. The method used by Park et al. (2016) and the proposed approach have similar color consistency, but ours is slightly better, especially on the eleventh image, where correction by Park et al. (2016) fails. This indicates that the proposed method is more robust than (Park et al., 2016), which is due to the stable correspondence of the SfM structure and quality-aware cost function. The results also demonstrates that our algorithm is suitable for images with large color inconsistencies as well as those with inconsistent exposure.

We also tested our algorithm on five internet datasets, namely Gendarmenmarkt, Notre Dame (Wilson and Snavely, 2014), Spilled Blood Cathedral St Petersburg (Enqvist et al., 2011), Brandenburg gate (Heinly et al., 2014), Train (Knapitsch et al., 2017). Fig. 6 shows the results of our algorithm on these datasets. Due to space limitations, we only show some images. The full results are available at the website (Link: <https://drive.google.com/drive/folders/1LjQ1cJbnLlyqPKZiBw7JH2w7XjBXXAVh?usp=sharing>). Internet images reflect different times, exposures, and cameras, so there is great color inconsistency between them. It can be seen that the proposed method achieves good color consistency on these five datasets, especially the Train dataset in Fig. 6, and there is no obvious gradient or contrast loss. The images of the Brandenburg gate, which were captured at night, are also well corrected. The building presents a consistent color. The visual results confirm again that the proposed method is powerful enough for challenging datasets.

#### 4.2.2. Quantitative evaluation

An important indicator to measure color correction is the similarity of the corrected image to the original. As a quantitative evaluation, the peak signal-to-noise ratio (PSNR) (Fisher, 2012) and structural similarity (SSIM) (Wang et al., 2004), which respectively measure the color and structure similarity, are used to evaluate the three methods. They are formulated as

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (10)$$

$$\text{PSNR}(x, y) = 20.0 * \log_{10}(L/RMS)$$

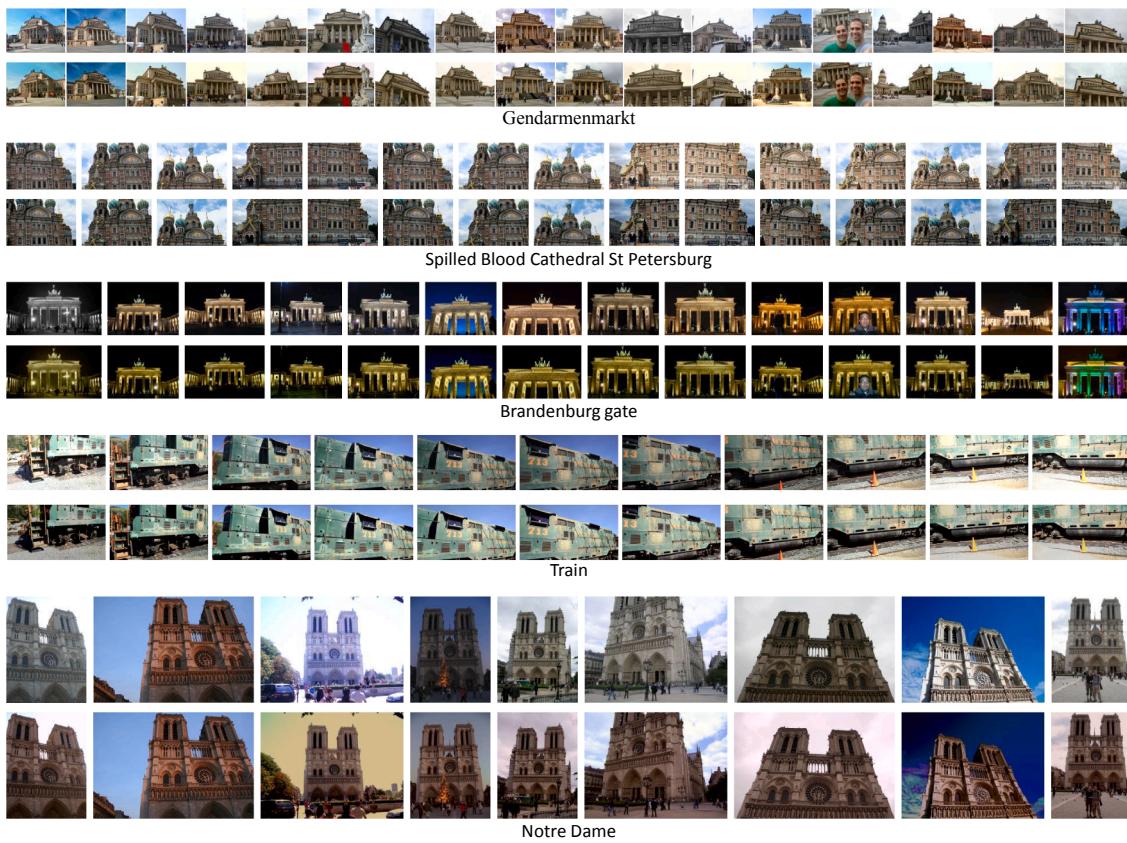
where  $L$  is the largest possible value in the dynamic range of an image;  $RMS$  is the root mean square difference between two images;  $\mu_x$  and  $\mu_y$  are the average values of  $x$  and  $y$ , respectively;  $\sigma_x^2$  and  $\sigma_y^2$  are their associated variances; and  $\sigma_{xy}$  is the covariance of  $x$  and  $y$ .  $c_1 = (k_1 L)^2$ ,  $c_2 = (k_2 L)^2$  are used to maintain stability constant. We set  $k_1 = 0.01$ ,  $k_2 = 0.03$ . The structural similarity ranges from 0 to 1. When the two images are exactly the same, SSIM is equal to 1. The larger the two values, the more similar the two images.

