A Robust Algorithm for Color Correction between Two Stereo Images

Conference Paper · September 2009

DOI: 10.1007/978-3-642-12304-7_38 · Source: DBLP

3 authors, including:

Zengfu Wang
University of Science and Technology of Ch…

A Robust Algorithm for Color Correction between Two Stereo Images

Qi Wang[1], Xi Sun[1], Zengfu Wang[1,1]

[1] Dept. of Automation, University of Science and Technology of China
crabwq@mail.ustc.edu.cn, snxi@mail.ustc.edu.cn, zfwang@ustc.edu.cn

Abstract. Most multi-camera vision applications assume a single common color response for all cameras. However, significant luminance and chrominance discrepancies among different camera views often exist due to the dissimilar radiometric characteristics of different cameras and the variation of lighting conditions. These discrepancies may severely affect the algorithms that depend on the color correspondence. To address this problem, this paper proposes a robust color correction algorithm. Instead of handling the image as a whole or employing a color calibration object, we compensate for the color discrepancies region by region. The proposed algorithm can avoid the problem that the global correction techniques possiblely give bad correction results in local areas of an image. Many experiments have been done to prove the effectiveness and the robustness of our algorithm. Though we formulate the algorithm in the context of stereo vision, it can be extended to other applications in a straightforward way.

Keywords: Color correction, stereo images, OF-SIFT, mean-shift

1 Introduction

Stereo vision has traditionally been, and continues to be, one of the most extensively investigated topics in computer vision. Generally, a vision algorithm employs two or more images to recover depth information of a specific scene. In the algorithms presented, some assumptions about the physical world and the image formation process are used explicitly or implicitly. For example, surfaces in the scene are assumed to be Lambertian ones whose appearance does not vary with viewpoint and the multiple cameras employed are assumed to have uniform properties. Based on these assumptions, numerous algorithms have emerged and the increased sophistication of newer algorithms is producing a commensurate improvement in their performance [1].

However, these assumptions are not always true in real applications. For example, one can obtain stereo images with consistent color appearance in some cases, such as

[1] Corresponding author: Zengfu Wang. Address: Dept. of Automation, University of Science and Technology of China, Hefei, Anhui 230027, P.R.China. This work was supported by the National Natural Science Foundation of China (No.60875026).

when the Benchmark images in the famous Middlebury dataset [2] are used to be stereo images, but it is not appropriate in a real application. In a real environment, significant luminance and chrominance discrepancies among different camera views often exist due to the dissimilar radiometric characteristics of different cameras — even of the same type, and the variation of lighting conditions. These discrepancies may severely affect the algorithms that depend on the color correspondence and these algorithms abound in stereo vision field [2]. So it is necessary to consider the color correction problem in such an application. Our aim in this paper is to present a color correction method to ensure the color consistency between multi-camera views.

Previous work aimed at color correction mainly falls into two categories: calibrating cameras in order to obtain some desired response and processing images after acquisition. A common approach taken toward the first category is to calibrate each camera independently through comparisons with known colors on a color calibration object [3, 4]. Though a feasible solution, it is indeed an inconvenient and complex procedure. Furthermore, once the system is moved to another environment, the whole procedure must be repeated again. The other category involves a large variety of techniques, such as histogram matching [5], multispectral imaging technique [6], energy minimization in camera view networks [7], dominant basic color mapping [8], general color transfer method [9], selective color correction [10], etc. These techniques can compensate for color discrepancies between two images by using the global color distribution information of the two images. The advantage of the techniques is that they don't require a standard reference object for color calibration. However, they sometimes give bad correction results in local areas of an image. Besides the two categories mentioned above, color correction has also been studied in the fields of printers, scanners and monitors [11, 12]. But few of the corresponding techniques developed have been extended to camera systems.

