

Fusing Multiple Color Images for Texturing Models *

Nobuyuki Bannai, Alexander Agathos, and Robert B. Fisher
School of Informatics, University of Edinburgh
nbannai@inf.ed.ac.uk, A.Agathos@sms.ed.ac.uk, rbf@inf.ed.ac.uk

Abstract

A commonly encountered problem when creating 3D models of large real scenes is unnatural color texture fusion. Due to variations in lighting and camera settings (both manual and automatic), captured color texture maps of the same structure can have very different colors. When fusing multiple views to create larger models, this color variation leads to a poor appearance with odd color tilings on homogeneous surfaces. This paper extends previous research on pairwise global color correction to multiple overlapping images. The central idea is to estimate a set of blending transformations that minimize the overall color discrepancy in the overlapping regions, thus spreading residual color errors, rather than letting them accumulate.

1 Introduction

When a realistic 3D Computer Graphics model is constructed from real scenes, multiple color texture maps are often necessary. Taking images from different view angles and with different camera settings creates mismatching colors due to lighting and camera conditions. As some researchers [1, 2, 3] showed, texturing from two different view images causes color discrepancies to appear on the rendered surfaces. The discrepancies can include color as well as lightness shifts, and results in a mosaic appearance on the surfaces. To remove the color discrepancies, Agathos and Fisher [1] introduced a global correction method by estimating the RGB color transformation from pixels where the two texture maps overlap. That research only considered pairwise texture map corrections. Beachesne and Roy [3] introduced a relighting method by computing the lighting ratio of two textures. They explained that the method could be modified to adjust n textures, however, it also estimated the RGB color transformation from pixels where two textures overlap. Baumberg [2] introduced another approach, a multi-band blending technique. This method blends low and high band images by generating a weighting function. This means that the method needs more images than the others, even if a target is small. Of course, if high quality color constancy processes [6] were available then we could use them, however, these processes (1) still seem to need further development and (2) do not usually exploit the additional information that we have here, namely two color samples at corresponding points. When constructing a large 3D Computer Graphics model, especially for a building or a complex shape object, more than two different view images would be needed (*e.g.* 10 - 20 images for all sides of a building, each image with its own distortions) and multiple color image fusion would be unavoidable. Figure 1 shows a VRML model texture mapped using 30 images. The highlighted region consists of three different images seen from right, bottom and left directions. The dark oval surrounds the overlapping region. Each image has different lighting conditions and the texture mapped model has color discrepancies at the image boundaries.

In this paper we introduce a new approach to remove color discrepancies between multiple overlapping images. To remove the color discrepancies, the pairwise global correction method [1] was extended for multiple overlapping images. The central problem with direct application of the previous pairwise global correction method can be illustrated as follows. Suppose we have three overlapping textures, A , B and C . Suppose A & B , B & C and A & C overlap. Then, we can estimate the correction $T_{B \rightarrow A}$ from B to A and $T_{C \rightarrow A}$ from C to A . This means that we can have, in theory, the patches A , B and C all in the color space of A . Unfortunately, this does not guarantee that no color discontinuities exist where B and C overlap, *i.e.* $T_{B \rightarrow A} T_{C \rightarrow A}^{-1} \neq T_{B \rightarrow C}$. When more than three patches overlap, perhaps with a complex topology of overlaps, then the

*©IEEE 2004



Figure 1. A VRML model texture mapped using 30 images. The highlighted line region consists of three different images seen from right, bottom and left directions. The dark oval surrounds the overlapping region. The model has color discrepancies at the image boundaries.

problem becomes more complicated. Our proposed solution is to choose a patch X with a good appearance, and then find transformations $T_{Y \rightarrow X}$ ($Y \neq X$) that minimize the total color error over all overlapping pixels.