We use the average value of PSNR and SSIM of the images as a metric. Table 2 shows the results of the quantitative evaluation of three methods. It can be seen that the results of Moulon et al. (2013) are inferior to those of the other two algorithms in two aspects, especially the performance on large datasets. Our algorithm with reference images achieved the best results on PSNR. (Park et al., 2016) achieved the best results on SSIM. This shows that our algorithm retains more color information than the Park algorithm, owing to the  $C_\varphi$  term, but less image structure information. Fig. 5 also shows that the details of the image have been enhanced. Because we want to prevent the loss of detail and contrast, we set  $\delta_{min}$  to 1.0, which may enhance the details. The global transformation method only transforms the pixel value, and the neighborhood information between pixels is not considered, which may lead to the loss of image structure. Lacking a reference image, our algorithm performs worse than the others on small datasets, both on PSNR and SSIM. If the reference image is missing, our algorithm will change the color information as much as possible to make it more consistent across images. However, our results are better than those of Moulon et al. (2013) on large datasets (TJH6, Navona, HK-0001BX01), and the values of PSNR and SSIM are stable and without significant fluctuations, which shows that our algorithm has strong robustness on big datasets.

In fact, the value of  $\lambda_1$  in Eq. 2 has a large impact on PSNR and SSIM. If  $\lambda_1$  is set to be small, the values of PSNR and SSIM will decrease accordingly, and the color consistency will be better. However, if the setting is large, the image information will be retained more and the color consistency will be worse. There is a tradeoff between color consistency and PSNR and SSIM. If  $\lambda_1$  is set to 0.001, then we will see expected and stable results, and this setting was used in all experiments. In addition, as shown in Fig. 7, the value of  $\delta_{min}$  is an important parameter that directly affects the details and gradient of the corrected images. To



**Fig. 5.** Color correction results for SCHOOL dataset. Due to space limitations, only some images are listed. All results can be found on the website (Link: <https://drive.google.com/drive/folders/1LjQ1cJbnLlyqPKZiBw7JH2w7XjBXXAVh?usp=sharing>). Row (1) input image data; Row (2) results of Moulon et al. (2013) using reference image; Row (3) results of Park et al. (2016); Row (4) results of proposed approach without reference image; Row (5) results of proposed approach with reference image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 6.** Color correction results of five internet datasets. The upper row in each pair shows the original images, and the lower row shows our color correction results. The readers can zoom in for details. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 2**  
PSNR and SSIM results of three methods on seven datasets.

