Optimization of Sparse Color Correspondences for Color Mapping

SHEIKH FARIDUL Hasan[∞], Jurgen STAUDER[*], Alain TREMEAU[∞]; Technicolor R&D France[*], Université Jean Monnet, France[∞]*

Abstract

This article addresses the problem of finding corresponding colors between multiple views of a same scene in order to compensate color differences by color mapping. Both, the dense and the sparse feature matching are studied in the literature to achieve those corresponding colors. However both methods suffer from spatial precision and occlusion. Moreover, in case of sparse feature matching, the spatial and the color space coverage are low. Therefore, it is difficult to generalize for colors where direct color correspondences are not known. Though dense feature matching may address this problem, it needs computational effort and may introduce additional occlusion errors. Therefore, in this work, we propose to consider the spatial neighborhood around sparse feature matching to select the "stable" corresponding colors. We estimate a color mapping model from the color correspondences which is able to compensate the color differences between the views. We compared the quality of several color mapping methods in a performance evaluation framework. From experimental results, we found that consideration of neighborhood can significantly increase the precision of color mapping in spite of increasing uncertainty of correspondence. Benchmark tests show good performance compared to recent methods from the literature.

Introduction

Applications using multiple views of the same scene often suffer from color differences between different views. These color differences may be caused, for example, from non-calibrated cameras, non-calibrated film scanners, inconsistent color corrections in post-production and physical light effects in the scene. These color differences between views can be compensated by so-called color mapping methods.

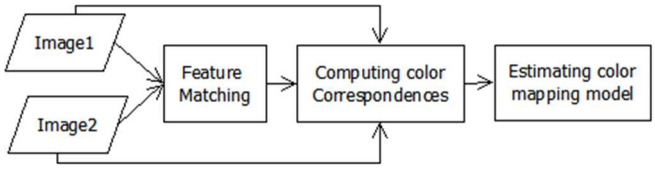

Figure 1. Main steps of color mapping

Color mapping is composed into three main steps as shown in Figure 1. It starts with finding the geometric relationship between the views using feature matching. From these findings of feature matching, the relationship of colors between different views is established, called *color correspondences*. These color correspondences tell us which colors from one view are corresponding with which colors from another view. Then, an appropriate color mapping model is chosen depending on the knowledge of how the colors are changed between the views. Finally, the color mapping model is fitted to the color correspondences by an estimation procedure.

Color correspondences, the first step of color mapping, usually extracts corresponding colors by utilizing the characteristics of the matched features. Indeed, the computation of color correspondences have a series of requirements:

- The spatial precision of feature matching in the views and thus the precision of color correspondences should be high. Here, precision refers to the feature's geometric parameters such as feature location, feature scale and feature orientation;
- Color correspondences should be robust against occlusion within the neighborhood of features;
- Outliers (wrongly matched features) should be as few as possible;
- Color correspondence should represent the scene colors sufficiently so that color changes can be generalized for colors where direct color correspondences are not known.

The requirement for choosing an appropriate color mapping model, the second step of color mapping, is that it should be able to describe the underlying actual color changes between views. Note that, the color mapping models that are typically used can be classified into parametric and non-parametric models.

Color mapping model parameter estimation, the third step of color mapping, needs to deal with limited precision and outliers in color correspondences.

In the literature, to achieve color correspondences, both sparse [1] [7] [14] [19] [20] [4] [10] and dense [6] [1] feature matching are explored. Sparse feature matching may not represent the scene colors sufficiently. On the other hand, though dense feature matching may represent the scene colors better than sparse, it needs high computational effort and may introduce more errors due to occlusion.

In our work, we want to enhance the color correspondences while keeping the other steps of color mapping constant. We choose SIFT feature correspondences[11] and a simple, parametric color mapping model. We also restrict our work to two views.

In this work, we address the following problems that have not yet been solved in the literature. First, known methods[19] that use sparse features extract color correspondences simply from the image colors of the matched features points . The precision of color correspondences is thus highly dependent on the spatial precision of the matched features. A second problem is that sparse features do not represent the scene's colors sufficiently. Finally, the third problem is the lack of quantitative quality evaluation. Often, state of the art results are reported as images only and evaluation is very often subjective.

