S & M 3877

Image Color Consistency Correction Method Based on Iterative Strategy and Quadratic Convex Optimization

Dinglong Yu,[1] Junxing Yang,[1*] Tianjiao Wang,[1] Qiaoye Yu,[2] He Huang,[1] and Fei Deng[3]

[1]School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture,
No. 15, Yongyuan Road, Huangcun Town, Daxing District, Beijing 102616, China
[2]Shijiazhuang Xibaipo Expressway Operation Center, Shijiazhuang, China
[3]School of Geodesy and Geomatics, Wuhan University, Wuhan, China

(Received April 18, 2024; accepted November 28, 2024)

Keywords: 3D reconstruction, quadratic convex optimization, color consistency correction, iterative strategy

Image-to-image color inconsistencies present a significant challenge in image-based 3D reconstruction, stemming from factors such as weather conditions, temporal variations, and exposure settings. Existing global color consistency correction algorithms exhibit limitations in simultaneously achieving effective color correction and efficiency, particularly in large-scale scenes. To address this issue, we propose an iterative strategy-based algorithm for multiview image color consistency correction. The algorithm initiates feature extraction and matching processes to establish point correspondences across images. For each set of correspondences, a priori color information is computed for each image. Subsequently, quadratic splines are computed by quadratic convex optimization for individual images. The raw pixel values are updated using these splines, and this iterative process is repeated to compute a series of quadratic splines until convergence. Simultaneously, gradient constraints are introduced to ensure that the image retains contrast. By applying these computed quadratic splines to transforming the images, we can achieve color consistency correction. Experimental results demonstrate that the proposed algorithm not only achieves higher quality than existing algorithms but also significantly improves operational efficiency by reducing the complexity of the algorithm. The inherent flexibility from the use of simultaneous splines facilitates the acquisition of a series of splines through iterative computations, further enhancing its color correction capability. This methodology can find broad applications in image stitching, image-based 3D reconstruction, and other related fields.

1. Introduction

Image-based 3D reconstruction has emerged as a crucial technique for creating realistic 3D models of real-world scenes. This methodology encompasses a series of steps, including feature detection and matching, motion-based structure reconstruction, dense reconstruction, surface reconstruction, and texture mapping, to produce 3D models. However, when multiview images

[*]Corresponding author: e-mail: yangjunxing@bucea.edu.cn

https://doi.org/10.18494/SAM5078

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

are captured under different conditions, such as varying times, weather conditions, and exposure settings, significant color variations may arise among the images. If image color differences are used directly in the texture mapping process, they can adversely impact the visual quality. Hence, addressing color disparities among images and achieving color consistency in 3D models are critical and formidable challenges.

Algorithms designed to address color consistency correction among images can generally be categorized into local fusion and global color consistency correction. A combination of these two approaches is commonly employed to achieve color consistency across images. Local fusion(1,2) techniques focus primarily on local color transitions between images to achieve seamless and visually pleasing transitions. On the other hand, global color consistency correction involves computing a global transformation function for the images, which adjusts their overall color tones and ensures consistent colors throughout the scene. The algorithms for global color correction can generally be categorized into three main types. The first type of color consistency correction algorithm is color style transfer, which typically involves two images.(3–6) Color style transfer occurs between two images. Recent advancements in deep learning-based style transfer algorithms(6) have demonstrated impressive results. Reinhard *et al.*(3) introduced a method of aligning the mean and variance of the target image with those of the reference image, performing adjustments in the lab color space to mitigate issues stemming from correlation among the red, green, and blue (RGB) channels. Vallet and Lelegard(5) employed partial iterations to symmetrize nonparametric color correction, facilitating the simultaneous adjustment of two images instead of using one image as a reference. Nguyen *et al.*(7) initially performed white balance operations on both images to rectify color cast issues arising from disparate lighting conditions. Subsequently, each image was aligned to a shared “white balance axis,” enabling gradient-preserving histogram matching along that axis to harmonize the tonal distribution between the two images. In the current landscape, machine-learning-based approaches that involve effectively combining algorithmic methods with prior knowledge to yield favorable outcomes have also emerged.(8–11) These methods apply only symmetric color correction to image pairs, necessitating iterative processing for managing multiview images in batch processing. The second type of algorithm is texture optimization, which focuses on correcting the color consistency of texture patches in the reconstructed model rather than directly correcting color inconsistencies in the input images.(12–15) Texture optimization algorithms focus on texture optimization, specifically emphasizing color consistency correction for texture patches within reconstructed models rather than directly addressing color inconsistencies in the input images. These methods critically rely on the accuracy of mesh vertices and the precision of image poses, as they directly affect the registration accuracy of the overlapping regions of texture patches. If significant misalignment occurs, satisfactory color consistency correction results may not be achieved. The third type of algorithm for color consistency correction is global color consistency correction. The general idea behind multiview image global color correction algorithms is to compute a global transformation for each image that minimizes differences between corresponding information across the images or the histograms of the images. Various models, including linear models,(16–18) gamma models,(19) matrix models,(20) and quadratic spline curves,(21–24) have been proposed for global transformations. These algorithms can achieve robust color consistency correction results on certain datasets, especially those utilizing quadratic spline curves as transformation models.(21–23) Several effective algorithms have been proposed in the field of image stitching and 3D reconstruction, most of which involve calculating a transformation matrix for each image by minimizing information or histogram differences between images to accomplish color transformation between them.(16,21,25–28) Moulon *et al.*(17) proposed a linear model-based global multiview color consistency correction approach, minimizing histogram differences between images using the same name points or lines as correspondences. Three channels of each image are then assigned a global registration model. Park *et al.*(19) addressed color consistency correction using scale-invariant feature transform (SIFT) feature matching and robust low-rank matrix factorization to estimate gamma model parameters. However, the inflexibility of the gamma model compared with the quadratic spline model(21) often results in an underestimation of albedo, particularly for rigid scenes. Despite improving the effectiveness of the algorithm, the efficiency decreases as the transformation model becomes more flexible, introducing more parameters and complicating optimization. To address this issue, in some algorithms, hierarchical optimization strategies are used to enhance efficiency.(22,23)

Color style transfer algorithms are specifically designed for pairs of images and are not suitable for achieving color consistency across large-scale multiview images. On the other hand, texture optimization algorithms are more versatile and applicable to a wider range of scenarios, but they require high image pose precision and mesh vertex accuracy. Global color consistency correction algorithms are effective at ensuring overall color tone consistency among images. However, these algorithms face challenges in balancing color consistency and computational efficiency. To address these limitations, we propose an iterative strategy for color consistency correction in multiview images, reducing computational complexity and significantly improving computational efficiency. Moreover, the iterative strategy enables the computation of a series of quadratic spline transformation functions for each image, providing a greater flexibility than when using a single transformation function with high image quality.

2. Method

We introduce a method to compute a global transformation for each image to minimize disparities in corresponding information among images, ensuring both contrast preservation and artifact-free images. A flowchart of the algorithm proposed in this article is depicted in Fig. 1.