	HOUSE		SCHOOL		HOUSE2		GROUND		TJH6		Navona		HK-0001BX01	
PSNR/SSIM	PNSR	SSIM												
Moulon	24.338	0.897	22.955	0.841	20.798	0.924	21.935	0.895	16.682	0.836	13.663	0.739	23.635	0.952
Park	21.564	<b>0.937</b>	21.062	<b>0.890</b>	25.472	<b>0.962</b>	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
ours	20.931	<b>0.879</b>	18.343	<b>0.819</b>	25.164	<b>0.936</b>	23.186	0.897	<b>24.794</b>	0.949	24.937	0.924	<b>30.232</b>	<b>0.983</b>
ours-ref	<b>25.571</b>	0.900	<b>25.751</b>	0.856	<b>26.945</b>	0.945	<b>25.027</b>	<b>0.905</b>	<b>24.447</b>	<b>0.950</b>	<b>25.922</b>	<b>0.936</b>	29.236	0.978

set it to be very small will cause loss of gradient and contrast. However, if it is large, then the details of the image will be excessively enlarged, which will enhance the image gradient. Promising results were obtained for all of our datasets when  $\delta_{min}$  was set to 1.0 for the remapping curves of every color channel.

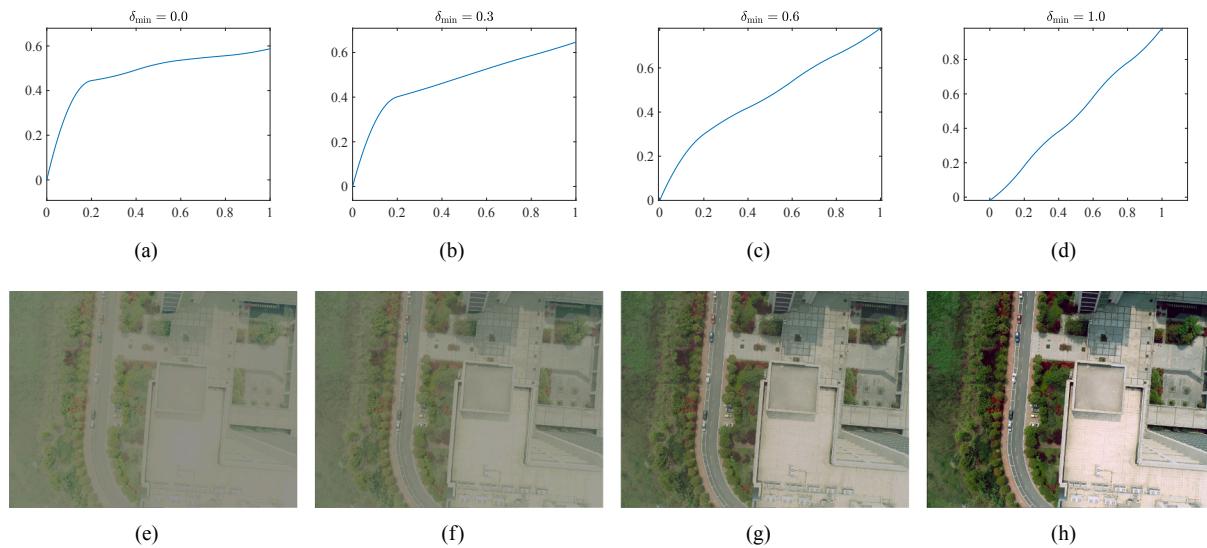
In summary, our algorithm achieved the best results of the three methods on visual performance on two types of datasets, and on PSNR on all datasets. This can be attributed to the quadratic spline transformation model, which is more flexible than linear and gamma models, and to the quality-aware cost function. Reliable correspondence of SfM ensures the robustness of the proposed method. Experimental results on internet data also show that our algorithm has a strong generalization ability. These all benefit from the stable correspondence of the SfM structure and quality-aware energy function. However, because our algorithm can enhance the image details, the image structure is not as good as (Park et al., 2016), which shows that our algorithm still has room for improvement. Adding some constraints on image structure information or adding the neighborhood information between pixels may improve the performance.