To address the first and second problem, we propose to optimize sparse color correspondences extracted from sparse features by analyzing the spatial neighborhood of the features. The idea is to select representative colors in the spatial neighborhood of each feature point and to match these colors to their corresponding colors in the other view. To address the third problem, we will develop a quality evaluation framework which allows quantitative comparison of color mapping methods. Our main contributions are as follows:

- A color mapping method that exploits the spatial neighborhood of sparse features;
- A color mapping quality evaluation framework.

In order to achieve and proof these contributions, we limit our scope to global color mapping. Here, global refers to a single color mapping model which maps all colors irrespective of their spatial location in the view. However, we think that such results can be extended to local color mapping as well.

Review of color mapping methods

Color mapping is not the only family of methods that can analyze and compensate color differences between multiple views of a same scene. Table 1 shows that existing methods can be broadly classified into three main categories: color mapping, color transfer and user assisted methods. In the first category, color mapping methods, our focus, begin by feature matching as shown in Figure 1. Then, image colors of the matched features build the color correspondences. Finally, the color correspondences are fitted to a "chosen" color mapping model by an estimation procedure[13].

In the second category, color transfer methods are based on statistical distribution transfer such as histogram matching [5]. Here, precise geometric correspondences are not required and the views to be compensated may even not show the same scene but just similar scenes. Color transfer methods are reported to be suitable for applications such as artistic color change [16], automatic color grading [15] and grayscale image colorization by example [18].

In the third category, user assisted methods are based on manual human supervision. These methods are suitable for grayscale image colorization[3] which usually are employed when automatic mapping methods fail.

Table 1. Classification of color compensation methods

	Color compensation method		
Category	Color mapping	Color transfer	User assisted
Key method	Feature matching	Statistical distribution transfer	Human
Articles	[1] [6] [7] [19] [20] [4] [21][14]	[18] [5] [15] [16]	[3]

Some existing color mapping methods

Hacohen et al. [6] uses dense feature matching along with a color mapping model of piecewise cubic spline having seven breaks. Tehrani et al. [19] as well as Yamamoto and Oi [21] use global, data-driven, look up table based color mapping model. Wang et al. [20] propose a local, region-based color mapping model with just a simple, constant color coordinate offset per region. Chen et al. [1] uses disparity to have dense feature matching on the rank-transformed domain to attain color correspondence where they use a histogram-offset based color mapping model. Doutre et al. [4] use block-based disparity to yield dense matched features. They extract color correspondences by averaging colors and use a linear color mapping model. Kagarlitsky et al. [9] addresses the problem where no single color mapping exists for the whole view such as images taken under different illumination. They use SIFT [11] for color correspondences and use histogram matching on simultaneously co-segmented images. Oliveira et al. [14] color segment [2] the images and assumes that a coarse registration between views gives correspondences. Finally, they apply color transfer [16] to each of the regions separately.

Some existing color transfer methods

As noted before, color transfer methods are based on statistical distribution transfer such as histogram matching [5]. Therefore, feature matching is not required in color transfer methods. For example, Reinherd [16] propose one of the very first color transfer method that matches mean and variance of colors for each channel separately.

Pitie at al.[15] propose an iterative color transfer method that can transform any N-dimensional probability density function into another one. If the color dynamics of the views are very different, this method may create artifacts which may be removed by a post-processing step. Since this work [15] is one of the most successful color transfer method available in literature, we have included it in our experimental evaluation.

Optimization of sparse color correspondence

This section describes a new method for the computation of color correspondences from sparse features. Computation of color correspondences is understood to be the first step of color mapping.

We perform color mapping between two views. The first view - called *test* view – will be compensated by color mapping. We expect that the compensated colors are as close as possible to the second view – called *reference* view. For example, Figure 2a shows a sample of reference and test view. This example shows two stereo views (two viewpoints) with two different exposures. Here, the reference view is lighter whereas the test view is darker.

In this work, we use a scale and rotation invariant feature matching algorithm called SIFT [11] which matches features and their spatial neighborhood – called patch – and provide features' deformation parameters (location, scale and orientation). We call the patch coming from the *reference* view reference patch and its corresponding patch coming from the *test* view test patch.

Figure 2b shows an example of reference and test patches where the patch sizes are magnified by two for visualization purpose. Visually, Figure 2b seems to be a good match. However, it still suffers from location errors - shown by bottom left ellipses of Figure 2b, orientation errors - shown by top left ellipses of Figure 2b and occlusion errors - shown by right ellipses of Figure 2b. Since, we cannot expect high spatial precision for matched features, we propose a method that can estimate which colors from

the left patch correspond to which colors of the right patch. We call this method the NEIGHBORHOOD method.