2.1 Transformation function solving

In this article, we employ the average color value of the projected points on the visual image for each sparse point as the target color value. The method for calculating the projection of sparse points on an image follows Eq. (1):

$$x = F \cdot D\left(\prod\left(O \cdot R\left(X - C\right)\right)\right) + x_0, \tag{1}$$

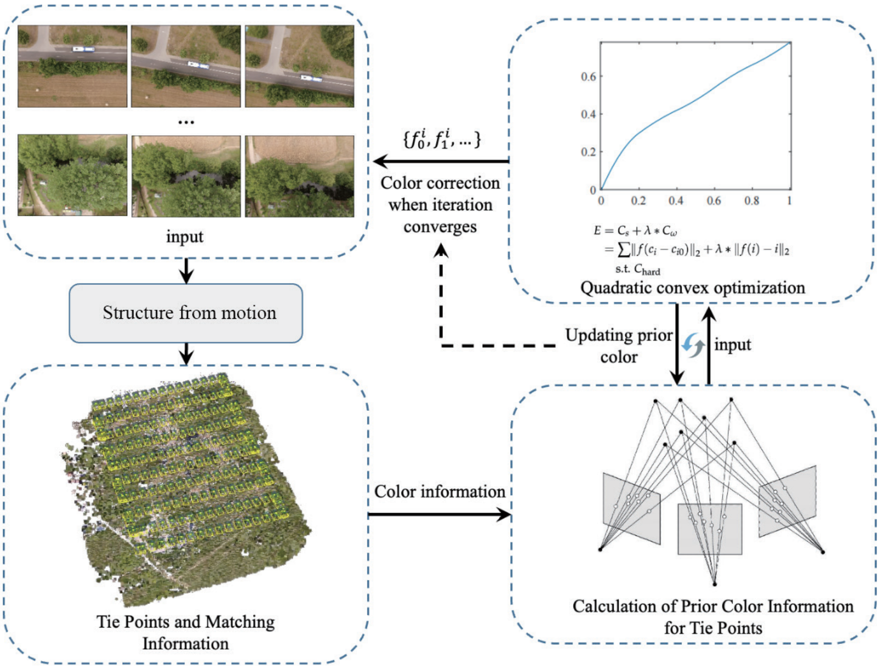

input
...
Structure from motion
Tie Points and Matching Information
Color information
Calculation of Prior Color Information for Tie Points
Updating prior color
input
$E = C_s + \lambda * C_\omega$
$= \sum \|f(c_i - c_{i0})\|_2 + \lambda * \|f(i) - i\|_2$
s.t. C_{hard}
Quadratic convex optimization
0
0.2
0.4
0.6
0.8
1
$\{f_0^i, f_1^i, \ldots\}$
Color correction when iteration converges

Fig. 1. (Color online) Overview of the proposed global color correction method.

where X represents the 3D coordinates of the sparse point, x represents the pixel coordinates of the sparse point on the image, x_0 is the offset, C represents the central coordinates of the image, R is the rotation matrix, F is the focal length, D is the image distortion function, and Π is the perspective projection function. However, it is important to note that the proposed algorithm can directly utilize matching point information for color consistency correction and does not require the geometric coordinates of the sparse points, as the algorithm is independent of geometric information.

There are two primary advantages of this approach: First, the optimization problem is decoupled, and each image can be optimized separately to greatly improve the efficiency of the algorithm. Second, the average value is a more representative target color value, which reduces the color cast problem of the image to a certain extent. Equation (2) is used to calculate the average color value of each sparse point.

$$c_{i0} = \frac{\sum c_i}{H}, \tag{2}$$

where c_i represents the color value of the sparse point in the i-th image and H is the number of images where the sparse point is projected. The average color value c_{i0} reduces the effect of individual image variations, such as lighting conditions or exposure differences, by averaging across all the images where the sparse point appears. This averaging helps ensure that the corrected color values are closer to the true, consistent color of the point across the scene.

To achieve consistent color results, our approach utilizes quadratic spline functions as the color transformation model, which provides greater flexibility in modeling color mappings.

Despite the increased number of parameters, the proposed iterative strategy ensures computational efficiency comparable to those of existing algorithms. Quadratic splines offer more adaptability than linear transformations, gamma transformations, S-curve transformations, and matrix transformations. The number of segments used in the spline determines its flexibility: higher-order polynomials allow for more adaptable transformations. To balance parameter count and functional flexibility, we utilize a five-segment quadratic spline curve in this study, where each segment is modeled as a quadratic polynomial. All color transformations are performed in the RGB color space, with each channel processed independently, resulting in 45 parameters per image.

$$f_i^k\left(x\right)=a_k+b_k\left(x-x_k\right)+c_k\left(x-x_k\right)^2,k=0,1,\ldots,n-1,x\in\left[0,1\right],x_k\in\left\{step*k\right\}_{k=0}^{n-1} \quad (3)$$

The parameter k represents the kth segment function, and a_k, b_k, and c_k are the parameters of the spline curve.

To minimize the discrepancy in correspondence information across images, Yang *et al.*(22) employed the variance of the color projections of each sparse point on different images as the loss function. However, a drawback of this approach is that the entire energy function incorporates the color transformation parameters of all images, making the optimization process challenging. The algorithmic complexity is $o(n^2)$, which hampers parallel acceleration and contributes to the lower efficiency of the algorithm. To address this issue, we first compute the mean value for each sparse point. This approach establishes a correspondence between the sparse point mean and the color values of the projected points on each image denoted by $Q=\left\{c_i,c_{i0}\right\},i\in\left[0,m-1\right]$, where m represents the number of sparse points corresponding to the images. On the basis of Q, we can calculate a quadratic spline curve as the color transformation function for each image. The cost function is defined as

$$\begin{aligned} E &= C_s+\lambda*C_\omega \\ &= \Sigma\left\|f\left(c_i-c_{i0}\right)\right\|_2+\lambda*\left\|f\left(i\right)-i\right\|_2 \text{ s.t. } C_{hard}, \end{aligned} \quad (4)$$

Here, C_s represents a term that induces the same-name information in the images to approach the average value after color correction. We use the second norm between the average values and the original color values for the loss function in this paper. C_ω is a regularization term that prevents the significant deviation of the color values from those in the original images after color correction. The parameter λ is a balancing parameter set to 0.001 in all the experiments, effectively balancing fidelity to the original color values and the regularization term. This choice yields good color correction results. C_{hard} represents the quadratic spline constraints, which include two types of constraint in this paper. The first type of constraint is the intrinsic constraint of the quadratic spline, which ensures first- and second-order continuities at the nodes of the spline function. The constraint can be formulated as shown in the first and second lines of Eq. (5). The second type of constraint preserves the contrast of the images and prevents contrast loss. In this paper, the gradients of the quadratic splines are constrained to control contrast preservation. The specific formulation is given in the third and fourth lines of Eq. (5). The parameter δ_{min} has a significant impact on the contrast of the images and has been heavily relied upon in previous algorithms such as that of Yang *et al.* However, in this paper, since the goal is to adjust the images to be closer to the average image, the reliance on this parameter is reduced. To facilitate comparison with other algorithms, we set δ_{min} to 1 and δ_{max} to a relatively large value, such as 5.