In this paper, we propose a new color correction method based on image segmentation and keypoint matching. The general idea of our method is, instead of handling the image as a whole or employing a color calibration object, we compensate for the color discrepancies region by region. Regions and color discrepancies are acquired by segmenting the reference image and by comparing the color information of matched keypoints extracted from the images respectively. Here two state-of-the-art techniques are employed. One is mean-shift based segmentation technique [13] and the other is SIFT keypoint extraction technique [14]. We also present a novel optical flow based algorithm for SIFT keypoint matching — OF-SIFT, which can greatly speed up the keypoints matching. Though we formulate the problem in the context of stereo vision, the proposed method can be easily extended to other applications.

The rest of this paper is organized as follows. Section 2 introduces our color correction algorithm. Section 3 gives a detailed description of OF-SIFT. Experimental results are shown in Section 4, and conclusions are finally given in Section 5.

2 Methodology

Our aim in this paper is to make the color appearances of two stereo images consistent with each other. The two images are respectively named as target image and source image. We use the proposed method to adjust the source image so that it conforms to the target image in color appearance. Since our focus is on color correction procedure, we assume that the two images used in this paper are geometrically calibrated and the epipolar constraint is basically satisfied between them. This is an acceptable and reasonable assumption. For one thing, geometric calibration is a sophisticated technique [15] and can be done easily by using a known toolkit. For another, based on this assumption, we can keep our attention on the main problem of color correction.

In our procedure, we first extract SIFT keypoints from the two images and get matched pairs with OF-SIFT. Then the source image is segmented by mean-shift segmentation algorithm. The following color correction is performed region by region. For a given region in the source image, the color discrepancy is calculated by averaging the color discrepancies of matched pairs within it. If there are no matched pairs within the region, five keypoints will be added to the region and their corresponding matches in the target image will be identified. In the following, we will clarify our color correction algorithm step by step.

(1) Step 1: image acquisition. In the first place, two images, target image and source image, are acquired from two stereo cameras and we assume that they are geometrically calibrated. The source image is different from target image in color appearance for some known or unknown reasons.

(2) Step 2: color image transformation. Both of the acquired images, which are usually obtained in RGB color components, are transformed into the gray scale images and the HSI color images. Here, the gray scale images are used for SIFT keypoint extraction and the HSI color images are for the color correction.

(3) Step 3: SIFT keypoint extraction and matching. SIFT is one of the most widely used feature point detection technique in computer vision. The SIFT features are invariant to image scale and rotation, and can provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. Therefore, there can be hundreds of matched pairs, even though the source image differs greatly with the target image. However，the original SIFT algorithm searches keypoint pairs in an exhaustive manner and is time-consuming. In order to speed up the process, we use the OF-SIFT algorithm we proposed, which will be discussed in Section 3, to find matched pairs.

(4) Step 4: segmenting source image and calculating color discrepancy. In this step, the source image is segmented by using mean-shift based segmentation algorithm. Most of the segmented regions have SIFT keypoints, whose counterpart matches are in the target image. These regions are called matched regions. But there are other regions that do not have any SIFT keypoints within them for the extracted SIFT keypoints are sparse and do not have a uniform distribution. In this case, they are called unmatched regions.

For a given matched region S, we calculate its color discrepancy with the target image by averaging the color discrepancies between the SIFT keypoints in this region and its corresponding matches in the target image. Note that both of the images are transformed into HSI ones. Therefore, the color discrepancies are computed on three channels separately. We use the following color correction function to correct every pixel's color information in the region,

$$C_{new}(i,j) = C_{old}(i,j) + \left(\sum_{x_s \in S} [CN(x_t) - CN(x_s)] \right) \Big/ keyNumInS. \quad \textbf{(1)}$$

where (i,j) is a pixel of the region S, $C_{old}(i,j)$ is the value of the original color component of the pixel (C can be H , S or I) , and $C_{new}(i,j)$ is the corresponding new color value after correction of $C_{old}(i,j)$, x_s is a SIFT keypoint in region S of the source image, x_t is its corresponding match in the target image, $keyNumInS$ is the number of keypoints in region S, and $CN(x_s)$ and $CN(x_t)$ are the mean values of the colors within a 3×3 neighbor of x_s and x_t respectively.