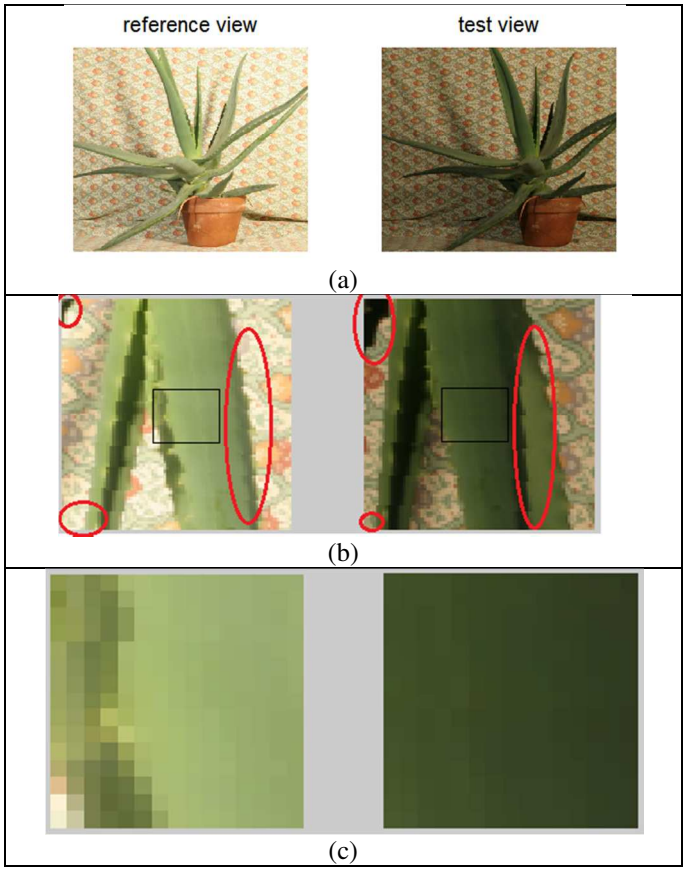

Figure 2. (a) Example of two views of a same scene),(b) close-up of the views showing a matched feature using SIFT [11] with a challenging occlusion errors (red ellipses),(c) further close-up of the same feature and its neighborhood - it can be seen from (b) and (c) that though it seems to be a good SIFT match, neither the feature location nor the colors in its neighborhood match well in terms of color correspondence.

The NEIGHBORHOOD method starts with selecting a spatial neighborhood (Figure 2b) – a patch - around the feature point. The image colors inside those patches are shown in Figure 2c. Neighborhood size is determined from the feature's scale parameter. To limit occlusion errors, we select 15x15 as maximum neighborhood size for images of size 437x370. To find corresponding colors from the selected neighborhood, our method applies color clustering / color segmentation. Both, the reference and the test patches are segmented into regions with constant color using the mean-shift algorithm [2]. Figure 3a shows an example where segmentation splits both, the reference and the test patches into clusters. Then, for each cluster of the reference patch, we first search a corresponding cluster in the test patch. Next, we compute the geometrically common area between two corresponding clusters. If the common area is large enough, these corresponding clusters are considered as good candidate for a color correspondence. Figure 4 shows the pseudo code of the proposed NEIGHBORHOOD method. In the next two subsections we explain the method in details.