#### 4.3. Performance on texture mapping

Based on our experiments, we found that the proposed method can globally improve the visual aspect of the textured model. We performed experiments with respect to performance on texture mapping.

The texture reconstruction performance of the proposed color-correction approach is demonstrated by its application to the Navona, HOUSE2, and HK-0043HY datasets. Navona and HOUSE2 were corrected with the proposed approach without hierarchical optimization. HK-0043HY, with large-scale images, was corrected by the proposed method with hierarchical optimization. The results were evaluated to determine if visual appearance can be improved. We used ContextCapture software to obtain a triangulation mesh as part of the 3D reconstruction process. This process includes SfM, dense matching, and triangular meshing. We used mvs-texturing to complete the texture mapping. Here, mvs-texturing implemented the algorithm of Waechter et al. (2014) to optimize an energy function based on a Markov random field (MRF) to obtain image labels for all triangles.

Fig. 8 shows visual results to illustrate the effectiveness of texture mapping of the proposed approach on the three datasets. There are large color discrepancies on HOUSE2 and exposure discrepancies on Navona. If we do not correct the images, because different texture blocks use



**Fig. 7.** Color correction results of one of image of SCHOOL dataset with different values of  $\delta_{\min}$ , where row 1 presents the remapping curves of the red channel of the image, and row 2 presents the results with different values of  $\delta_{\min}$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

different image pixel values, there will be color inconsistency and obvious seams on the textured model, which detracts from the scene's realism. It can be seen that the color-corrected image can be used for texture mapping, the global color is consistent, and local visible seams have basically disappeared. Our method achieves a uniform color tone and better visual appearance on these two challenging datasets, whether with color or exposure inconsistency.

HK-0043HY has different exposures between different flight belt images, which causes the overall color of the texture result to show a global color inconsistency (Fig. 8(g)). The upper-left corner is darker, the lower-right corner is brighter, and there are obvious seams such as on buildings and roads, as seen in Fig. 8 (1)–(6). Through hierarchical color optimization, we found that the result shows a global consistent style, and the seams have basically disappeared. This shows that our hierarchical algorithm has robust properties on larger datasets, because we treat a group of images as a virtual image for color correction, which expands the field of view of a single (virtual) image. It can not only ensure the effect on large datasets, but can improve the efficiency of the algorithm. However, there is some loss of contrast in the upper left corner, this is because there is a weakly textured building in the upper left corner. These areas have fewer features, which will be detrimental to the color correction results as analyzed in Section 3.2. It is worth mentioning that the purpose of our algorithm is to globally correct image color without considering local color differences. A better visual experience can be obtained by combining local color differences using a local blending algorithm, like that of Waechter et al. (2014).

#### 4.4. Efficiency analysis

The efficiency of the proposed approach is compared to that of the algorithms developed by Moulon et al. (2013) and Park et al. (2016) on the HOUSE, SCHOOL, TJH6, GROUND, HK-0012GZHP, HK-0043HY, HK-0003FH, and ZIEGELIEPARK datasets. The algorithm of Moulon et al. (2013) and the proposed algorithm were both implemented in C++. The method of Moulon et al. (2013) was integrated into the SfM toolchain of openMVG, so we called it directly. The MATLAB code provided by the authors was employed for the algorithm of Park et al. (2016). This code implements feature extraction, feature matching using image pairs as the unit, and, color correction. However, this implementation of the algorithm can require days or even weeks when processing more than twenty high-resolution images. Note that the time

efficiency of Park et al. (2016) is for reference only and is not for comparison. In addition, the time of feature extraction and feature matching steps was ignored in efficiency evaluations, and only the time of the color correction step was counted. All experiments were conducted on a desktop PC with 16 GB RAM and an Intel i7-7700 CPU operating at 3.60 GHz.