For a given unmatched region, we compute its color discrepancy by using five keypoints. Fig. 1(a) shows the detail of this procedure. Suppose the shadow area in Fig. 1(a) is an unmatched region to be processed. In the first place, we find the four boundary points of the region: Top A, Left B, Bottom C and Right D. Then a rectangle EFGH is formed based on the four boundary points. After that, perpendicular bisector of each edge is drawn and we get two line segments between the two pairs of opposite edges. At last, we trisect each line segments with three points and label them all in the figure. There are five points (M0, M1, M2, M3 and M4) in together because two centre points are coincident in position. Therefore, the unmatched region has five added points as keypoints. The added keypoints are located the way that we explained above because we want them to sample as much color information of the entire region as possible. In addition to what we have done, another existing SIFT keypoint outside the unmatched region is also needed as a reference point for the five added keypoints. It is identified by finding the nearest SIFT keypoint to the added keypoint M0. In that case, we can easily find corresponding points of the five added keypoints according to their relative position to the reference keypoint and the corresponding match of the reference keypoint in the target image. After that, we can compute the color discrepancy in the same way as matched regions.

(5) Step5: color correction region by region. For each region in the source image, we first calculate the color discrepancy according to the way discussed in Step 4. Then all the pixels belonging to this region are corrected according to the region's color discrepancy. This process is repeated until all the regions are processed.

3 OF-SIFT

Each SIFT keypoint is described by a 128 high dimensional vector. To get the best candidate match for a SIFT keypoint in the source image, the closest neighbor and the second-closest neighbor should be identified first. The closest neighbor is defined as the keypoint with minimum Euclidean distance in the target image and the second-closest is the one with the second closest distance. After that, a ratio of the closest distance to the second closest distance is calculated to determine whether to accept the closest neighbor as the correct match or discard it as a false match. This is the way SIFT algorithm takes to obtain the matched pairs.

However, an exhaustive search process is involved to establish each pair of matched keypoints. That means, for each keypoint in the source image, that we need to calculate the distances of the keypoint and all of the other keypoints in the target image to determine the closest and second-closest neighbors. Obviously, it is a time-consuming process. In order to speed up the processing, we present a new algorithm based on optical flow calculation – OF-SIFT.

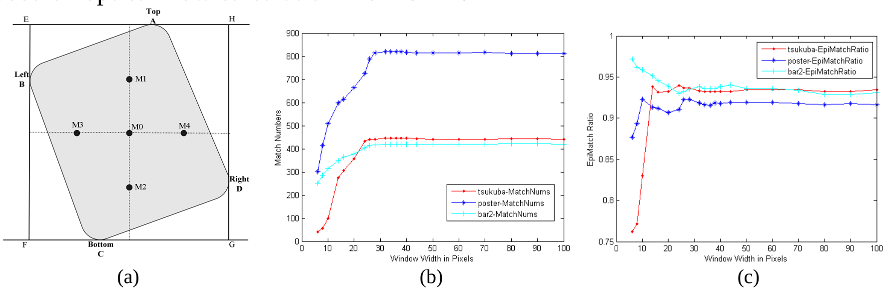


Fig. 1. (a) is an illustration of how to add five keypoints for an unmatched region in step 4 of our color correction method. (b) and (c) are the keypoint matching performance varied with widow width in OF-SIFT

The general idea of OF-SIFT is as follows. For each keypoint in the source image, we calculate its optical flow by Horn-Schunks algorithm [16]. According to the optical flow, we can get an estimation of the potential match's position in the target image. However, this estimation is not accurate enough because of the existence of noise and the algorithm itself. Subsequent keypoint matching is proceeded within a rectangle window centered at the estimated point. When making a decision on the true matched pair, a relaxed epipolar constraint should be satisfied. This means that when a keypoint in the source image lies in scan line i, its corresponding match in the target image should lie in scan line i-1, i or i+1. We check the matched pair this way because generally speaking, the calibration can not be so precise that the corresponding match lies exactly in the same scan line. Besides that, we also set a discrepancy threshold T ranged [0, 1] to control the acceptance level of color discrepancy (In our process, the HSI channels are all scaled to [0, 1]). If the color discrepancy of the matched pair is smaller than T, we accept them as a true matched pair. Otherwise, they are discarded as a false matched pair.