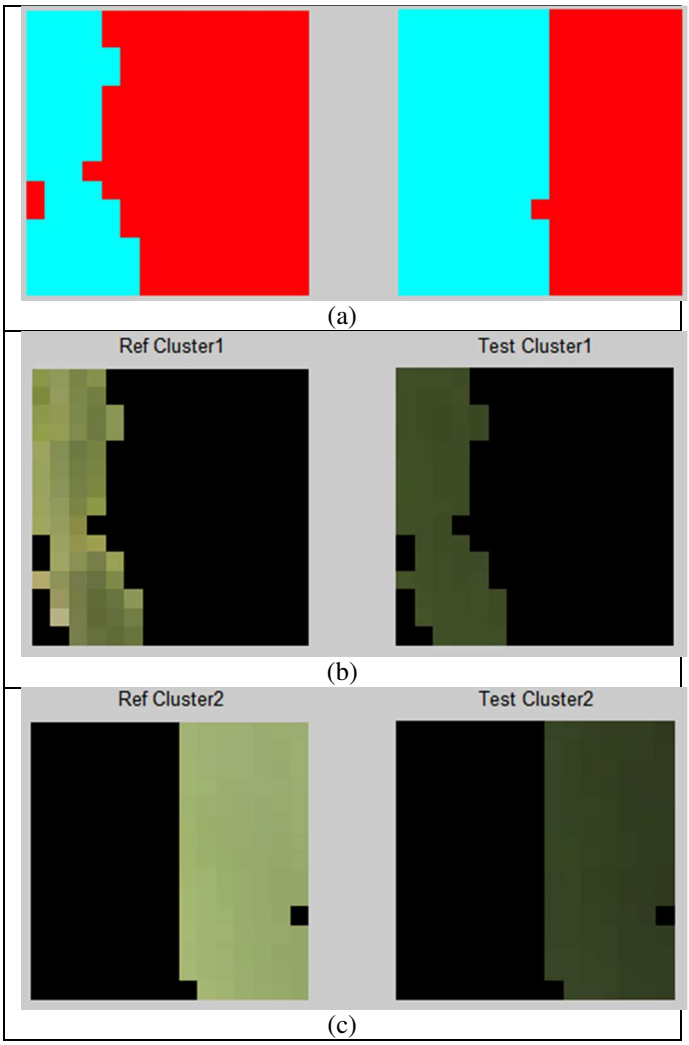

Figure 3. (a) Color clustering result of the patches shown in Figure 2(c) using mean-shift algorithm[2]. (b) and (c) shows the result of two cluster correspondence respectively having a set of selected colors (within μ±2σ) from that cluster correspondence.

Input: *ref* and *test* patch
Output: color correspondences

Step 1. Do color clustering in the *ref* and the *test*
Step2: For each cluster of the *ref* patch do
- I. Find the corresponding cluster in the *test* patch.
- II. Find geometrically common area between the corresponding clusters.
- III. if common area > threshold

set of colors in ref and test which satisfies:
$\mu-2\sigma<\{colors\}<\mu+2\sigma$ (1)
{colors} is a color correspondences
else
Go to step II and continue;

Figure 4: Proposed NEIGHBORHOOD color correspondences method; μ=mean, σ=standard deviation of {colors}

Finding corresponding clusters

In the following, we discuss how to find corresponding clusters given two color patches (Figure 2a) and its associated color clusters (Figure 3a). The color clusters comes with numbered labels. Next, we have to find out which label of the reference patch corresponds to which label of the test patch. Note that, the total number of clusters in the reference patch and the total number of clusters in the test patch may not be the same.

In a first step, for each cluster of the reference patch, we extract all corresponding pixel positions from the test patch.

In a second step, we identify the cluster of the test patch that has the largest overlap with these corresponding pixel positions. We call this overlap the "corresponding" cluster.

Finally, we identify the "corresponding" cluster by counting the statistical mode of the labels found in the overlap.

Matching colors between the corresponding clusters

From the previous section, we know which cluster of the reference patch corresponds to which cluster of the test patch. In this section, we first discuss whether these corresponding clusters are "good candidates" for color correspondences or not. Next, we discuss how to produce color correspondences from "good candidates". As noted before, we face couple of challenges in this task:

- occlusion,
- bad alignment of *reference* and *test* patch,
- incoherent color cluster size and number due to two independent clustering operations.

To solve these issues, we first assume that a good candidate for color matching corresponds to a cluster for which the common area between clusters should be at least equal to or greater than 50% of the smallest of the two clusters. If two corresponding clusters satisfies that condition, we assume that the corresponding regions are a "good candidate" for color correspondence.

From these "good candidates" of cluster correspondence, we have to produce the color correspondences. We therefore assume that two corresponding clusters follow a normal color distribution. For each good candidate and for each color channel, we select the colors where

$$\mu - 2\sigma \leq color\ coordinate \leq \mu + 2\sigma \qquad (1)$$

Here, μ is the mean and σ is the standard deviation of color coordinates per channel. Inside a cluster, if a pixel color satisfies this equation for all the channels, we call it a "stable" color. In other words, "stable" colors follow a normal distribution. Finally, all these selected "stable" colors from all patches give a list of color correspondences for that particular *reference* and *test* view.