$$\begin{cases} f(x_i) = f(x_{i+1}) \\ \dfrac{df(x_i)}{dx} = \dfrac{df(x_{i+1})}{dx_{i+1}} \\ \dfrac{df(x_i)}{dx} \geq \delta_{min} \\ \dfrac{df(x_i)}{dx} \leq \delta_{max} \end{cases} \tag{5}$$

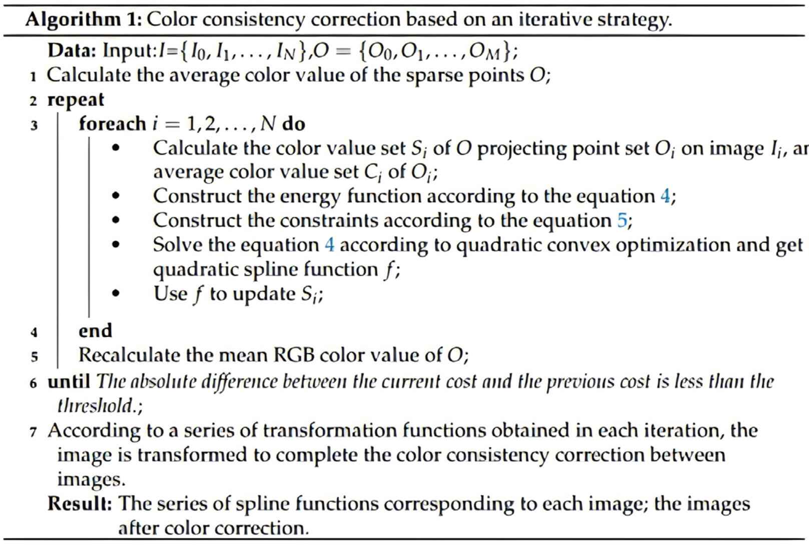
Equation (4) can be solved by using the quadratic programming tool QuadProg++[1], which implements a quadratic programming solver,[(29)] to obtain the fitted quadratic splines. Finally, by iterating the process, we obtain a series of quadratic splines, which are used to transform the images and to achieve color consistency correction across the images.

2.2 Iterative strategy

Although quadratic spline curves offer greater flexibility than linear or gamma curves and can yield satisfactory color consistency results on general datasets, we further enhance color consistency by employing an iterative strategy. The iterative approach enables the computation of a series of quadratic spline functions for each image, leading to improved color consistency correction across the images.

The iterative process begins by calculating the average color value of sparse feature points projected onto each image. Utilizing this average color value and the original values of the projected points, we compute quadratic spline functions for each image using the proposed method. These spline functions are subsequently applied to transform the images, and the average color value is updated accordingly. The process iterates until convergence is achieved, ensuring color consistency correction across the images. The primary objective of the iterative approach is to map shared information between images to the average value. By calculating each image independently, we can reduce the complexity of the optimization problem, enabling the possibility of employing parallel acceleration algorithms and enhancing computational efficiency. Moreover, the iterative process generates a series of quadratic spline transformation functions, enabling progressive refinement and superior color consistency correction compared with using a single function.

A specific algorithm flowchart is shown in Table 1. The average value of sparse points corresponding to image I_i is denoted by C_i, and the color value on image I_i corresponding to C_i is



```
Algorithm 1: Color consistency correction based on an iterative strategy.
  Data: Input:$I=\{I_0, I_1, \ldots, I_N\}$,$O = \{O_0, O_1, \ldots, O_M\}$;
  Calculate the average color value of the sparse points $O$;
  repeat
      foreach $i = 1, 2, \ldots, N$ do
          • Calculate the color value set $S_i$ of $O$ projecting point set $O_i$ on image $I_i$, and the
            average color value set $C_i$ of $O_i$;
          • Construct the energy function according to the equation 4;
          • Construct the constraints according to the equation 5;
          • Solve the equation 4 according to quadratic convex optimization and get the
            quadratic spline function $f$;
          • Use $f$ to update $S_i$;
      end
      Recalculate the mean RGB color value of $O$;
  until The absolute difference between the current cost and the previous cost is less than the
    threshold.;
  According to a series of transformation functions obtained in each iteration, the
    image is transformed to complete the color consistency correction between
    images.
  Result: The series of spline functions corresponding to each image; the images
          after color correction.
```

Table 1
Detailed information on the datasets.

	Small building	SCHOOL	GROUND	TJH6	Navona	Small building	HK-0001BX01	HK-0003FH
No. of images	36	55	50	55	92	13	188	1320
Size	4592*3058	4864*3648	4608*3456	4864*3648	4000*3000	3872*2592	5472*3648	6000*4000
Platform	UAV	UAV	Tbox	UAV	Handheld	Handheld	UAV	UAV
Sparse points	12.7K	5.4K	1.9K	89.5K	24.3K	5.3K	16.4K	459.30K

S_i. In the k-th iteration, through quadratic convex optimization, S_i and O_i are used to calculate the transformation function f_k^i of the image, and then f_k^i is used to update S_j. The other images also undergo the same operations simultaneously to obtain the updated S'_j and C'_j. This process is repeated to obtain a series of spline transformation functions f_k^i for each image until convergence. Finally, f_k^i is used to transform each image and complete the color consistency correction between the images. Because multiple transformations of images can be time-consuming, considering that the pixel values of each image are between 0 and 255, table lookup can be used to accelerate the transformation process.

To assess the performance of the iterative approach, we conducted an experimental analysis on the dataset, examining the relationship between the number of iterations and the loss. Figure 2 illustrates this relationship, demonstrating that convergence is typically achieved within 2–3 iterations, with a relatively rapid convergence rate for the majority of the dataset. This can be attributed to the high flexibility of quadratic spline functions, as even a single transformation can yield satisfactory color consistency results.