For the selection of the matching window size, we have the following considerations. Since the images in our experiment are geometrically well calibrated and we assume they satisfy the relaxed epipolar constraint, the height of the matching window is set to be 3 pixels. This is able enough to tolerate some errors in actual practice. As for the width of the matching window, we have done a lot of experiments to examine its effect on matching performance. All the 38 pairs of stereo images in our experiments are selected from the Middlebury dataset [2]. They are all well calibrated. We vary the window width to see its influence on the numbers of true matches and the true matching rates. In order to see the results clearly, graphs of only three pairs of images (Tsukuba, Poster and Bar2) are shown in Fig. 1(b, c). Other results that do not appear in the Fig. are nearly the same as those of the three. The graph (b) in Fig. 1 shows that with the increase of window width, the total matched pairs increase accordingly. But when the width is larger than forty, the numbers remained unchanged. We can also find from graph (c) that the true matching rates vibrate violently when the width is small. However, when the width increases larger than forty, the curves stay flat. Therefore, a width of forty pixels is an appropriate selection for the matching window, because larger window will not improve the performance. Instead, it will cost much more processing time.

4 Experiments

Many experiments are done to examine the performance of OF-SIFT and the presented color correction algorithm. All the 38 pairs of stereo images mentioned in Section 3 are employed in our experiments. Due to space limitation, only results of the mostly used four pairs (Venus, Tsukuba, Teddy, Cones) are reported here. All the programs are run in a computer with Pentium 4 CPU 2.93GHz and 1G memory.

4.1 Performance of OF-SIFT

The original SIFT algorithm was implemented by the author in Matlab environment. In order to compare our OF-SIFT with SIFT fairly, all the programs are run in Matlab environment.

Table 1 lists the experimental results. The first column in the Table is the aspects to be compared. They are the true matched pairs, the matching operations (One matching operation means to calculate the Euclidean distance between two keypoints one time.), time consumed in the whole matching process, reduced matching operation and time compared with SIFT respectively. We can see clearly that our OF-SIFT can find more matched pairs than SIFT. That's because SIFT searches the matched keypoints in the entire target image. In order to avoid the false matches, the strict judgement is needed. On the contrary, OF-SIFT restricts the searching scope to a reasonable smaller window area and the candidates of the matched pairs obtained are more likely to be true. The Table also indicates that OF-SIFT costs far less matching operations, therefore far less time than SIFT. At least 97% matching operations and matching time can be saved. This is a critical advantage for real-time applications and it demonstrates the efficiency of OF-SIFT.

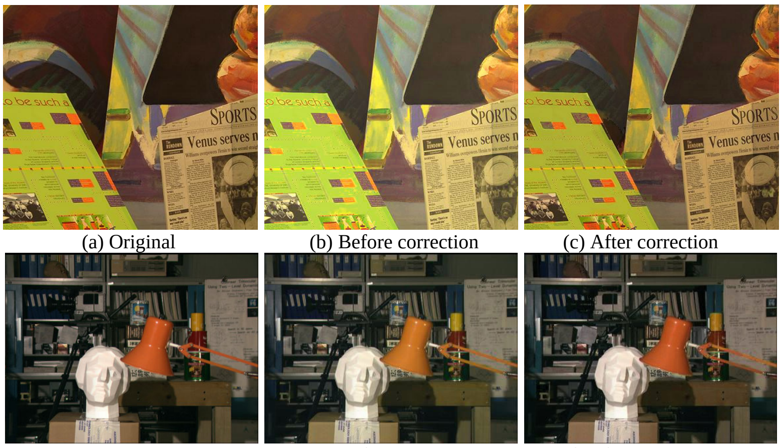
Table 1. Comparison of SIFT and OF-SIFT. For the consideration of limited paper length, results of only four images are reported here