We first analyzed the relationship between the time consumption and the number of images without the use of hierarchical optimization algorithms on the Navona and SCHOOL datasets. It can be seen from Fig. 9 (left) that the relationship curve between time consumption  $t$  and number of images  $n$  is a parabola, and  $t$  will increase sharply as  $n$  increases. This is because Eq. 9 calculates the squared difference, which involves a large number of matrix multiplications, hence the complexity of the algorithm is  $o(n^2)$ . To improve performance on large datasets, we propose a hierarchical optimization method to reduce the algorithmic complexity to  $o(n)$ . Fig. 9 (right) shows the result of the hierarchical optimization algorithm on the HK-0043HY dataset. We set the maximum number of each group to 40, so the image tree has three levels. Fig. 9 (right) shows the time statistics of each level and I/O. It can be seen intuitively that  $t$  and  $n$  show a linear relationship which benefits from our hierarchical optimization. I/O accounts for a large proportion of the total time, and is the bottleneck of the proposed algorithm. We also noticed some slight parabolic trends in the middle of the total time curve. When the number of images to be constructed is a full m-tree, the number of images is  $N$  and the time consumption is  $T$ . As discussed in Section 3.5,  $N$  and  $T$  present a linear relationship. But the HK-0043HY dataset does not meet the full m-tree condition, as the number of images  $n < N$ , so the time  $t < T$ . Precisely because of this, as the number of images increases, the group number of level 2 increases, and the time of level 2 is bound to assume the form of a parabola, as discussed above, which leads to a slight parabolic form of the entire time curve. However, the time consumption is mainly in I/O and level 1. The group optimization time decreases exponentially from bottom to top. The time for levels greater than 1 is very small and can be ignored, as Fig. 9 shows.

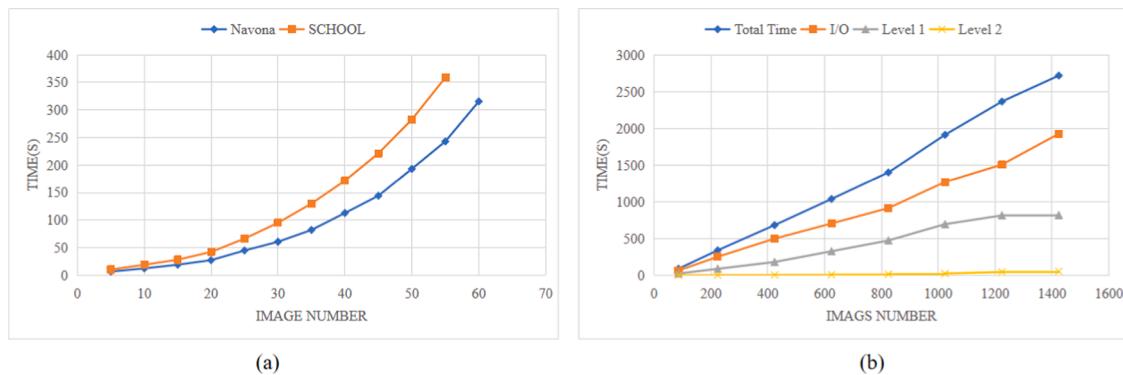
The running times required by the three algorithms for the nine considered datasets are listed in Table 3. (Park et al., 2016) is implemented using MATLAB, so the efficiency is the lowest. The results demonstrate that the proposed algorithm is much more efficient and robust than that of Moulon et al. (2013), despite the fact that we use a greater number of unknowns for more complex transformation models. As discussed in Section 3.5, the complexity of the proposed algorithm is