3. Results and Discussion

3.1 Datasets and settings

To comprehensively validate the effectiveness of our algorithm, we conducted experiments on various types of dataset, as shown in Table 1. The experimental setup involved a desktop

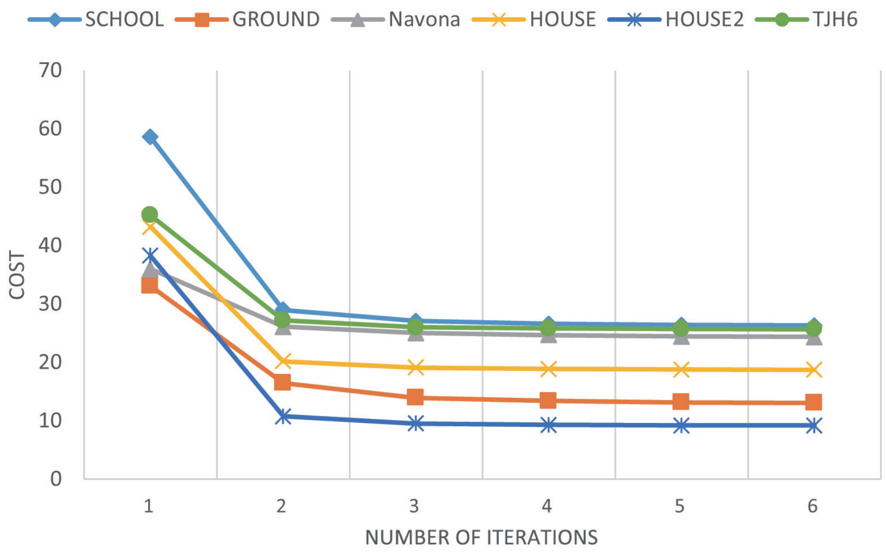


Fig. 2. (Color online) Relationship between the number of iterations and the cost on five datasets.

computer equipped with a 64-bit operating system, an i9-12900KF CPU running at 3.2 GHz, and 32 GB of RAM.

3.2 Efficiency analysis

In this subsection, we present an efficiency analysis of the proposed algorithm. The algorithm in this paper is based on the algorithm proposed by Yang *et al.*,(22) which is known for its efficiency compared with the algorithms of Moulon *et al.*(17) and Park *et al.*(19) A comparison is made with Yang *et al.*'s(22) algorithm only. Additionally, Yang *et al.* and other researchers have employed hierarchical optimization strategies to improve the efficiency of their algorithms on large-scale datasets. The goal of the algorithm in this paper is to enhance the efficiency of each layer of optimization rather than employing a hierarchical approach as in Yang *et al.*'s algorithm. To provide an objective analysis of algorithm complexity, we focus on analyzing the optimization part of the color correction algorithm and do not consider the time spent on I/O operations, as this is beyond our control. Table 2 presents a runtime comparison between Yang *et al.*'s algorithm and our algorithm on eight different datasets. Our algorithm takes significantly less time to run than Yang *et al.*'s algorithm, with optimization times typically in the range of a few seconds. The slower performance of Yang *et al.*'s algorithm can be attributed to optimizing all the color correction parameters together, requiring a substantial amount of time for energy function construction and solving. Overall, the complexity of their algorithm is on the order of $o(n^2)$, resulting in lower efficiency. On the other hand, our algorithm decouples the optimization process for each image, eliminating the issue of many parameters when optimizing all images together. Furthermore, our algorithm can be further accelerated by using parallel optimization strategies, contributing to its higher efficiency than Yang *et al.*'s algorithm.

The relationship between the efficiency of the proposed algorithm and the number of images is shown in Fig. 3. As the number of images increases, the runtime of our algorithm shows a

Table 2
Efficiency comparison analysis. Unit: seconds.

	Small building	SCHOOL	GROUND	TJH6	Navona	Small building	HK-0001BX01	HK-0003FH
Yang *et al.*	2.773	0.577	50.818	4.106	59.341	845.000	>1 hour	N/A
Ours	0.971	0.630	0.445	1.423	1.492	1.767	4.016	33.808

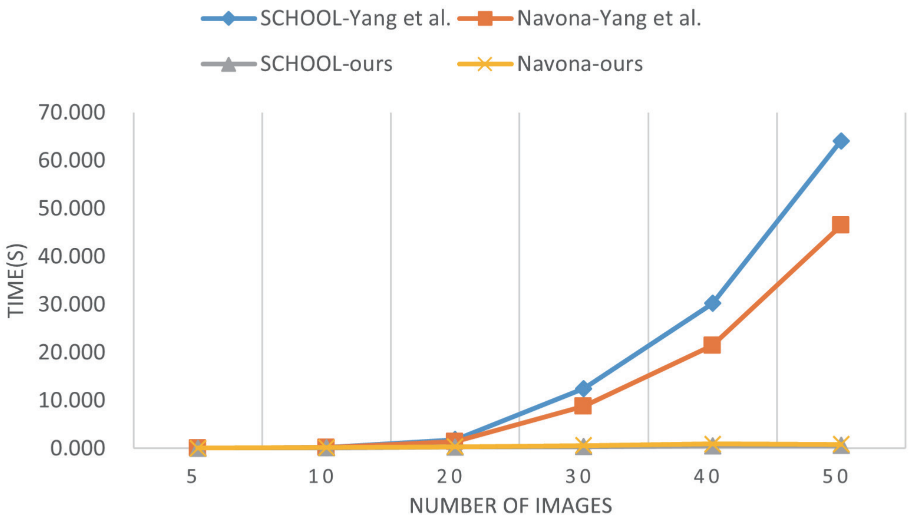


Fig. 3. (Color online) Comparative analysis of efficiency.

linear growth trend, whereas that of Yang *et al.*'s algorithm exhibits a parabolic trend. Moreover, the runtime of our algorithm is significantly lower than that of Yang *et al.*'s algorithm. From the curves, we observe that the complexity of our algorithm is $O(n)$, whereas Yang *et al.*'s algorithm has a complexity of $O(n^2)$ in each group. This difference is attributed to the decoupling of joint optimization among images in our algorithm, allowing for the separate optimization of each image and greatly enhancing efficiency. Although our algorithm achieved results visually comparable to those of Yang *et al.*'s algorithm, our algorithm has a significant advantage owing to its iterative approach, which provides a notable improvement in efficiency. This efficiency advantage is a key strength, enabling significantly faster computation while maintaining color consistency.

3.3 Performance on image stitching and texture mapping

To assess the effectiveness of the algorithm, we applied it to remote sensing image stitching and texture mapping. Figure 4 displays the results of orthophoto stitching. In cases where there is a color difference between images, significant color disparities can manifest at both ends of the stitching line. It is evident that remote sensing images, captured at different times and under varying lighting conditions, exhibit substantial color differences between stitched images, significantly impacting visual perception. After applying the algorithm proposed in this article for color correction, the color differences essentially disappear, even without using the local

