Color Balancing for Change Detection in Multitemporal Images

Jim Thomas, Kevin W. Bowyer and Ahsan Kareem
University of Notre Dame
Notre Dame IN 46556
{jthoma14,kwb,kareem}@nd.edu

Abstract

Automatic color balancing approaches for different applications have been studied by different research communities in the past decade. However, in this paper we address color balancing for the purpose of change detection. Images of a scene taken at different times may have variations in lighting and structural content. For such multitemporal images, an ideal color correction approach should be effective at transferring the color palette of the source image to the target image for the unchanged areas while being able to transfer the global color characteristics for the changed area without creating visual artifacts. Towards this goal, we propose a new local color balancing approach that uses adaptive windowing. We evaluated the proposed method against other state-of-the-art ones using a database consisting of aerial image pairs. The test image pairs were taken at different times, under different lighting conditions, and with different scene geometries and camera positions. On this database, our proposed approach outperformed other state-of-the-art algorithms.

1. Introduction

Color transfer or color balancing is the process of transferring the color characteristics of a source image to a target image. Depending on the application, the color differences between source and target images arise due to different reasons. In aesthetic applications, the source and target images may belong to completely different scenes. When both the images roughly correspond to different views of the same scene, the goal becomes to correct the color differences of the overlapping regions. Several factors affect the intensities recorded by cameras. They include: exposure variances, white balancing, gamma correction, vignetting and digitizer parameters [10].

While most previous work in color balancing has been done for aesthetic purposes or for image stitching, to the best of our knowledge no effort has been put into color correction for change detection. For change detection applications, we ignore non-overlapped areas. As the overlapped area may contain changed regions, the ideal algorithm should transfer local characteristics of the unchanged regions while transferring global characteristics for changed regions. While transferring local and global characteristics, the transition should also be smooth so that no artifacts are produced.

Figure 1. *(Top left)* Source image (after storm). *(Top right)* Target image (before storm). *(Bottom)* New target image after applying color balancing using the proposed algorithm.

We are particularly interested in the problem of color balancing for images taken at different times; such as aerial imagery taken before and after a hurricane or other disaster. A damage assessment from such before- and after-storm images require color balancing in the preprocessing stage. In addition to previously mentioned factors, the color differences in this case arise due to different camera parameters, local reflective properties of the object on the ground, changes due to storm damages and changes that occur over time. Previous research [13] in damage assessment has ignored color balancing and hence the outcome of change detection was affected by lighting differences. The scope of this paper does not include evaluation of improvement in change detection after color balancing. Instead, we focus

on a solution to correct images where such natural or man-made changes occur to landscape (see Figure 1).

2. Related Work

Color balancing algorithms can be classified into model-based parametric approaches and modeless non-parametric approaches. Parametric approaches use statistical models to transfer characteristics to a target image. Such approaches may use a global model that is applied to one or all three channels [9]. Global modeling usually provides only a rough mapping between the color of two images. A local color transfer scheme based on probabilistic image segmentation, region mapping using Gaussian mixture models (GMM) and the EM algorithm was proposed by Tai et al. [11]. Non-parametric methods [4], [5], [8], [3] assume no particular parametric format of the color mapping function and most of them use a look-up table to directly record the mapping of the full range of color/intensity levels.

A recent survey of color correction algorithms has compared nine color correction methods [14]. The study found that parametric approaches outperform their non-parametric counterparts. Gain compensation [2] and EM segmentation-based local color transfer algorithm [11] outperformed all others and are recommended as the first options to try for a general image and video stitching application. Very recently, a mean-shift segmentation based local color transfer was proposed in [7]. This approach was found to outperform [11] in terms of color similarity but structural similarity was not taken into account in the evaluation.

Using a segmentation-based method greatly constrains the granularity of local mapping. For instance, reflective properties of a small rooftop may cause it to vary in how the color changes from source to target, from all the other rooftops of the same color but a different material. Further, segmentation-based methods require a matching of segments which gets harder when parts of a region have changed significantly. The main contributions of this work can be summarized as: a) For a fine-grained local mapping, we use moving windows which adapt in size to transfer local statistics. b) The proposed method ensures that the structural integrity of the target image is preserved. c) We use integral images to make computations faster. d) We also discuss how images are registered with a reasonably high accuracy before color correction.