Robust estimation of color mapping model

From the previous section, we have a list of color correspondences. In this section, we discuss - given a list of color correspondences and a color mapping model - how to robustly estimate the color mapping model's parameter. We choose a simple three parameter, non-linear color mapping model as shown by Equation 2:

$$C_{ref} = G\{C_{test}\}^{\gamma} + b \qquad (2)$$

Here, C_{ref} is a color coordinates of the reference view and C_{test} is a color coordinate of the test view . Parameter G defines the gain, γ the gamma and b the offset of the nonlinear function used. This function which usually is called GOG (Gamma, Offset and Gain) is our chosen color mapping model. The model is simple, parametric and channel-wise. As stated, this simple choice is justified in order to limit the experimental effort. We think that our proposed NEIGHBORHOOD method to obtain color correspondences is valid for a wider range of models.

In the following, we explain how to robustly estimate the parameters of this color mapping model. An example of the red color channel is shown in Figure 5. The figure shows the corresponding red coordinates which come from the two views of Figure 2a using the proposed NEIGHBORHOOD method. In this figure, the horizontal axis refers to the test view and the vertical axis refers to the reference view. Given these correspondences, the challenge is how to robustly estimate the model parameters (γ, b, G). Here, robust means that the model estimation should not be influenced by the outliers.

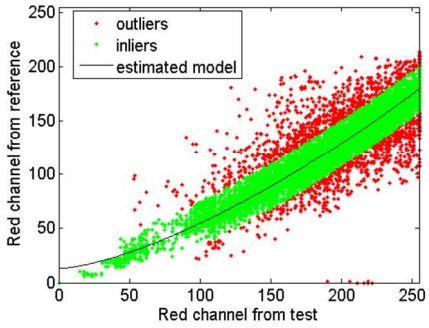

Figure 5. Color correspondences of the red channel; The correspondences are separated into outliers (red dots) and trusted data (green dots).

At this point, we face two challenges. First, we observe that the more the color correspondences contain lot of outliers, as it is usually the case, the more the least square estimation model is erroneous. Second, the color mapping model is nonlinear (equation 2). Therefore, to address these two main challenges, we propose to use a robust non linear regression model which is based upon a method called ROUT [13].

Inspired by ROUT, [13] the robust estimation is performed in two steps. First, outliers and inliers are classified iteratively. Next, we estimate model parameters from the inliers using least square method. In Figure 5, the red dots represent the outliers whereas the green dots represent the inliers. The black line shows the robustly estimated GOG (γ, b, G) curve.

Results

In order to evaluate our color mapping method we selected test images and built a quality evaluation framework.

Test images

Our test images include multiple views of a same scene having color differences but also additional views with known color differences that can be used as ground truth. Our test images are taken from 2006 stereo datasets of Middlebury [17, 8]. It

contains 21 datasets. Each data set contains several views of a same scene (laboratory style) taken under three different illuminations. For each illumination three different exposures are available and for each exposure there are seven views (three * three * seven = 63 images per dataset).

To conduct our experiments, we select the datasets belonging to the scenes named 'Aloe', 'baby1', 'baby2', 'baby3', 'bowling1', 'bowling2', 'cloth1', 'cloth2', 'cloth3', 'cloth4', 'flowerpots', 'rocks1' and 'rocks2'. We assume that in these datasets, there is no color differences between views having the same settings (same illumination and same exposure). By this assumption, we neglect image noise, any unintended temporal illumination fluctuations and any unintended changes of camera or scene geometry.

We need multiple views of a same scene having color differences. For each scene, in order to have color differences, we select a first view at a certain exposure and a second view at a different exposure. Since the exposure is different, image colors will be different. For example, Figure 6 (a) shows the scene called 'bowling1' where we selected view no. 0 at exposure 1 (500ms) and view no. 6 at exposure no. 2 (2000ms), under the same illumination no. 3. For each scene, the selected two views show such color differences that we want to compensate using color mapping. Once again, the main intuition behind this choice is the availability of ground truth. Figure 6 (b) shows the ground truth for the scene "bowling1" which is view no. 0 at exposure no. 2 under illumination no. 3.

Figure 6. (a) stereo pair(left and right) with color differences. (b) additional right view with same exposure as left view used as ground truth.

For all selected scenes of the 2006 stereo datasets of Middlebury, we define view no.6 at exposure no. 2 as reference view, I_{ref} and view no. 0 at exposure no. 1 as test view, I_{test} as shown in Table 2. Here, the term test view means that we want to compensate its colors by color mapping in order to be as close as possible to the reference view. We choose an additional view no. 0 at exposure no.2 as ground truth, I_{true}.