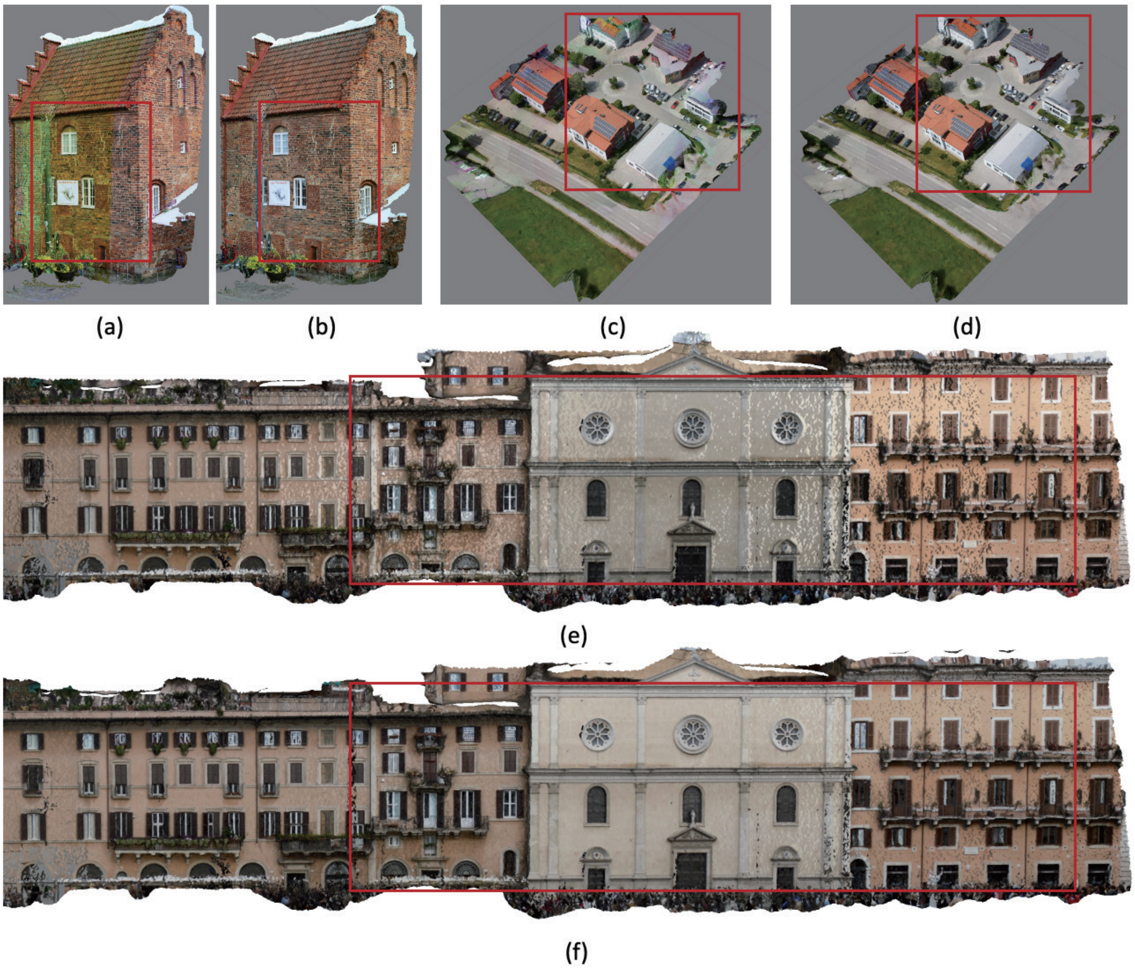


Fig. 4. (Color online) Performance in image stitching. (a) and (c) are the results without color correction, whereas (b) and (d) are the results corresponding to (a) and (c), respectively, after color correction based on the proposed method.

color transition algorithm. The effect on the real 3D model aligns with that on the orthophoto image, as shown in Fig. 5. This indicates that the algorithm proposed in this paper can achieve color consistency correction between sequential images by minimizing the difference in homonymous information in the images. As a result, a sequence image with color consistency is produced, thereby improving the perception of orthophoto images and real 3D models.

3.4 Comparative results

We conducted a comprehensive qualitative analysis of four algorithms using the SCHOOL and Small Building datasets to evaluate their efficacy in addressing color inconsistencies. Owing to space limitations, we provide additional experimental results here. To simulate significant color variations, we applied adjustments to the images using Photoshop. The SCHOOL dataset comprises images with varying exposure levels, mimicking images with

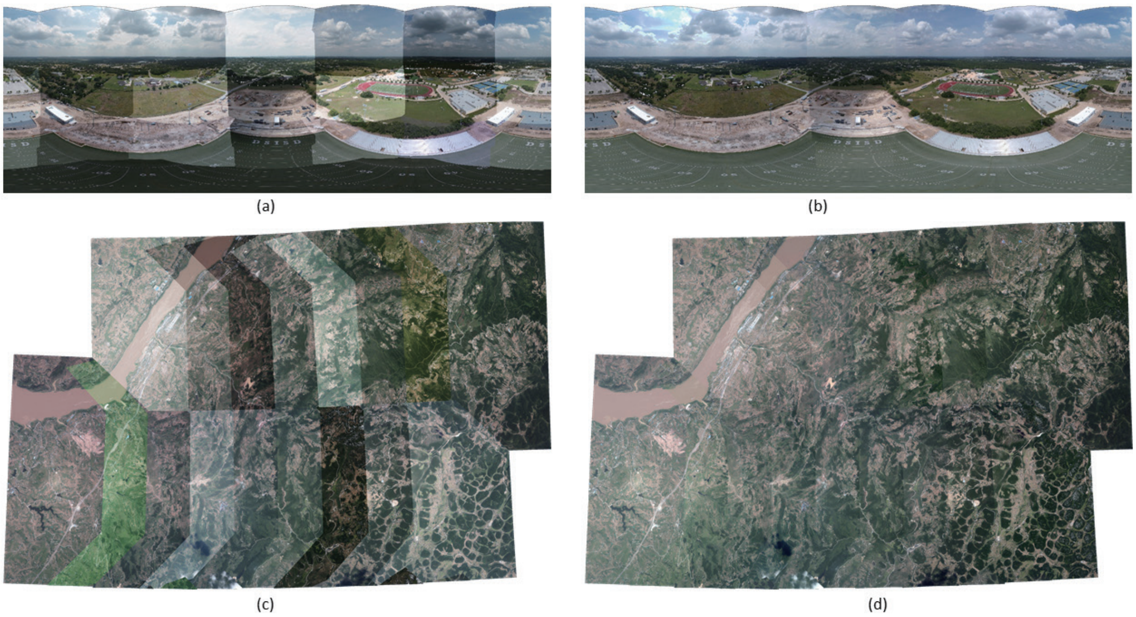


Fig. 5. (Color online) Performance in texture mapping. (a), (c), and (e) are the results without color correction, whereas (b), (d), and (f) are the results corresponding to (a), (c), and (e), respectively, after color correction based on the proposed method.

different exposures, whereas the Small Building dataset exhibits substantial color variations, representing images with inconsistent colors. Figure 6 shows the results of the four methods on the Small Building dataset. Upon careful examination, it is evident that Moulon *et al.*'s algorithm performed the poorest. Although it achieved satisfactory color consistency, it suffered from severe loss of contrast. This drawback stems from the algorithm's utilization of linear transformation functions with small slopes to enhance color consistency, resulting in a considerable loss of contrast due to the pronounced color differences within the Small Building dataset. In comparison, Park *et al.*'s algorithm yielded better outcomes than Moulon *et al.*'s algorithm by preserving contrast without significant loss. This improvement can be attributed to the implementation of a more flexible gamma function. However, notable color disparities were still present in the first and third images compared with the remaining images, suggesting that the flexibility of the gamma function may be inadequate or that the algorithm may lack robustness in managing complex color inconsistencies. In contrast, both Yang *et al.*'s algorithm and our algorithm achieved consistent and superior results compared with the previous two algorithms in terms of color consistency and contrast preservation. This was due to the use of highly flexible quadratic spline functions in both algorithms, which can manage larger color variations.