	Venus	Tsukuba	Teddy	Cones
SIFT/OF-SIFT True Matched Pairs	397/450	344/416	417/486	641/753
SIFT/ OF-SIFT Matching Operations	532170/6141	595968/7343	911028/7723	2202175/14046
SIFT/ OF-SIFT Matching Time (s)	4.697/0.060	5.119/0.111	8.271/0.070	21.368/ 0.120
Reduced Matching Operations (%)	98.84	98.76	99.15	99.36
Reduced Matching Time (%)	98.72	97.83	99.15	99.44

4.2 Performance of Color Correction Algorithm

In this part, we will examine the performance of our color correction algorithm. Image pairs for experiments are all from Middlebury dataset. In each pair, one image is adjusted to be different from the other in color appearance. The adjusted image is treated as source image and the other one as target image. Then we use our proposed algorithm to correct the adjusted image to be the same as the original one as possible. The reason for employing the Middlebury dataset and adjusting one image as the source image is based on the following considerations. Middlebury dataset is a publicly available and widely accepted dataset in stereo vision. All the image pairs are well calibrated and there is ground truth depth information for each pair. If we use these images to conduct our experiments, the acquired results can be objectively compared with the standard known information from the dataset. That will make our conclusion more convincing.

We adjust the images separately on three channels of H, S and I, using the Photoshop software. We select regions of different sizes in the image to adjust their colors. Different regions have different adjustments and the largest variation in pixel value from the original image is up to 25%. This means we set the threshold T as 0.25. We program our algorithm in Microsoft Visual Studio .NET 2003 environment and the experimental results are evaluated from the following three aspects.



(a) Original (b) Before correction (c) After correction

(d) Original (e) Before correction (f) After correction

Fig. 2. Illustration of color correction results for subjective evaluation. The first row lists Venus images and the second Tsukuba. Each row from left to right, is respectively the original image, adjusted image before correction and result image after correction

4.2.1 Subjective Evaluation

To compare color information qualitatively is a known hard problem. Objective evaluation usually gives a qualitative value to represent the goodness of the results. But it does not necessarily coincide with the human perception. This was noted by Cinque et al. [17]:“Although it would be nice to have a quantitative evaluation of performance given by an analytical expression, or more visually by means of a table or graph, we must remember that the final evaluator is man." Therefore, we should assess our color correction results in a subjective manner. Fig. 2 displays the results of two pairs of images, Venus and Tsukuba. Each row from left to right, they are respectively the original image, adjusted source image before correction and the result image after correction. We can see clearly that the adjusted image differs greatly with the original one. Furthermore, since different regions of one image are adjusted differently, their color discrepancies are not the same. But after correction of our algorithm, the result images look nearly the same as the original one.

4.2.2 Histogram Evaluation

We also assess our color correction results in the form of histogram comparison. Color channels are compared separately. In order to see the difference clearly, we draw the histogram envelop curves of the original image, adjusted source image before correction and the result image after correction, together in one graph. The range of the three horizontal axes is scaled to [0, 1] and that of the vertical axes is normalized by the total pixel number of the image. Obviously from Fig. 3 we can see that, histograms of the Venus and Tsukuba result images have a greater resemblance with the original ones than that of the adjusted images.