3. Finding Correspondences

As discussed in previous section, color balancing algorithms require a matching of local regions. In our approach, we use a standard feature based matching to find correspondences. The images are then registered and cropped to neglect the non-overlapping areas. While a comprehensive study of image registration is out of the scope of this paper, in this section we discuss our approach briefly. We adopt the scale- and rotation-invariant interest point detector and descriptor, coined SURF (Speeded Up Robust Feature) first introduced in [1] .

3.1. Matching and Transformation

Once the interest points are detected and described, a k-Nearest Neighbor algorithm is used to find the first and second nearest neighbors. Since our images were large (4077×4092) and large number of points were detected, we used an approximate k-Nearest Neighbor algorithm called FLANN [6]. Since this approximation, along with noise/non-overlapping area decreased the number of actual inliers, we used a modified version of RANSAC to find the subsets. This constrained form of RANSAC was different in that a) when subsets of 4 matched pairs were chosen, these 4 pairs had to follow certain geometric constraints; namely, cyclic or anti-cyclic order had to be conserved and collinearity had to be conserved. b) the upper limit on the number of iterations of RANSAC was determined by a greedy method which counts the iterations since last best subset was found. This modified form of RANSAC along with FLANN was used to find the best matching of interest points and consequently, compute the homography matrix. All the images in our database were thus registered with an accuracy of 1-5 pixel registration error. This was true even for image-pairs with only 5-10$\%$ of the SURF detectors being inliers.

4. Local Color Transfer

Consider the source image $s(i,j)$, target image $t(i,j)$ and new target image $t^{new}(i,j)$ formed after color transfer. Global color transfer as proposed by [9] first converts the RGB color space into $l\alpha\beta$ color space. Once the channels have thus been decorrelated, the statistics are transfered by the following equations:

$$t^{new}(i,j) = \mu_s + \frac{\sigma_s}{\sigma_t}(t(i,j) - \mu_t) \tag{1}$$

where μ_s and μ_t are means of the source and target image respectively. Similarly, σ_s and σ_t are the standard deviations of the source and target images respectively.

We propose to extend the above method to a local approach where statistics are calculated for $k \times k$ windows over the image. The choice of using [9] as a starting point is due to the fact that the computations are simple, no training is required and the algorithm makes no strong assumptions such as constant illumination. The above equation now becomes:

$$t^{new}(i,j) = \mu^k_{s_{(i,j)}} + \frac{\sigma^k_{s_{(i,j)}}}{\sigma^k_{t_{(i,j)}}}(t(i,j) - \mu^k_{t_{(i,j)}}) \tag{2}$$

$$\mu^k_{s_{(i,j)}} = \frac{1}{k^2} \sum_{l=i-\frac{k}{2}}^{i+\frac{k}{2}} \sum_{m=j-\frac{k}{2}}^{j+\frac{k}{2}} s(l,m) \tag{3}$$

$$\sigma^k_{s_{(i,j)}} = \frac{1}{k} \sqrt{\sum_{l=i-\frac{k}{2}}^{i+\frac{k}{2}} \sum_{m=j-\frac{k}{2}}^{j+\frac{k}{2}} (s(l,m) - \mu^k_{s_{(i,j)}})^2} \tag{4}$$

The means are now indicated by $\mu^k_{s_{(i,j)}}$ and $\mu^k_{t_{(i,j)}}$, where k denotes the length of the window used for transferring the statistics around the pixel (i,j). Similarly, the standard deviations are indicated by $\sigma^k_{s_{(i,j)}}$ and $\sigma^k_{t_{(i,j)}}$. However, in this approach, k remains a constant for the transfer functions corresponding to all pixels. The outcome of the transfer thus depends on the value of k. If k is too small, it causes artifacts to appear as the transfered statistics distort the structural components of the target image (See Figure 2). This can be solved by choosing a sufficiently large size for k.