Similarly, on the SCHOOL dataset (Fig. 6), the results of the four algorithms align with those on the Small Building dataset. Moulon *et al.*'s algorithm performed the worst, with significant contrast loss, followed by Park *et al.*'s algorithm. Yang *et al.*'s algorithm and our algorithm achieved the best results in terms of both color consistency correction and contrast preservation. This demonstrates that our algorithm can address complex color inconsistency issues and

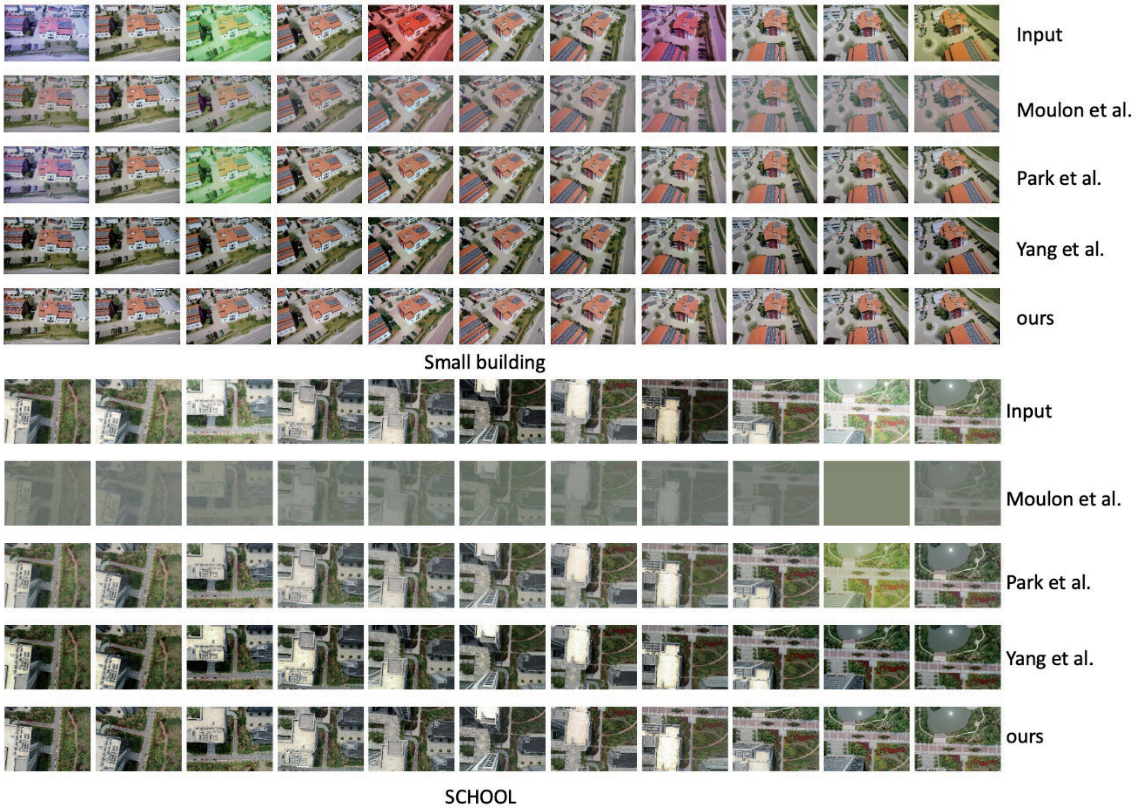


Fig. 6. (Color online) Comparison of the results for the Small Building dataset. First row: original images; second row: results from Moulon *et al*.'s algorithm; third row: results from Park *et al*.'s algorithm; fourth row: results from Yang *et al*.'s algorithm; fifth row: results from our algorithm.

achieve good results in terms of exposure and color aspects. Moreover, our algorithm demonstrates better efficiency and quality than Yang *et al.*'s algorithm, which will be discussed later.

To verify the capability of the algorithm to correct color consistency, in this study, experiments were conducted on five datasets$^{(29–32)}$ with significant color differences, and the results are compared with those of Yang *et al.*'s algorithm. The results on three internet datasets are shown in Fig. 7 for comparison, revealing substantial color differences due to the images being captured by different people at different times and using different devices. The algorithm results of Yang *et al.* and our results both exhibit good color consistency, albeit with differences in color tones. Figure 6 displays two sets of images captured using single-lens reflectors, and the results are consistent with those in Fig. 7. The good color consistency indicates the strong color correction capabilities of both algorithms, given their utilization of flexible spline curves as transformation functions. The inconsistent color tones suggest that the two algorithms adjust colors in different directions. Yang *et al.*'s algorithm, without any related constraints, selects a hue that minimizes cost as a result. In contrast, our algorithm employs an iterative strategy to make the color tones of the image tend towards an average, aligning more with human visual perception. Tone inconsistency is a common problem in color consistency correction algorithms,