Table 2. Test image with ground truth from 2006 stereo datasets of Middlebury [17, 8]

	Illumination	Exposure	View no.
Reference view,I_{ref}	3	2(2000ms)	6
Test view,I_{test}	3	1(500ms)	0
Ground truth, I_{true}	3	2(2000ms)	0

Color mapping quality evaluation framework

To compare different color mapping methods, we propose an evaluation framework as shown in Figure 7. All color mapping methods take I_{ref} and I_{test} views as input. After estimating the color mapping model, each color mapping method tries to correct I_{test} view and produces an output called $I_{corrected\_test}$. Finally, the quality of the color mapping method is evaluated by comparing $I_{corrected\_test}$ with I_{true}.

This evaluation framework computes the remaining color differences between the color mapped view $I_{corrected\_test}$ and the ground truth I_{true} using CIE 2000 color-difference formula, CIEDE2000, ΔE00 [12]. The better the color mapping method works, the smaller the color difference remains.

For each pixel, we compute the remaining color differences after color mapping, $DE00_{after}(i,j)$, between I_{test} and $I_{corrected\_test}$ where (i,j) is the pixel's index.

$$DE00_{after}(i,j) = CIEDE2000\big(I_{test}(i,j), I_{corrected\_test}(i,j)\big)$$

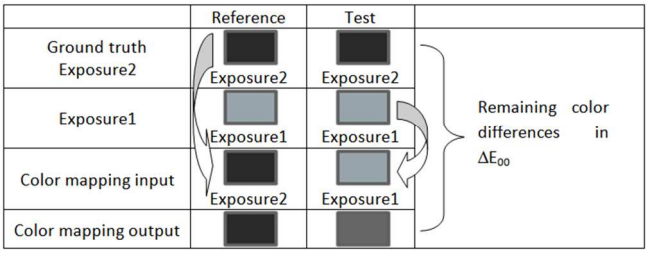

Figure 7. Quality evaluation of a color mapping method starts with a ground truth stereo pair having no color differences. Then, One view of exposure1 and another view of exposure2 is chosen as input to color mapping. Finally color mapping generates corrected test which is compared with ground truth to compute the remaining color differences.

Similarly the color differences before color mapping, $DE00_{before}(i,j)$, are computed as follows