In order to test the extended method in a controlled manner, sets of test images were artificially created by dividing real images into three fragments to have an overlapping region. The color balance was changed arbitrarily and noise (Gaussian) was added to the test set. The extended method was applied to the test sets and sets of blending transformations that successfully minimized the overall color discrepancy in the overlapping region were estimated. Finally, we demonstrate the extended method with real data sets obtained with a digital camera with different views and camera settings. Since the real data sets had perspective and lens distortions, a common image rectification method (Homography Transformation) [4, 5] was used to remove them and place corresponding pixels in the overlapping region correctly. In addition, some of images had different luminous intensity between their overlapping region and texture mapping region due to highlights and shadows. To remove the luminous intensity difference, a patch based color correction shown in Section 2.1 was used and the luminous intensity was adjusted before the extended method was applied to the test sets.

2 Method

In this section we present a color correction method for N overlapping images. Assume we have N patches $p_1 \dots p_N$, each of which overlaps with one or more other patches. This forms a graph of patches, where the nodes represent patches and arcs indicate an overlap. Suppose a circuit in the graph exists, say $A - B - C - \dots - Z - A$ where A is the patch to register to. Then we may have a problem as the color alignment of Z to A via B & C etc. may be different from that of Z to A directly. *I.e.*, a sort of systematic accumulation of bias might occur. What we propose here is an algorithm to spread the average error over all arcs of the graph.

When taking a photo, the photo often has perspective and lens distortions. To place corresponding pixels in the overlapping region correctly, the distortions should be removed before the method is applied to the images. In addition, the images have luminous intensity differences between the overlapping region and the texture mapping region. Since our color correction approach uses corresponding pixel values in the overlapping region, the differences should be adjusted in advance, too. Thus, our color correction process is as follows: 1) Rectify and identify corresponding pixels in the overlapping region; 2) Estimate color transform matrices; 3) Transform color using the matrices.

2.1 Image Rectification and Luminosity Adjustment (Patch Based Color Correction)

The color correction method presented below requires corresponding color pixels. We can achieve this by two methods: 1) acquire the color in 3D, where each 3D point acquired from a range sensor (or stereo) has an attached RGB value, or 2) find corresponding pixels in texture map images. The latter is the process used here, but the process below is applicable to both sensors.

First, to remove perspective distortions and place corresponding pixels in the overlapping region, a common image rectification method, the Homography Transformation [4, 5], is used. Suppose p_i and P_i are points in homogeneous coordinates in an original image and a transformed image respectively, linked by the homography H (3 x 3 matrix). Then:

$$P_i = Hp_i \quad (1)$$

By computing the matrix H and transforming an image with H , aligned images can be obtained.

The method applied in Section 2.3 assumes that all corrected surfaces have the same illumination conditions. This is not always the case, so we first do an adjustment on pairwise image texture maps. When two regions are expected to have the same color within each image, but do not due to difference in illumination, we apply the following initial luminosity correction. In this case we do not have exactly corresponding pixels, so we choose pixels from patches that should have the same color.

Suppose images A and B are different view images and include luminosity differences. Suppose $PA_1, PA_2, PA_3, PB_1, PB_2$ and PB_3 are patches within images A and B . Suppose PA_1 overlaps PB_1 , PA_2 overlaps PB_2 , PA_3 does not overlap PB_3 . Suppose all patches have the same color. Second, to adjust luminosity, we need color transformations $T_{B \rightarrow A}$ (from image B to image A), $T_{A_2 \rightarrow A_1}$ (from PA_2 to PA_1), $T_{A_3 \rightarrow A_1}$ (from PA_3 to PA_1), $T_{B_2 \rightarrow B_1}$ (from PB_2 to PB_1), $T_{B_3 \rightarrow B_1}$ (from PB_3 to PB_1) that minimize the total color error E_1 over all patch pixels. Figure 2 illustrates these relations. We named this method: patch based color correction. The color error to minimize is as follows:

$$\begin{aligned} E_1 = & 1/N_1 \sum_i \|p\vec{a}1_i - T_{B \rightarrow A}p\vec{b}1_i\|^2 \\ & + 1/N_2 \sum_i \|T_{A_2 \rightarrow A_1}p\vec{a}2_i - T_{B \rightarrow A}T_{B_2 \rightarrow B_1}p\vec{b}2_i\|^2 \\ & + \|T_{A_3 \rightarrow A_1}p\vec{a}3_{mean} - T_{B \rightarrow A}T_{B_3 \rightarrow B_1}p\vec{b}3_{mean}\|^2 \\ & + \|T_{A_3 \rightarrow A_1}p\vec{a}3_{mean} - p\vec{a}1_{mean}\|^2 \\ & + \|T_{A_2 \rightarrow A_1}p\vec{a}2_{mean} - p\vec{a}1_{mean}\|^2 \\ & + \|T_{B \rightarrow A}T_{B_3 \rightarrow B_1}p\vec{b}3_{mean} - T_{B \rightarrow A}p\vec{b}1_{mean}\|^2 \\ & + \|T_{B \rightarrow A}T_{B_2 \rightarrow B_1}p\vec{b}2_{mean} - T_{B \rightarrow A}p\vec{b}1_{mean}\|^2 \end{aligned} \quad (2)$$

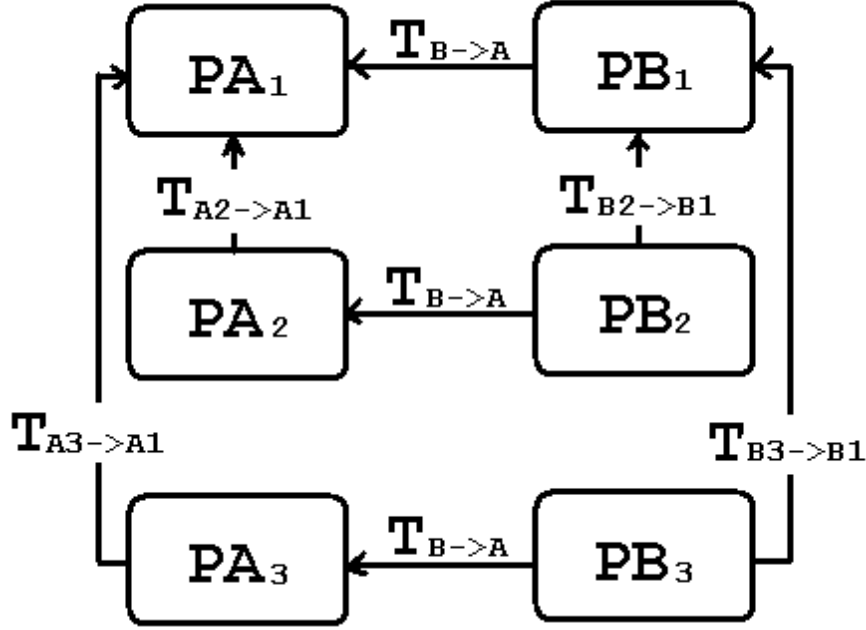


Figure 2. Color transformation flows of a patch based color correction.

where, N_1 and N_2 are the numbers of corresponding pixels in patch PA_1 (or PB_1) and PA_2 (or PB_2). \vec{pa}_i , \vec{pa}_i and \vec{pa}_i are the RGB color of pixel i in patches PA_1 , PA_2 and PA_3 . \vec{pb}_i , \vec{pb}_i and \vec{pb}_i are the RGB color of the corresponding pixel i in patches PB_1 , PB_2 and PB_3 . \vec{pa}_{mean} , \vec{pa}_{mean} , \vec{pa}_{mean} , \vec{pb}_{mean} , \vec{pb}_{mean} and \vec{pb}_{mean} are the mean RGB color pixel in patches PA_1 , PA_2 , PA_3 , PB_1 , PB_2 and PB_3 .

We apply these ideas to adjust the color and luminosity of all patches within an image that should have the same appearance (variations due to illumination (e.g. $PA_1 : PA_2$)), non-identical patches seen in two images (e.g. $PA_3 : PB_3$), as well as the appearance of overlapping patches (e.g. $PA_1 : PB_1$).