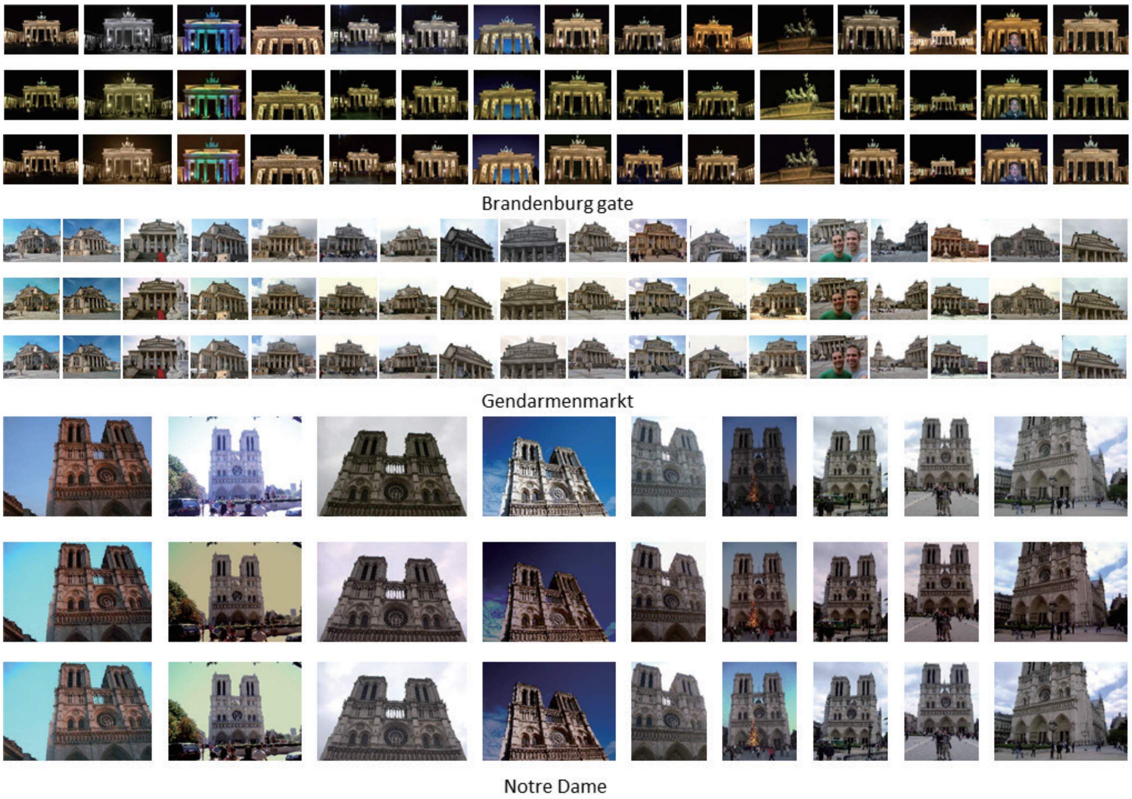


Fig. 7. (Color online) Comparison of the results for the internet dataset. First row: original images; second row: results from Yang *et al.*'s algorithm; third row: results from our study.

and addressing this issue is an important direction for improvement. These experiments demonstrate that our algorithm possesses color correction capability comparable to that of Yang *et al.*'s algorithm, with significantly higher efficiency. This efficiency advantage is a key strength of our algorithm. Furthermore, a quantitative evaluation reveals that the algorithm proposed in this paper exhibits greater stability in preserving image quality than Yang *et al.*'s algorithm, as detailed in the subsequent sections.

We also conducted a quantitative evaluation of the four algorithms using the PSNR and SSIM metrics on the internet dataset. PSNR represents the similarity to the original images, with higher values indicating better results. SSIM measures the preservation of image structure, where higher values indicate better preservation (with a maximum value of 1). Table 3 presents the experimental results. Our algorithm achieved the best results in terms of PSNR, indicating the superior preservation of the original image information. Although the color consistency results of our algorithm are visually similar to those of Yang *et al.*'s algorithm, our algorithm outperformed it in terms of PSNR and SSIM. The reason behind this lies in the fact that our algorithm transforms the images in such a way that the corresponding information converges to the average value, which is theoretically reasonable. On the other hand, Yang *et al.*'s algorithm lacks a target solution for color correction after transformation and relies on a larger search space to find the optimal color consistency correction solution. This may result in significant

Table 3
Detailed PSNR and SSIM results of four methods on eight datasets.

	Small building		SCHOOL		Small building2		GROUND		TJH6		Navona		HK-0001BX01	
PSNR-SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Moulon *et al.*	24.338	0.897	22.955	0.841	20.798	0.924	21.935	0.895	16.682	0.836	13.663	0.739	25.251	0.948
Park *et al.*	21.564	0.937	21.062	0.890	25.472	0.962	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Yang *et al.*	20.931	0.879	18.343	0.819	25.164	0.936	23.186	0.897	24.794	0.949	24.937	0.924	30.232	0.983
ours	25.087	0.925	23.362	0.919	26.530	0.947	24.871	0.922	25.196	0.945	27.502	0.945	32.782	0.989

deviations from the original images and potential color shifts. Although there is a regularization term to encourage the preservation of original image information, its capability is limited. In terms of SSIM values, Park *et al.*'s algorithm achieved good results, followed by our algorithm. This can be attributed to the color modeling approach used in Park *et al.*'s algorithm. Additionally, our algorithm performed well in preserving details owing to the constraint on the gradients of the quadratic spline curve ensuring that they are greater than 1. This approach outperformed the algorithms of Moulon *et al.* and Yang *et al.* once again, demonstrating that our algorithm achieves overall improvements in preserving image information.

4. Conclusion

Our proposed iterative-based approach presents an efficient and effective solution for color consistency correction in multiview images, notably enhancing the visual perception of 3D reconstruction models. By decoupling the joint optimization of color transformation function parameters and the optimization of each image's parameters individually, we developed an algorithm with a notable reduction in complexity without resorting to hierarchical optimization. Furthermore, our algorithm demonstrates superior performance in both efficiency and color consistency correction compared with existing methods, leveraging multiple flexible quadratic spline transformation functions. While ensuring streamlined processing, our algorithm not only enhances color consistency but also attains optimal results in quantitative evaluations. The incorporation of gradient constraints ensures that the images do not suffer from contrast loss.

However, some limitations should be addressed in future work. First, color cast (color shift) may occur when there are significant color differences among images. This can result in unnatural colors, where one or more images might be overcorrected or undercorrected, causing an overall color imbalance. Additionally, the algorithm may face insufficient color consistency across multiple views, especially in extreme cases where the color differences are beyond the algorithm's correction capabilities. This can lead to visible discrepancies in the final multiview image set, especially along the edges where images are stitched together.

To mitigate these issues, in future work, several potential improvements should be explored. Local area optimization may be applied to regions with large color differences, ensuring more natural transitions without global correction affecting smaller areas. Additionally, an adaptive color correction method may be introduced, where correction parameters are automatically adjusted on the basis of the degree of color variation between images. These enhancements will help the algorithm better handle cases of extreme color variation, ultimately improving both the visual and quantitative outcomes of multiview image correction.

Funding

This research was funded by the National Natural Science Foundation of China (Grant Number 42201483), the China Postdoctoral Science Foundation (Grant Number 2022M710332), and Funding for Postdoctoral Research Activities in Beijing (2023-zz-140).

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Funding

This research was funded by the National Natural Science Foundation of China (Grant Number 42201483), the China Postdoctoral Science Foundation (Grant Number 2022M710332), and Funding for Postdoctoral Research Activities in Beijing (2023-zz-140).

Data Availability Statement

Not applicable.

Acknowledgments

The authors express their gratitude to Pei Tao and Jian Lu for their assistance in our experiments.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

1 P. Pérez, M. Gangnet, and A. Blake: Proc. ACM SIGGRAPH 2003 Papers (SIGGRAPH, 2003) 313–318. https://doi.org/10.1145/1201775.882269.

2 P. J. Burt and E. H. Adelson: ACM Trans. Graph. **2** (1983) 217. https://dl.acm.org/doi/pdf/10.1145/245.247.

3 E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley: IEEE Comput. Graph. Appl. (2001). https://doi.org/10.1109/38.946629.