4.1. Adaptive Windowing

In order to characterize local differences between the images which may occur due to lighting variations, damages, shadows or other structural variations, we propose to compute normalized cross correlations (NCC) between the corresponding windows in source and target images. NCC corresponding to a pixel (i,j) and window length k can be computed as:

$$NCC(i,j) = \frac{\sigma^k_{s_{(i,j)}t_{(i,j)}}}{\sigma^k_{s_{(i,j)}} \sigma^k_{t_{(i,j)}}} \tag{5}$$

$\sigma^k_{s_{(i,j)}}$ and $\sigma^k_{t_{(i,j)}}$ are local standard deviations of source and target images. $\sigma^k_{s_{(i,j)}t_{(i,j)}}$ is the cross-covariance between corresponding windows, calculated as:

$$\sigma^k_{s_{(i,j)}t_{(i,j)}} = \sum_{l=i-\frac{k}{2}}^{i+\frac{k}{2}} \sum_{m=j-\frac{k}{2}}^{j+\frac{k}{2}} \frac{(s(l,m) - \mu^k_{s_{(i,j)}})(t(l,m) - \mu^k_{t_{(i,j)}})}{k^2} \tag{6}$$

The window length k for each pixel can be fixed by calculating the value of $NCC(i,j)$ for a range of window sizes and choosing the smallest window size that gives a sufficiently high NCC value. Thus, the proposed method has three parameters, k_{min} and k_{max} which control the range of window sizes to be considered and NCC_{min} which decides the minimum value of NCC that is required to fix a window size. Our approach can be summarized as the following: For each pixel (i,j) compute the final values using equation (2) where k is minimized such that $k_{min} \leq k \leq k_{max}$ and $NCC(i,j) \geq NCC_{min}$.

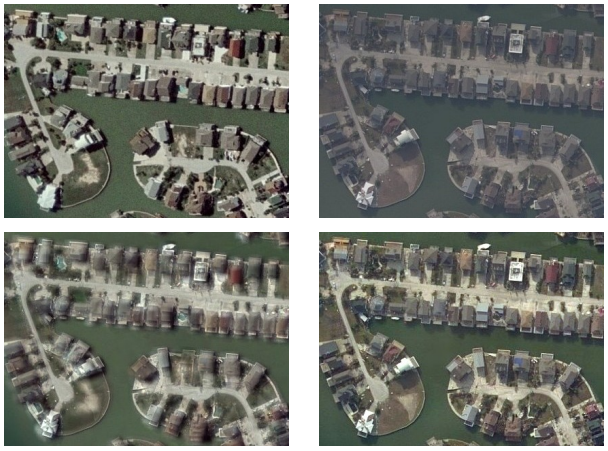

Figure 2. *(Top left)* Source image. *(Top right)* Target image. *(Bottom left)* 11X11 windows, artifacts visible. *(Bottom right)* 101X101 windows, no artifacts visible.

4.2. Computation using Integral Images

Since computing the means, standard deviation and cross-covariance for multiple large window sizes could be expensive, we used integral images to speed up the process. Once calculated for an image, an integral image can be used to perform summation over any rectangular area in the image in constant time. An integral image I of an input source image s is defined as the image in which the intensity at a pixel position is equal to the sum of the intensities of all the pixels above and to the left of that position in the original image. So the intensity at position (i,j) can be written as:

$$I_s(i,j) = \sum_{l=0}^{i} \sum_{m=0}^{j} s(l,m) \tag{7}$$

The integral image of any greyscale image can be efficiently computed in a single pass. Once we have the integral image, the local mean for any window size can be computed simply by using two addition and two subtraction operations instead of the summation over all pixel values within that window: $\mu^k_{s_{(i,j)}} = (I(i+k/2, j+k/2) + I(i-k/2, j-k/2) - I(i+k/2, j-k/2) + I(i-k/2, j+k/2))/k^2$