This process does an initial coarse correction of the luminosity and initial color correction. Both of these will be adjusted again in the process introduced in Section 2.3

2.2 Approximate Color Correction

Before using the color correction method, estimating good initial color transformations $S_{X \rightarrow A} (X \neq A)$ is necessary to reduce the number of calculations and to find best solutions easily. To do this, we used a pairwise color correction based on [1] when images do not have luminosity differences. Otherwise, the identity matrix was used as the initial color transformation, since the initial global color correction was already done in Section 2.1.

2.3 Color Correction Over All Images

Now we are ready to correct color discrepancies between N partially overlapping images. For simplicity, we start with 3 images A , B and C , but the theory below applies for any arbitrary topology of overlapping regions. Suppose A & B , B & C and A & C overlap and A has a good color appearance. Then, A 's color space will be the target and B 's and C 's color spaces should be transformed to A 's color space. We could estimate the color transformations $T_{B \rightarrow A}$ from B to A and $T_{C \rightarrow A}$ from C to A by using corresponding pixel values in the overlapping region and then using the global correction method of [1]. However, the transformations do not guarantee that there is no color discrepancy between B and C . To solve this problem, we propose finding color transformations $T_{X \rightarrow A}$ that minimize the total color error E_2 over all overlapping

pixels. The equation is as follows:

$$E_2 = \sum_{i,j,k(i \neq k)} f(\|T_{i \rightarrow A} c_{ji}^{\vec{r}} - T_{k \rightarrow A} c_{jk}^{\vec{r}}\|^2) \phi(i, k, j) \quad (3)$$

where A is the target color space image. $c_{ji}^{\vec{r}}$ is the RGB color of pixel j in image i . $c_{jk}^{\vec{r}}$ is the RGB color of pixel j in image k . $\phi(i, k, j) = 1$ if pixel j is seen in both image i and image k and 0 otherwise.

$$f(x) = \begin{cases} x & (x < \tau) \\ 0 & (otherwise) \end{cases} \quad (4)$$

as this robustifies the error against outliers. This robustification by use of a threshold τ is possible because the color spaces are almost aligned already by the process in Section 2.2. Initial T matrices are estimated as S matrices by the pairwise color correction method.

This formula computes the error in color matching between two samples $c_{ji}^{\vec{r}}$ and $c_{jk}^{\vec{r}}$ at pixel j , which is then summed over all corresponding samples in all overlapping images. The goal is to spread the color matching error across all images, so we use the squared error, which penalizes large discrepancies.

$T_{A \rightarrow A}$ is fixed to be the identity matrix I_3 and no optimization affects it. This also prevents the optimization from converging to the degenerate solution $T_{X \rightarrow A} = 0$.

2.4 Color Space Error Minimization

To estimate color transformations that minimize the total color error E over all overlapping pixels, the *MATLAB* minimization function *fminsearch* was used. The entries of the transformation matrices T were unfolded to form the vector used by *MATLAB*. In addition, to reduce calculations, subsampled 5 - 10 % pixels in the overlapping region were chosen and the best solution was estimated by calculating the total color error E from the subsamples.

3 Results and Discussion

You really need to see the results in color because of the correction of the chromaticity. See URL: <http://www.ipab.informatics.ed.ac.uk/mvu/>

3.1 Experiments with Artificial Data Sets

Before the method was applied to real data sets, it was tested with artificially perturbed data sets. The data sets were created by dividing a base image into 3 fragments, top (A), bottom left (B) and bottom right (C), with *Microsoft Photo Editor*. For example, see Figures 3 and 5. These images only show the overlapping regions. The images in the top and bottom rows in both figures show test set before and after the color correction respectively. The first data set (Figure 3) just had its color balance, brightness and contrast changed and the other had some Gaussian noise added. The leftmost and middle images are slightly transformed versions of the rightmost image. Figures 4 and 6 show RGB color difference histograms before and after color correction. The top rows show RGB color difference histograms between the target (rightmost) image and the leftmost image. The middle ones show RGB color difference histograms between the target image and the middle image. The bottom ones show RGB color difference histograms between the leftmost and middle images. The dotted and solid lines in the histograms show the differences before and after color correction respectively. The dashed ones show those after the initial color correction, *i.e.* the pairwise color correction based on [1]. In addition, color transformation matrices were estimated as the pairwise ($S_{B \rightarrow A}$ and $S_{C \rightarrow A}$) and multiple ($T_{B \rightarrow A}$ and $T_{C \rightarrow A}$) color corrections for each data set as follows:

Estimated for data sets without noise:

$$S_{B \rightarrow A} = \begin{pmatrix} 0.7355 & 0.3454 & -0.1235 \\ 0.2401 & 0.7099 & 0.0908 \\ -0.0957 & 0.1060 & 0.9123 \end{pmatrix}$$

$$S_{C \rightarrow A} = \begin{pmatrix} 0.7453 & 0.3568 & -0.1410 \\ 0.2098 & 0.6845 & 0.1408 \\ -0.0695 & 0.1151 & 1.0186 \end{pmatrix}$$

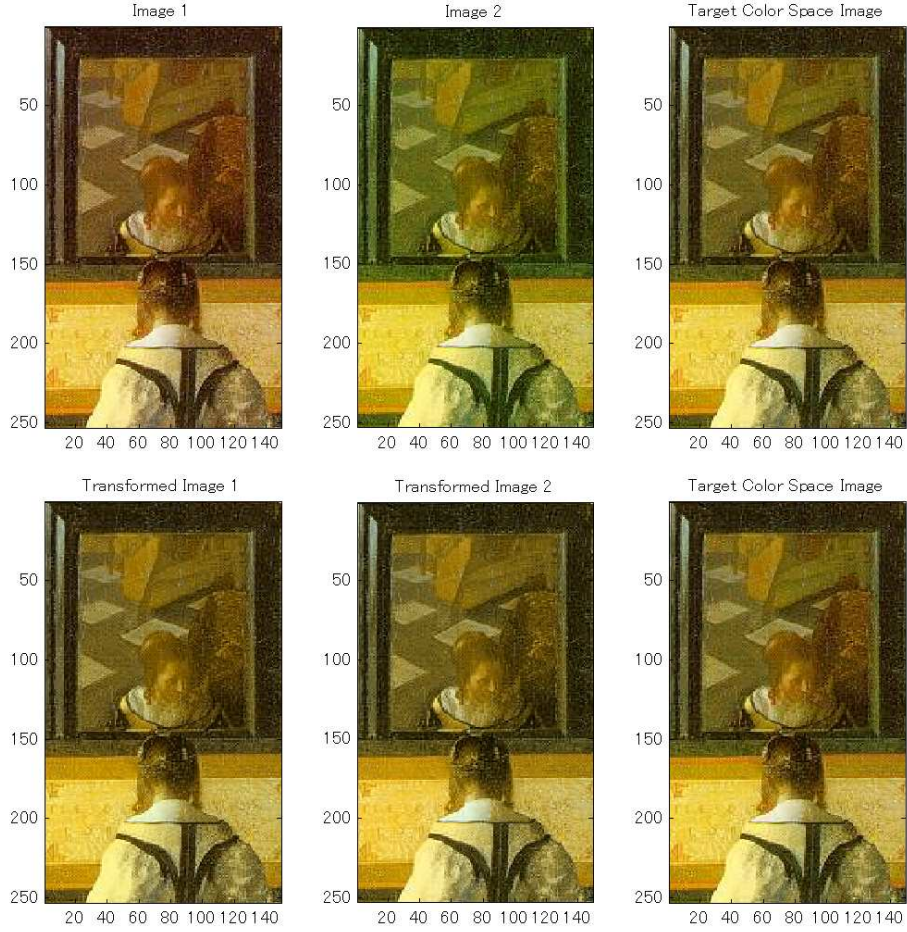


Figure 3. The top row images show test set, image 1 (left), 2 (middle) and a target color space one (right), in the overlapping region before the color correction. The bottom row shows the result images after the color correction. No noise is added to the test data set.

$$T_{B \rightarrow A} = \begin{pmatrix} 0.6596 & 0.4628 & -0.1692 \\ 0.3219 & 0.5848 & 0.1371 \\ -0.1301 & 0.1607 & 0.8878 \end{pmatrix}$$

$$T_{C \rightarrow A} = \begin{pmatrix} 0.6316 & 0.4916 & -0.1881 \\ 0.2865 & 0.5933 & 0.1727 \\ -0.0866 & 0.1372 & 1.0058 \end{pmatrix}$$