4 X. Xiao and L. Ma: Comput. Graph. Forum (2009). https://doi.org/10.1111/j.1467-8659.2009.01566.x.

5 B. Vallet and L. Lelégard: ISPRS J. Photogramm. Remote Sens. (2013). https://doi.org/10.1016/j.isprsjprs.2013.05.005.

6 Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song: IEEE Trans. Vis. Comput. Graph. **26** (2019) 3365. https://doi.org/10.1109/TVCG.2019.2921336.

7 R. M. Nguyen, S. J. Kim, and M. S. Brown: Comput. Graph. Forum (2014). https://doi.org/10.1111/cgf.12500.

8 B. Wang, Y. Yu, and Y.-Q. Xu: ACM Trans. Graph. (2011). https://doi.org/10.1145/1964921.1964959.

9 L. A. Gatys, A. S. Ecker, and M. Bethge: Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR, 2016) 2414–2423. https://openaccess.thecvf.com/content_cvpr_2016/html/Gatys_Image_Style_Transfer_CVPR_2016_paper.html

10 F. Luan, S. Paris, E. Shechtman, and K. Bala: Proc. 2017 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR, 2017) 4990–4998. https://openaccess.thecvf.com/content_cvpr_2017/html/Luan_Deep_Photo_Style_CVPR_2017_paper.html

11 K. Hong, S. Jeon, H. Yang, J. Fu, and H. Byun: Proc. IEEE/CVF Int. Conf. Comput. Vis. (2021) 14609–14617. https://openaccess.thecvf.com/content/ICCV2021/html/Hong_Domain-Aware_Universal_Style_Transfer_ICCV_2021_paper.html.

12 M. Kuse and S. P. Jaiswal: Proc. Int. Conf. Image Process. (ICIP, 2015) 1394–1398. https://doi.org/10.1109/ICIP.2015.7351029.

13 M. Waechter, N. Moehrle, and M. Goesele: Proc. Comput. Vis. ECCV 2014: 13th European Conf. Comput. Vis., Part IV (Springer, 2014) 836–850. https://doi.org/10.1007/978-3-319-10602-1_54.

14 V. Lempitsky and D. Ivanov: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2007). https://doi.org/10.1109/CVPR.2007.383078.

15 Y. Fu, Q. Yan, J. Liao, and C. Xiao: Proc. 2020 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR, 2020). https://doi.org/10.1109/cvpr42600.2020.00599.

16 M. Brown and D. Lowe: Int. J. Comput. Vis. **74** (2007) 59. https://doi.org/10.1007/s11263-006-0002-3.

17 P. Moulon, B. Duisit, and P. Monasse: Global Multiple-View Color Consistency (2013). https://enpc.hal.science/hal-00873517/.

18 T. Shen, J. Wang, T. Fang, S. Zhu, and L. Quan: Proc. Asian Conf. Comput. Vis. (ACCV 2017), Lecture Notes in Computer Science, vol. 10111 (Springer, 2017) 392–407. https://doi.org/10.1007/978-3-319-54190-7_24.

19 J. Park, Y.-W. Tai, S. N. Sinha, and I. S. Kweon: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR, 2016) 430–438. https://doi.org/10.1109/CVPR.2016.53.

20 Y. Li, H. Yin, J. Yao, H. Wang, and L. Li: ISPRS J. Photogramm. Remote Sens. **190** (2022) 1. https://doi.org/10.1016/j.isprsjprs.2022.05.009.

21 Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski: ACM Trans. Graph. **32** (2013) 1. https://doi.org/10.1145/2461912.2461997.

22 J. Yang, L. Liu, J. Xu, Y. Wang, and F. Deng: ISPRS J. Photogramm. Remote Sens. **173** (2021) 209. https://doi.org/10.1016/j.isprsjprs.2020.12.011.

23 Y. Li, L. Li, J. Yao, M. Xia, and H. Wang: IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **15** (2022) 4941. https://doi.org/10.1109/JSTARS.2022.3183188.

24 M. Xia, J. Yao, and Z. Gao: ISPRS J. Photogramm. Remote Sens. **157** (2019) 188. https://doi.org/10.1016/j.isprsjprs.2019.09.004.

25 Y. Xiong and K. Pulli: Proc. ACM Multimedia 2010 Int. Conf. (2010). https://doi.org/10.1145/1873951.1873989.

26 W. Wei, S. Zhanfeng, L. Junli, Y. Haiping, and L. Jiancheng: J. Surv. Mapping (2013) 6.

27 H. Cui, L. Zhang, H. Ai, B. Xu, and Z. Wang: J. Surv. Mapping **46** (2017) 12. https://doi.org/10.11947/j.AGCS.2017.20170402.

28 D. Goldfarb and A. Idnani: Math. Program. **27** (1983) 1. https://doi.org/10.1007/BF02591962.

29 K. Wilson and N. Snavely: Proc. Comput. Vis. ECCV 2014: 13th European Conf. (2014) 61–75. https://doi.org/10.1007/978-3-319-10578-9_5

30 O. Enqvist, F. Kahl, and C. Olsson: Proc. IEEE Int. Conf. Comput. Vis. Workshops (2011) 264–271. https://doi.org/10.1109/ICCVW.2011.6130252.

31 J. Heinly, E. Dunn, and J.-M. Frahm: Proc. Comput. Vis. ECCV 2012: 13th European Conf. (2012) 759–773. https://doi.org/10.1007/978-3-642-33709-3_54.

32 A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun: ACM Trans. Graph. **36** (2017) 1. https://doi.org/10.1145/3072959.3073599.

About the Authors

Dinglong Yu graduated from Beijing University of Civil Engineering and Architecture, China, in 2022 and is currently pursuing his master's degree at the same university. His research focuses on deep learning and 3D reconstruction. (1628094687@qq.com)

Junxing Yang graduated from the School of Surveying and Mapping of Wuhan University, China, in December 2021 with a Ph.D. degree in engineering. He is mainly engaged in research on 3D reconstruction, computer vision, autonomous driving, image stitching, and other related fields. (yangjunxing@bucea.edu.cn)

Tianjiao Wang is currently pursuing his master's degree at the School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture. His research interests are photogrammetry and remote sensing, 3D reconstruction, image stitching, and scene understanding.

Qiaoye Yu works at the Shijiazhuang Xibaipo Expressway Operation Center, Shijiazhuang, China, and her research field is in remote sensing science and technology.
(1840670339@qq.com)

He Huang received his B.S. degree from Wuhan University, China, in 2000 and his M.S. and Ph.D. degrees from Sungkyunkwan University, South Korea, in 2004 and 2010, respectively. Since 2010, he has been a lecturer and associate professor at the Beijing University of Civil Engineering and Architecture, China. His research interests are in autonomous driving, high-precision navigation maps, and visual navigation and positioning. (huanghe@bucea.edu.cn)

Fei Deng obtained his Ph.D. degree in engineering from Wuhan University, China, in 2006, specializing in photogrammetry and remote sensing, computer vision, 3D reconstruction, point cloud processing, and other related fields.