**Fig. 8.** Performance on texturing mapping. (a), (b): results without and with color correction on Navona; (c), (b): results without and with color correction on HOUSE2; (d), (e): details of (c) and (d); (g), (h): results without and with color correction on HK-0043HY; (1)–(6): details of (g) and (h). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$O(n)$ . However, the complexity of the (Moulon et al., 2013) algorithm is  $O(\lambda^* \rho^* C_n^2) = O(n^2)$ , where  $\lambda$  represents the time required to calculate the cost of a pair of images and  $\rho^* C_n^2$  is the number of image matching pairs. Furthermore, (Moulon et al., 2013) failed when applied to the large-scale datasets. In contrast, the proposed hierarchical strategy

enables the proposed approach to process more than one thousand images, and PSNR and SSIM remain stable, which shows the robustness of our algorithm on large-scale datasets. As (Park et al., 2016) was implemented in MATLAB, the running time provided in Table 3 cannot be directly used for comparison. However, the iterative scheme of matrix decomposition is not as efficient as our quadratic programming



**Fig. 9.** The relationship between image and time consumption. (a) Results of proposed method without hierarchical optimization on SCHOOL and Navona. (b) Proposed method with hierarchical optimization on HK-0043HY.

**Table 3**

Runtime comparisons of the three color-correction approaches for nine datasets. We used 5 images of HOUSE and 6 images of SCHOOL for efficiency analysis of small data. Unit: second.

	HOUSE	SCHOOL	GROUND	TJH6	Navona	HK-0012GZHP	HK-0043HY	HK-0003FH	ZIEGELEIPARK
Moulon et al.	9	14	442	595	Failed	Failed	Failed	Failed	Failed
Park et al.	894.503	601.749	>3600	>3600	N/A	N/A	N/A	N/A	N/A
ours	2.773	4.106	50.818	59.341	735.0	12213.803	2717.548	6110.41	45475.217

solution.

In summary, through the hierarchical optimization algorithm, we reduced the algorithmic complexity  $o(n^2)$  to  $o(n)$ , which can greatly improve the efficiency of the proposed approach on large-scale datasets. But proposed method still has two bottlenecks. First, to reduce memory consumption, during implementation, it is necessary to read image information from a disk at each level of optimization, which causes a serious I/O burden. This is one reason why I/O accounts for a large proportion, as shown in Fig. 9. There is a tradeoff between algorithmic efficiency and computer memory. Second, the calculation Eq. 9 on one group involves a large number of matrix operations, and as the number of images increases, this part of the time will increase sharply, which will greatly reduce the efficiency of the algorithm. Therefore, reducing  $C(m)$  is critical to improving the performance of our algorithm. Reducing the value of  $m$  will improve performance, but the level of the corresponding tree will also increase. This represents an increase in the number of transformations for each image, which will result in greater deviation from the original image.

## 5. Conclusion

We presented an algorithm to efficiently address color inconsistencies in large-scale multiple-view images. The essential contribution of this work is an efficient optimization method with some constraints to obtain a series of global functions. The construction of graph structures during color correction is avoided by minimizing the variance in color values for all observations of every sparse point. This approach not only can significantly increase the efficiency of the algorithm but can provide a more elegant means to solve the issue of color inconsistency. A flexible quadratic spline function is used as the global transformation function, and some simple and effective constraints are added to retain the gradient and contrast of the image, which was proved effective by extensive experiments on challenging datasets. In addition, strong geometric constraints are applied in the multi-view images to ensure the reliability of matching, which guarantees the robustness of the algorithm. Finally, a parallelizable hierarchical image color correction strategy was proposed, and a strategy was demonstrated to facilitate the easy processing of large-scale image sets by increasing the efficiency of the algorithm and reducing memory requirements.

Comparison with representative work on various datasets showed that our algorithm performed best in terms of both efficiency and effectiveness, which can improve the visual experience of large-scale images and 3D textured models.

As with most color-correction algorithms, the proposed algorithm relies on correspondence. Accordingly, the color of a weakly textured area cannot be corrected if feature points cannot be extracted, which results in color discrepancies in these areas. In addition, we did not consider the white balance in the energy function. If the reference image is not provided, there may be a color cast problem. Our future work will focus on these two problems. In addition, we will implement our algorithm on GPUs to improve its efficiency.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The work presented in this article was supported by Sichuan Key Research and Development Project (High-tech Field) “Research on Major Engineering Risk Identification and Countermeasures of Sichuan-Tibet Railway” (No. 2019YFG0460). The authors thank Park et al. and Moulon et al., who released the codes of their algorithms, and the contributor of the open-source libraries OpenCV, OpenMVG, and mvs-texturing, which enabled our experiments to run smoothly. The authors appreciate the anonymous reviewers for their careful work and thoughtful suggestions that have helped improve this paper substantially.