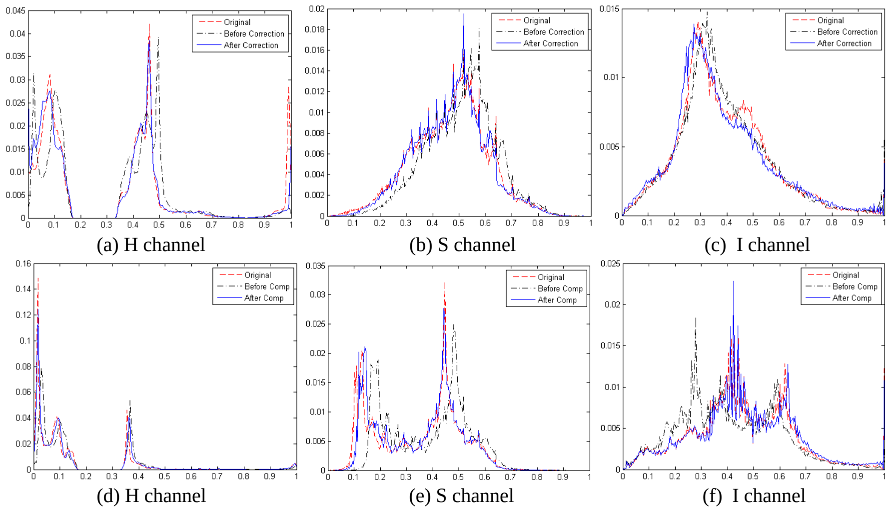
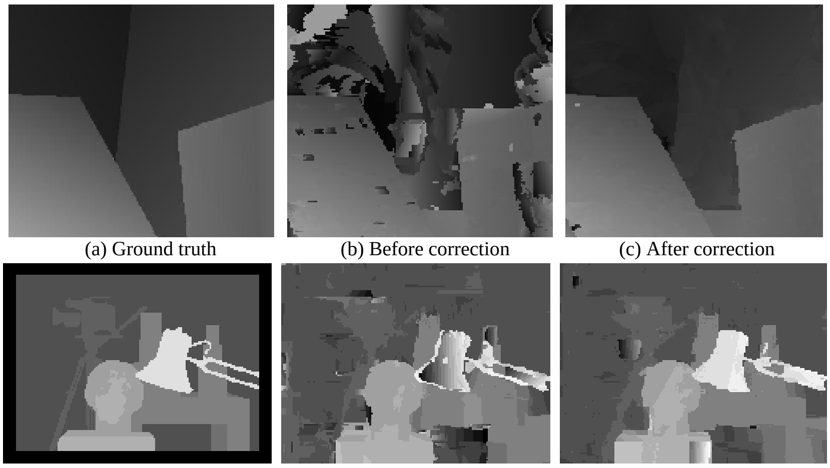


Fig. 3. Illustration of color correction results for histogram evaluation. The first row lists Venus results and the second Tsukuba. Each row from left to right, they are respectively the histograms of H, S and I channels

4.2.3 Stereo Vision Evaluation

We also evaluate our algorithm in the context of stereo matching. One state-of-the-art stereo matching algorithm, which is based on cooperative optimization [18] and is the best ranked algorithm [2], is employed in our experiment. The adjusted source image before correction is first used to calculate the disparity map. Then the result image is used the same way. At last, their results are compared with the ground truth disparity map issued by the Middlebury website. Fig. 4 shows our experimental results of Venus and Tsukuba image pairs. We can find that the disparity maps calculated by the adjusted images have much noise and differ greatly with the original ground truth maps. But after correction, they are much smoother and show greater resemblance to the ground truth disparity maps. Due to the sake of exactly known disparity information, we can give a quantitative comparison of the color correction performance. We compare our calculated disparity maps with the ground truth ones. Pixels diverting from the ground truth values larger than one are treated as bad pixels. Then we compute the percentage of the bad pixels in the entire map. Table 2 shows the results, from which we can easily see that the error rates drop sharply after our color correction process. Besides, the poor performance of stereo algorithm confronting with the color discrepancy also reveals an existing problem that, some stereo matching algorithms neglect the color correction procedure. Although Middlebury website provides convenient stereo image pairs that are well calibrated as a platform to compare stereo matching algorithms, we may encounter the challenging problem in real applications that the images captured by different cameras may not coincide with each other in color appearance. In this case, even the best existing stereo matching algorithm may not work perfectly. This reflects the meaning of our work from another point of view.