Estimated for data sets with noise:

$$S_{B \rightarrow A} = \begin{pmatrix} 0.6377 & 0.4625 & -0.0859 \\ 0.3537 & 0.5295 & 0.1528 \\ -0.0455 & 0.0475 & 0.7715 \end{pmatrix}$$

$$S_{C \rightarrow A} = \begin{pmatrix} 0.7164 & 0.3591 & -0.0747 \\ 0.3717 & 0.4955 & 0.1956 \\ -0.0052 & 0.0327 & 1.0189 \end{pmatrix}$$

$$T_{B \rightarrow A} = \begin{pmatrix} 0.5889 & 0.5082 & -0.0661 \\ 0.3791 & 0.4842 & 0.1738 \\ -0.0429 & 0.0708 & 0.7280 \end{pmatrix}$$

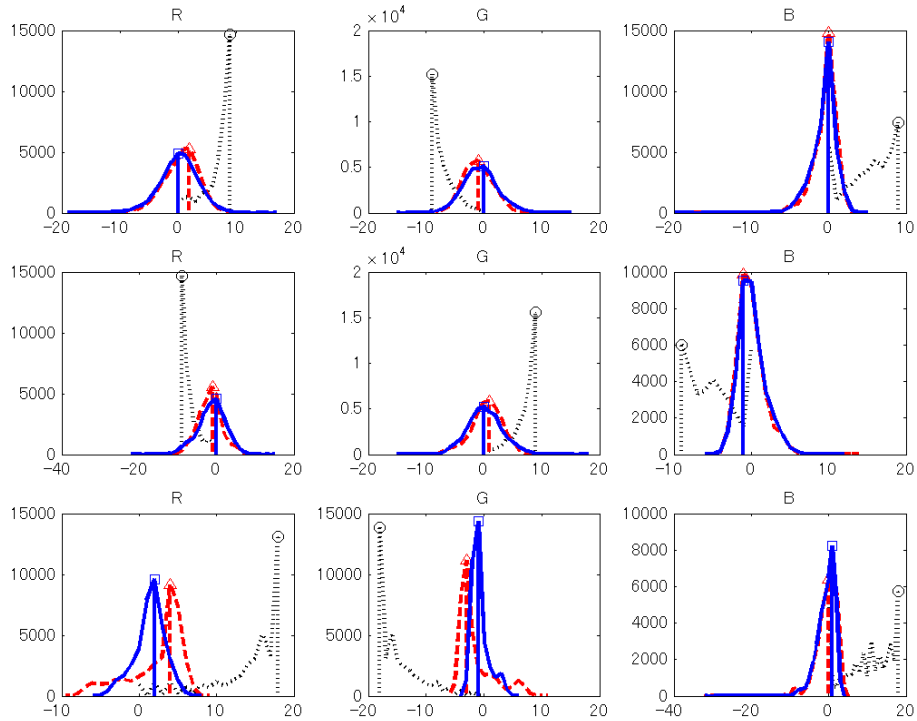


Figure 4. RGB Color difference histograms of the artificial data sets without noise before and after the color corrections. The top row graphs show RGB color difference histograms between the target image and image 1. The middle row shows those between the target image and image 2. The bottom row shows those between images 1 and 2. The dotted and solid lines in the histograms show RGB color differences before and after the color correction. The dashed ones show those after the initial (pairwise) color correction on [1]. As you can see in this figure, all peaks have moved to approximately 0 (no color difference point) after the color corrections. The color differences between images 1 and 2 after the final color correction are smaller than after the initial color correction, though the color differences between the target image and image 1, image 2 are almost same after optimization. *I.e.* the optimization process improves the overall global color balance without ruining the pairwise adjustment.