## References

- Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R., 2011. Building Rome in a day. Commun. ACM. <https://doi.org/10.1145/2001269.2001293>.
- Agisoft photoscan professional, version 1.4.5 (software), 2018. <https://www.agisoft.com/>.

- Brown, M., Lowe, D.G., 2007. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vision* 74 (1), 59–73. <https://doi.org/10.1007/s11263-006-0002-3>.
- ContextCapture, Version v4.4.9.516 (software), 2018. <https://www.bentley.com/en/products/brands/-contextcapture>.
- Crandall, D., Owens, A., Snavely, N., Huttenlocher, D., 2011. Discrete-continuous optimization for large-scale structure from motion. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/CVPR.2011.5995626>.
- Eden, A., Uyttendaele, M., Szeliski, R., 2006. Seamless image stitching of scenes with large motions and exposure differences, in. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/CVPR.2006.268>.
- Enqvist, O., Kahl, F., Olsson, C., 2011. Non-sequential structure from motion. In: Proceedings of the IEEE International Conference on Computer Vision. <https://doi.org/10.1109/ICCVW.2011.6130252>.
- Farenzena, M., Fusello, A., Gherardi, R., 2009. Structure-and-motion pipeline on a hierarchical cluster tree. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009, 2009. doi: 10.1109/ICCVW.2009.5457435.
- Faridul, H.S., Pouli, T., Chamaret, C., Stauder, J., Tremeau, A., Reinhard, E., 2014. A Survey of Color Mapping and its Applications, Eurographics 2014-State of the Art Reports (2014), pp. 43–67. doi: 10.2312/egst.20141035.
- Fisher, Y., 2012. *Fractal image compression: theory and application*. Springer Science & Business Media.
- Fu, Y., Yan, Q., Liao, J., Xiao, C., 2020. Joint Texture and Geometry Optimization for RGB-D Reconstruction. <https://doi.org/10.1109/cvpr42600.2020.00599>.
- Gherardi, R., Farenzena, M., Fusello, A., 2010. Improving the efficiency of hierarchical structure-and-motion, in. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/CVPR.2010.5539782>.
- Goldfarb, D., Idnani, A., 1983. A numerically stable dual method for solving strictly convex quadratic programs. *Math. Programm.* 27 (1), 1–33. <https://doi.org/10.1007/BF02591962>.
- HaCohen, Y., Shechtman, E., Goldman, D.B., Lischinski, D., 2013. Optimizing color consistency in photo collections. *ACM Trans. Graphics* 32 (4). <https://doi.org/10.1145/2461912.2461997>.
- He, M., Liao, J., Yuan, L., Sander, P.V., 2017. Neural Color Transfer between Images. arXiv: 1710.00756arXiv:1710.00756.
- Heinly, J., Dunn, E., Frahm, J.-M., 2014. Correcting for Duplicate Scene Structure in Sparse 3D Reconstruction. In: European Conference on Computer Vision (ECCV).
- Hwang, Y., Lee, J.Y., Kweon, I.S., Kim, S.J., 2014. Color transfer using probabilistic moving least squares, in. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/CVPR.2014.427>.
- Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V., 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. In: ACM Transactions on Graphics. <https://doi.org/10.1145/3072959.3073599>.
- Kuse, M., Jaiswal, S.P., 2015. Graph modelling of 3D geometric information for color consistency of multiview images. In: Proceedings - International Conference on Image Processing, ICIP 2015-Decem (November), pp. 1394–1398. <https://doi.org/10.1109/ICIP.2015.7351029>.
- Lempitsky, V., Ivanov, D., 2007. Seamless mosaicing of image-based texture maps, in. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/CVPR.2007.383078>.
- Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*. <https://doi.org/10.1023/B:VISL.0000029664.99615.94>.