(a) Ground truth (b) Before correction (c) After correction

(d) Ground truth (e) Before correction (f) After correction

Fig. 4. Illustration of color correction results for stereo vision evaluation. The first row lists Venus disparity maps and the second Tsukuba. Each row from left to right, is respectively the disparity maps of original ground truth, calculated by using the adjusted image before correction and by the result image after correction

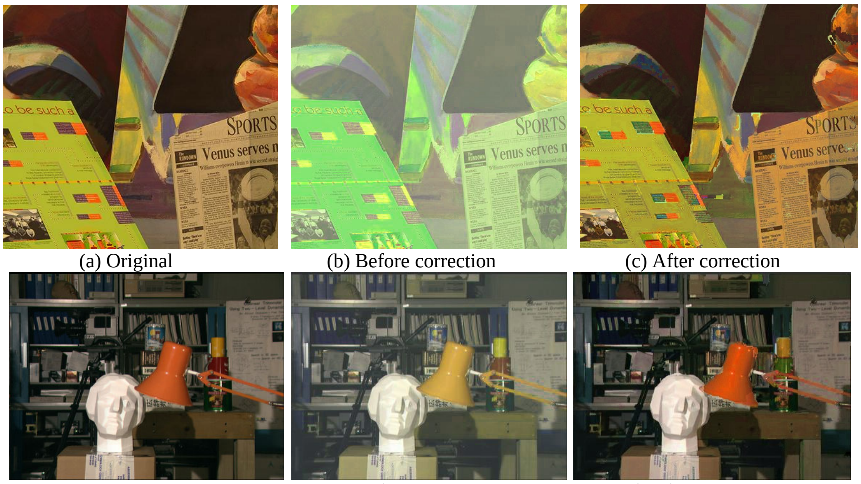
4.3 Experiments under Extreme Conditions

Experiments in Section 4.2 are conducted when the source and original images have a discrepancy up to 25% in color value. In this part, we evaluate our algorithm in the condition of greater discrepancy. Fig. 5 displays the results of Venus and Tsukuba image pairs. Different regions of source image have different adjustments and the most salient variation in pixel value from the original image is up to 80%. That means we set the discrepancy threshold T as 0.8.

Table 2. Error rates comparison of disparity calculation before and after color correction process

	Error Rates of Different Images			
	Venus	Tsukuba	Teddy	Cones
Before Correction(%)	44.64	12.37	35.26	64.90
After Correction(%)	4.17	6.46	15.01	24.85

From the results, we can see that the adjusted source images look absolutely different from the original one. Fortunately, our algorithm works well enough to correct most of the regions to make them resemble to the original ones. But there are still some regions not properly corrected because of the inaccurate color discrepancies, which are resulted from the false matched pairs. Although the relaxed OF-SIFT can reduces false matches compared with SIFT, it can not eradicate mismatches from happening. Therefore, when there are regions with false matched pairs, their color correction results may not be correct.



(a) Original (b) Before correction (c) After correction

(d) Original (e) Before correction (f) After correction

Fig. 5. Color correction results under extreme conditions. The first row lists Venus images and the second Tsukuba. Each row from left to right, is respectively the original image, adjusted image before correction and result image after correction

4.4 Experiments on Real World Scene

The images used in the above experiments are all from Middlebury dataset. In this part, we present our results on real world scene. The stereo images are taken by two cameras, FinePix S5000, with different parameter settings. The original right image differs greatly in color appearance with the actual scene color. But after our correction process, their color consistency is improved a lot. Fig. 6 shows our results.