$$DE00_{before}(i,j) = CIEDE2000(I_{test}(i,j), I_{ref}(i,j))$$

If color mapping can compensate color differences, we expect that

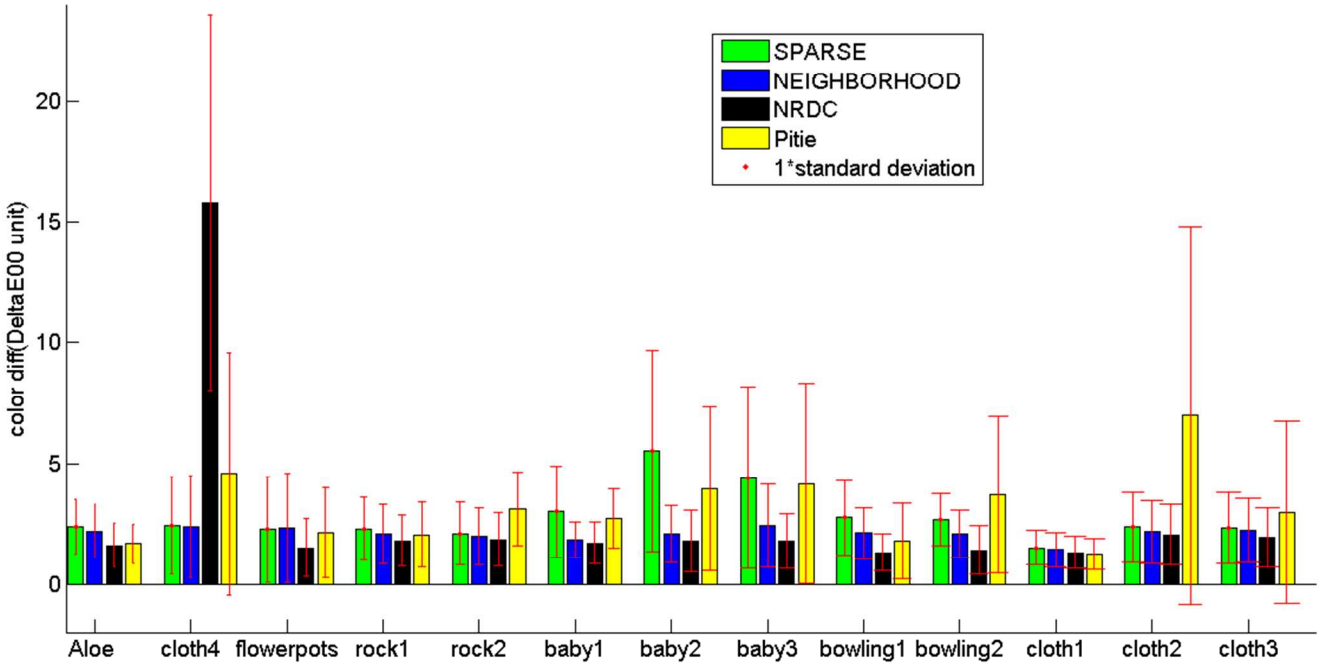

Figure 8. Comparison of four color compensation methods; These methods are SPARSE[19], NEIGHBORHOOD(proposed), NRDC[6] and Pitie[15]; The bar height represents the remaining color differences I_diff in DE00 unit ; The red line on each bar shows ±1standard deviation from the mean.

$$\mathrm{DE00}_{\mathrm{after}}(\mathrm{i},\mathrm{j}) < \mathrm{DE00}_{\mathrm{before}}(\mathrm{i},\mathrm{j})$$

In this color difference computation, each pixel contributes a single color difference. An average of all these color differences results in a single metric. We call this single metric "image color difference", I_{diff}

$$I_{diff} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{DE00}(\mathrm{i},\mathrm{j})}{\mathrm{N}}$$

where N is the total number of pixels of the image. Finally, we evaluate the quality of several color mapping methods by comparing their I_{diff}

Experimental results

In this section, we present the experimental results by comparing the proposed NEIGHBORHOOD method with three existing methods as shown in Table 3. Method no.1, called SPARSE, uses color correspondences extracted from sparse feature points such as the work of Tehrani et al. [25]. Method no.2, called NEIGHBORHOOD, is the proposed algorithm where spatial neighborhood around the sparse feature points is exploited. Method no. 3, called NRDC, refers to dense feature matching [8]. Finally. Method no.4 is not a color mapping but one of the most successful color transfer methods from Pitie et al. [20]

All these four methods shown in Table 3 are applied on all selected pairs of reference and test views and each method produces its own version of color mapped views. All those color mapped views are compared with the ground truth to calculate the remaining color difference I_{diff}.

Table 3. These four color compensation methods are compared in our experiment

Method	Method name
1	SPARSE [19]
2	NEIGHBORHOOD(proposed)
3	NRDC [6]
4	Pitie [15]

Figure 8 shows the remaining color differences for all 13 scenes using the four methods. In this figure, the red straight line on bars shows ±1standard deviation from the average remaining color difference. Note that, color difference less than one unit is usually barely noticeable to human eye. We observe from Figure 8 that the proposed NEIGHBORHOOD method always gains over SPARSE method. Hence, we found in Figure 8 that the consideration of neighborhood significantly increases the precision of color mapping over SPARSE in majority of the scenes, in spite of increasing uncertainty of correspondences.

If we analyze Figure 8, we can see that the performance of the proposed NEIGHBORHOOD method and the state of the art NRDC method are close. And, these two methods outperform the other two methods. Moreover, It is clear that the color transform method from Pitie et al.[20] does not perform well. We can also see in Figure 8 that the state of the art NRDC method performs comparatively well in most of the scenes except 'cloth4'. On the other hand, the proposed NEIGHBORHOOD method is comparatively stable throughout all of the 13 scenes and never significantly worse than any other method.

For each of the four color mapping method, we computed the average remaining color differences which is shown in Figure 9.

The NRDC methods fails for scene "cloth4". We therefore excluded this scene from averaging in Figure 9. Though, on average, NRDC outperforms the other methods, it seems to be less stable. One reason is that compared to other methods, NRDC's color mapping model utilizes more parameters and therefore, for some images it suffers from "over fitting" such as in "cloth4". On the other hand, from the analysis of Figure 8 and Figure 9, the proposed NEIGHBORHOOD method is more consistent and has limited maximal errors.