Similarly, if we consider the computations of the local standard deviation and local cross-covariance

$$\sigma^k_{s_{(i,j)}} = \frac{1}{k} \sqrt{\sum_{l=i-\frac{k}{2}}^{i+\frac{k}{2}} \sum_{m=j-\frac{k}{2}}^{j+\frac{k}{2}} s^2(l,m) - (\mu^k_{s_{(i,j)}})^2} \tag{8}$$

$$\sigma^k_{s_{(i,j)}t_{(i,j)}} = \sum_{l=i-\frac{k}{2}}^{i+\frac{k}{2}} \sum_{m=j-\frac{k}{2}}^{j+\frac{k}{2}} \frac{s(l,m)t(l,m)}{k^2} - \mu^k_{s_{(i,j)}}\mu^k_{t_{(i,j)}} \quad (9)$$

The first term in Equations 8, 9 can be computed in a similar way by using integral images of the squared pixel intensities and integral images of pixel products respectively. Once the integral image is calculated, local means, standard deviations and NCC can be computed in constant time and independent of the local window size.

4.3. Proposed Algorithm

1. All the integral images $I_s(i,j)$, $I_t(i,j)$, $I_{st}(i,j)$, $I_{s^2}(i,j)$ and $I_{t^2}(i,j)$ are computed for the three channels using the procedure mentioned in Section 4.2.

2. Convert $s(i,j)$ and $t(i,j)$ to grayscale and initialize $k = k_{min}$. Record values of k for each pixel (i,j) in a window index map $WIM(i,j)$.

3. For each pixel (i,j), compute $NCC(i,j)$ using integral images. If $NCC(i,j) < NCC_{min}$ and $WIM(i,j) < k_{max}$, increment $WIM(i,j)$ by a small value δk.

4. Repeat Step 3 until there is no more change for any pixel (i,j) in $WIM(i,j)$.

5. Smooth $WIM(i,j)$ using a Gaussian blur to allow smooth transition of window sizes and avoid artifacts at boundaries.

6. Using the value of k recorded in $WIM(i,j)$, apply Equation 5 for all pixels (i,j) using integral images.

The use of integral images brings down the time complexity of the algorithm to $O(rN)$ where N is the number of pixels in the image and $r = k_{max} - k_{min}$.

5. Evaluation

For our evaluation, we used 15 pairs of images, each of size 4077×4092 pixels before registration. The images were obtained from a mix of different sources. Most of the aerial imagery was acquired by the NOAA Remote Sensing Division. The images were approximately 50 cm resolution. The NOAA images used were from Hurricane Dennis (2005) and Hurricane Ivan (2004) images of Pensacola, Florida. Additionally, images were downloaded from USGS High resolution Orthoimagery database which were of 60 cm resolution.

We use the two measures of performance that were used in [14]; namely, color similarity and structural similarity. While color similarity is of primary importance in evaluating the performance, structural similarity is essential to ensure that the color transfer does not lead to destroying the structure of the target image. For instance, if the algorithm equates the values of each pixel in the source to the target, we have an optimal color similarity measure but poor structural similarity. Let r be the transformed image, s is the source image and t be the target image. Color similarity (CS) measure $CS(r,s)$ is defined as $CS(r,s) = PSNR(r,s)$ where $PSNR = 20 * log_{10}(L/RMS)$ is the peak signal to noise ratio. L is the largest possible value in the dynamic range of an image, and RMS is the root mean square difference between two images. The higher the value of $CS(r,s)$ the more similar the color between the two images r and s. The other measure used is structural similarity index measure or $SSIM(r,t)$, which is described in [12]. $SSIM$ is as a combination of luminance, contrast and structure components and is computed over local windows. The higher $SSIM$ is, the higher the similarity between the structure of r and t, and $SSIM(r,t) = 1$ if there is no structure difference.

We compared our algorithm with 7 other algorithms that were presented in [14]. The algorithms used are listed in Table 1. The results of running these algorithms on our image pairs are shown in Figure 3 and Figure 4. Note that our proposed algorithm, Algorithm 8 in the Table, has the highest color similarity measures, outperforming the other algorithms. From Figure 4, it is clear that the superior performance in transferring local color characteristics is done so without sacrificing the structural integrity of the target image. The slightly lower values of SSIM index as compared to some of the other algorithms is an expected result as color balancing could lead to changes in illumination. Similar to what was observed in [14], gain compensation and local color transfer are the next best performers. The output of our algorithm and the output of gain compensation is showed in Figure 5 for some of the images in our database.