(a) Original left image (b) Original right image (c) Right image after correction

Fig. 6. (a) and (b) are the original image pair taken from stereo cameras. (c) is the right image after color correction

5 Conclusions

In this paper, we present a color correction algorithm to compensate for the color discrepancy between two stereo images. Instead of correcting the image in a global manner or employing a calibration object, our color correction process is conducted region by region. This makes our method more convenient and accurate. Many experiments have also been done to prove the efficiency and robustness of the proposed algorithm. But for the consideration of limited paper length, only a few of them are reported. The results of other image pairs are consistent with the conclusion.

We also present an optical flow based algorithm to speed up the SIFT keypoint matching process. This is indeed effective. However, SIFT keypoint extraction and mean-shift based segmentation are both time-consuming. How to find a more rapid as well as more robust keypoint extraction and image segmentation methods is a challenging work in the future. In addition, how to eradicate mismatches of SIFT keypoints is another problem to be researched.

Although we formulate the color correction problem in the context of stereo vision, the presented algorithm can be extended to other applications in a straightforward way. For example, if there are more than two cameras in a vision system, we chose one as the reference camera. Images captured by other cameras can be separately corrected according to the target image from the reference camera. Another example can be found in multi-view video coding. In this field, different coding schemes have been proposed which explore not only temporal correlation between subsequent frames but also the special correlation between neighboring camera views. Unfortunately, ununiform camera responses often exist. In this case, the presented algorithm can be helpful.

References

1. Scharstein, D., Szeliski, R.: A Taxonomy and Evaluation of Dense Two-frame Stereo Correspondence Algorithms. International Journal of Computer Vision. 47(1/2/3), (2002) 7–42
2. http://vision.middlebury.edu/stereo/
3. Ilie, A., Welch, G.: Ensuring Color Consistency across Multiple Cameras. In Proc. Tenth IEEE International Conference on Computer Vision (ICCV). 2(2005) 17–21
4. Unal, G., Yezzi, A.: A Variational Approach to Problems in Calibration of Multiple Cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence. 9(8), (2007) 1322–1338
5. Chen, Y., Cai, C., Liu, J.: YUV Correction for Multi-View Video Compression. In Proc. 18th International Conference on Pattern Recognition (ICPR). 3(2006) 734–737
6. Cherdhirunkorn, K., Tsumura, N., Nakaguchi1, T., Miyake, Y.: Spectral Based Color Correction Technique Compatible with Standard RGB System. Optical Review. 13(3), (2006) 138–145
7. Yamamoto, K., Oi, R.: Color Correction for Multi-view Video Using Energy Minimization of View Networks. International Journal of Automation and Computing. 5(3), (2008) 234–245
8. Shangguan, L., Sun, J.: Multi-View Video Coding Using Color Correction. Workshop on Power Electronics and Intelligent Transportation System. (2008) 149–152
9. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color Transfer between Images. IEEE Computer Graphics and Applications. 21(5), (2001) 34–41
10. Inoue, A., Tajima, J.: Selective Color Correction for Arbitrary Hues. In Proc. International Conerence on Image Processing. 3(1997) 38–41
11. Bala, R., Sharma, G., Monga, V., Capelle, J.-P Van de. :Two-Dimensional Transforms for Device Color Correction and Calibration. IEEE Transactions on Image Processing. 14(8), (2005) 1172–1186
12. Kang, H.: Color Technology for Electronic Imaging Devices. SPIE-International Society for Optical Engineering. 1997
13. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. 24(5), (2002) 603–619
14. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision. 60(2), (2004) 91–110
15. Forsyth, D.A., Ponce, J.: Computer Vision: A Modern Approach. Prentice Hall. 2002
16. Horn, B.K.P., Schunck, B.G.: Determining Optical Flow. Artificial Intelligence. 1981
17. Cinque. C., Guerra, C., Levialdi, S.: “Reply: On the Paper by R. M. Haralick,” CVGIP: Image Understanding. 60(2), (1994) 250–252
18. Wang, Z.F., Zheng, Z.G.: A Region based Stereo Matching Algorithm Using Cooperative Optimization. In Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2008) 1–8