- Luan, F., Paris, S., Shechtman, E., Bala, K., Deep photo style transfer. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017. arXiv: 1703.07511. doi: 10.1109/CVPR.2017.740.
- Moulou, P., Duisit, B., Monasse, P., Moulou, P., Duisit, B., Monasse, P., 2013. G.M.-v. Color, Global Multiple-View Color Consistency To cite this version: Global Multiple-View Color Consistency.
- Moulou, P., Monasse, P., Perrot, R., Marlet, R., 2016. Openmvg: Open multiple view geometry. In: International Workshop on Reproducible Research in Pattern Recognition. Springer, pp. 60–74.
- Nguyen, R.M., Kim, S.J., Brown, M.S., 2014. Illuminant aware gamut-based color transfer. *Comput. Graphics Forum*. <https://doi.org/10.1111/cgf.12500>.
- Pan, R., Taubin, G., 2015. Color adjustment in image-based texture maps. *Graph. Models* 79, 39–48. <https://doi.org/10.1016/j.gmod.2015.04.002>.
- Park, J., Tai, Y.W., Sinha, S.N., Kweon, I.S., 2016. Efficient and robust color consistency for community photo collections. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-Decem, pp. 430–438. <https://doi.org/10.1109/CVPR.2016.53>.
- Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P., 2001. Color transfer between images. *IEEE Comput. Graphics Appl.* <https://doi.org/10.1109/38.946629>.
- Schonberger, J.L., Frahm, J.M., 2016. Structure-from-Motion Revisited. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2016.445.
- Shen, T., Wang, J., B., T.F., Zhu, S., Quan, L., 2017. Color Correction for Image-Based Modeling in the Large. In: ACCV 10111, pp. 392–407. <https://doi.org/10.1007/978-3-319-54181-5>.
- Su, Z., Zeng, K., Liu, L., Li, B., Luo, X., 2014. Corruptive artifacts suppression for example-based color transfer. *IEEE Trans. Multimedia*. <https://doi.org/10.1109/TMM.2014.2305914>.
- Vallet, B., Lelégard, L., 2013. Partial iterates for symmetrizing non-parametric color correction. *ISPRS J. Photogramm. Remote Sens.* <https://doi.org/10.1016/j.isprsjprs.2013.05.005>.
- Waechter, M., Moehrle, N., Goesele, M., 2014. Let There Be Color! Large-Scale Texturing of 3D Reconstructions BT - Computer Vision – ECCV 2014. In: Computer Vision–ECCV, 2014.
- Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* 13 (4), 600–612.
- Wang, B., Yu, Y., Xu, Y.Q., 2011. Example-based image color and tone style enhancement. *ACM Trans. Graph.* <https://doi.org/10.1145/1964921.1964959>.
- Wilson, K., Snavely, N., 2014. Robust global translations with 1dsfm, in. In: Proceedings of the European Conference on Computer Vision (ECCV).
- Xia, M., Yao, J., Gao, Z., 2019. A Closed-Form Solution for Multi-view Color Correction with Gradient Preservation. *ISPRS J. Photogramm. Remote Sens.* 157 (May), 188–200. <https://doi.org/10.1016/j.isprsjprs.2019.09.004>.
- Xiao, X., Ma, L., 2009. Gradient-preserving color transfer. *Comput. Graphics Forum*. <https://doi.org/10.1111/j.1467-8659.2009.01566.x>.
- Xiong, Y., Pulli, K., 2010. Color matching of image sequences with combined gamma and linear corrections, in: MM'10 -. In: Proceedings of the ACM Multimedia 2010 International Conference. <https://doi.org/10.1145/1873951.1873989>.
- Xu, W., Mulligan, J., 2010. Performance evaluation of color correction approaches for automatic multi-view image and video stitching, in. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/CVPR.2010.5540202>.
- Zhu, S., Zhang, R., Zhou, L., Shen, T., Fang, T., Tan, P., Quan, L., 2018. Very Large-Scale Global SM by Distributed Motion Averaging, in. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/CVPR.2018.00480>.