#	Approach	Type
1	brightness transfer function [5]	local
2	cumulative histogram mapping [3]	local
3	gain compensation [2]	local
4	global color transfer [9]	global
5	local color transfer [11]	local
6	iterative color distribution transfer [8]	global
7	principle regions mapping [15]	local

Table 1. List of algorithms used in our study.

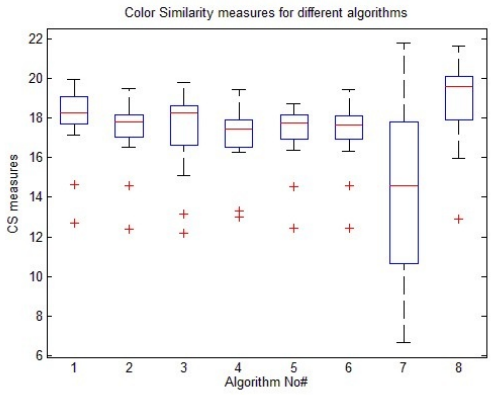

Figure 3. Color similarity measures for different algorithms.

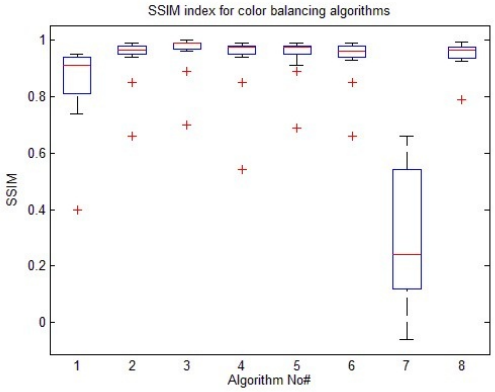

Figure 4. SSIM for different color balancing algorithms.

6. Conclusion

This paper presented a parametric local color balancing approach that uses adaptive windowing. The proposed algorithm is shown to outperform other state of the art algorithms for multitemporal aerial images. The simplicity of approach and application of integral images makes it easy to implement and use. As a part of future work the method can be extended to transfer global color characteristics to non-overlapping areas. Also, window size selection can be further improved to make it faster. A limitation to our proposed algorithm is that the images need to be registered with reasonable accuracy. However, for applications in change detection our approach fits well into any system.

7. Acknowledgment

The support for this study was provided in part by the Global Center of Excellence at TPU funded by the MEXT, Japan.

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006.

[2] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. In *International Journal of Computer Vision*, 2007.

[3] 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, sept. 2008.

[4] J. Jia and C. Tang. Tensor voting for image correction by global and local intensity alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1), 2005.

[5] S. J. Kim and M. Pollefeys. Robust radiometric calibration and vignetting correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:562–576, 2008.

[6] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application*, pages 331–340. INSTICC Press, 2009.

[7] M. Oliveira, A. D. Sappa, and V. Santos. Unsupervised local color correction for coarsely registered images. In *Computer Vision and Pattern Recognition*, 2011.

[8] F. Pitie, A. C. Kokaram, and R. Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1434 –1439 Vol. 2, oct. 2005.

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

[10] A. Sadeghi, M. Hejrati, and N. Gheissari. Poisson local color correction for image stitching. In *International Conference on Computer Vision Theory and Applications*, 2008.

[11] Y. W. Tai, J. Jia, and C. K. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *Computer Vision and Pattern Recognition*, 2005.

[12] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr 2004.

[13] J. A. Womble, K. C. Mehta, and B. J. Adams. Remote-sensing assessment of wind damage. In *5th International Workshop on Remote Sensing Applications to Natural Hazards*, 2007.

[14] W. Xu and J. Mulligan. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In *Computer Vision and Pattern Recognition*, 2010.

[15] M. Zhang and N. D. Georganas. Fast color correction using principal regions mapping in different color spaces. *Real Time Imaging*, pages 23–30, 2004.

Figure 5. *(1st column)* Source images. *(2nd column)* Target images. *(3rd column)* New target image transformed using gain compensation. *(4th column)* New target image transformed using the proposed adaptive windowing based local color transfer.