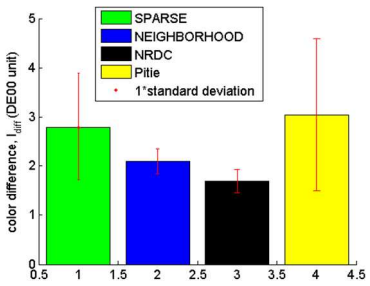

Figure 9. Average of the remaining color differences(I_{diff}) for all datasets except 'cloth4' obtained from four methods.

Conclusion

We present new color mapping method based on sparse feature matching. To address the problem of spatial precision of features, occlusion and insufficient representation of scene colors, we compute rich color correspondences from the spatial neighborhood of features based on clustering, matching and outlier detection. We propose a quality evaluation framework which is based on ground truth images and is capable of evaluating color mapping methods quantitatively.

From experimental comparisons, we find that the consideration of neighborhood of sparse features significantly increases the precision of color mapping in spite of the increasing uncertainty of correspondence. Comparisons with state of the art dense color mapping and color transfer methods show that our proposed method is able to achieve similar quality while only exploiting sparse correspondences. Moreover, our proposed method seems to be more stable for simple scenes and has smaller maximal errors. These results give some evidence that dense feature matching might not be necessary to solve the color mapping problem.

Future work will focus on reaching a more complete representation of scene colors. We also would like to extend the use case to with wider stereo baseline or strong viewpoint changes.

References

[1] Y. Chen, K.K. Ma, and C. Cai. Histogram-offset-based color correction for multi-view video coding. In *Proc. of ICIP*, pages 977–980, 2010.

[2] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(5):603–619, 2002.

[3] O. Dalmau-Cedeno, M. Rivera, and P.P. Mayorga. Computing the alpha-channel with probabilistic segmentation for image colorization. In *Proc. of ICCV*, pages 1 –7, oct. 2007.

[4] C. Doutre and P. Nasiopoulos. Color correction preprocessing for multiview video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(9):1400–1406, 2009.

[5] U. Fecker, M. Barkowsky, and A. Kaup. Histogram-based prefiltering for luminance and chrominance compensation of multiview video. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(9):1258–1267, 2008.

[6] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. Non-rigid dense correspondence with applications for image enhancement. volume 30, pages 70:1–70:9, 2011.

[7] Hasan, S F. and Stauder, J. and Tremeau, A. Robust Color Correction for Stereo. In *Proc. of Conference for Visual Media Production (CVMP)*, pages 101 –108, 2011.

[8] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8, 2007.

[9] S. Kagarlitsky, Y. Moses, and Y. Hel-Or. Piecewise-consistent color mappings of images acquired under various conditions. In *Proc. of ICCV*, pages 2311–2318, 2009.

[10] S.J. Kim and M. Pollefeys. Robust radiometric calibration and vignetting correction. *IEEE transactions on pattern analysis and machine intelligence*, pages 562–576, 2007.

[11] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[12] M.R. Luo, G. Cui, and B. Rigg. The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Research & Application*, 26(5):340–350, 2001.

[13] H. Motulsky and R. Brown. Detecting outliers when fitting data with nonlinear regression–a new method based on robust nonlinear regression and the false discovery rate. *BMC bioinformatics*, 7(1):123, 2006.

[14] M. Oliveira, A.D. Sappa, and V. Santos. Unsupervised Local Color Correction for Coarsely Registered Images. In *Proc. of CVPR*, pages 201–208, June 21-23 2011.

[15] F. Pitié, A.C. Kokaram, and R. Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1-2):123–137, 2007.

[16] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.

[17] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *Proc. of CVPR*, pages 1–8, 2007.

[18] Y.W. Tai, J. Jia, and C.K. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *Proc. of CVPR*, volume 1, pages 747–754, 2005.

[19] Mehrdad Panahpour Tehrani, Akio Ishikawa, Shigeyuki Sakazawa, and Atsushi Koike. Iterative colour correction of multicamera systems using corresponding feature points. *Journal of Visual Communication and Image Representation*, 21(5-6):377 – 391, 2010.

[20] Qi Wang, Xi Sun, and Zengfu Wang. A robust algorithm for color correction between two stereo images. In *Proc. of ACCV*, volume 5995, pages 405–416, 2010.

[21] K. Yamamoto and R. Oi. Color correction for multi-view video using energy minimization of view networks. *International Journal of Automation and Computing*, 5(3):234–245, 2008.