Figure 5. The top row images show test set, image 1 (left), 2 (middle) and a target color space image (right), in the overlapping region before the color correction. The bottom row shows the result images after the color correction. Gaussian noise is added to the reference color space images.

$$T_{C \rightarrow A} = \begin{pmatrix} 0.6598 & 0.4072 & -0.0491 \\ 0.4053 & 0.4437 & 0.2334 \\ -0.0008 & 0.0501 & 0.9620 \end{pmatrix}$$

As you can see in Figures 3 and 4, color discrepancies were almost removed and peaks of the histograms were moved to approximately 0 (no color difference point) after the color correction. In addition, RGB color differences between the leftmost and middle images after the color correction is smaller than that of the initial (pairwise) color correction, though the RGB color differences between the three images are almost same. This means that the proposed color correction method has an advantage. A similar result was also obtained in experiments with test set with Gaussian noise. In Figures 5 and 6, color discrepancies were almost removed and peaks of the histograms were moved to approximately 0 as well. This means that the color correction has tolerance for noise, *e.g.* image blur, pixel mismatching, *etc.*

3.2 Experiments with Real Data Sets

We show in Figure 7 a VRML model textured with three original images before and after the color correction method was applied. The squares in the left image show color discrepancies at image boundaries. The circle shows the overlapping region in which pixels were used to estimate color transformation matrices. As you can see in the right figure, the color correction method effectively removes color discrepancies. However, the red color region at the right face (the circle in the right in

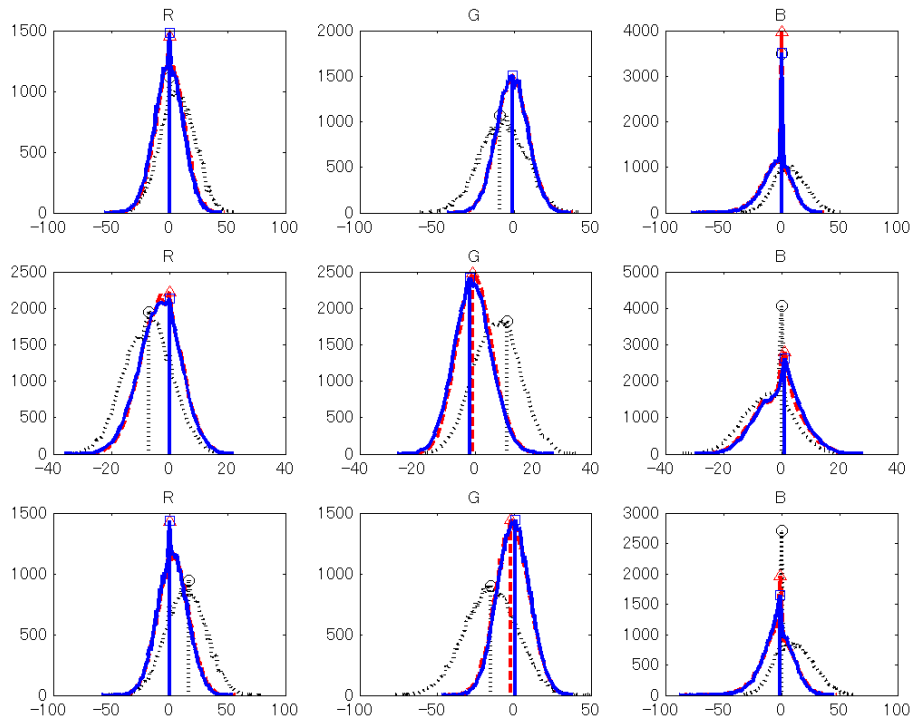


Figure 6. RGB Color difference histograms of the artificial data sets with noise before and after the color corrections. The top row graphs show RGB color difference histograms between a target image and image 1. The middle row shows those between the target image and image 2. The bottom row shows those between images 1 and 2. The dotted and solid lines in the histograms show RGB color differences before and after the color correction. The dashed ones show those after the initial (pairwise) color correction on [1]. As you can see in this figure, all peaks of the histograms have moved to approximately 0 (no color difference point) after the color corrections. The color differences between images 1 and 2 after the final color correction are almost same as after the initial color correction, but the means are generally closer to 0.

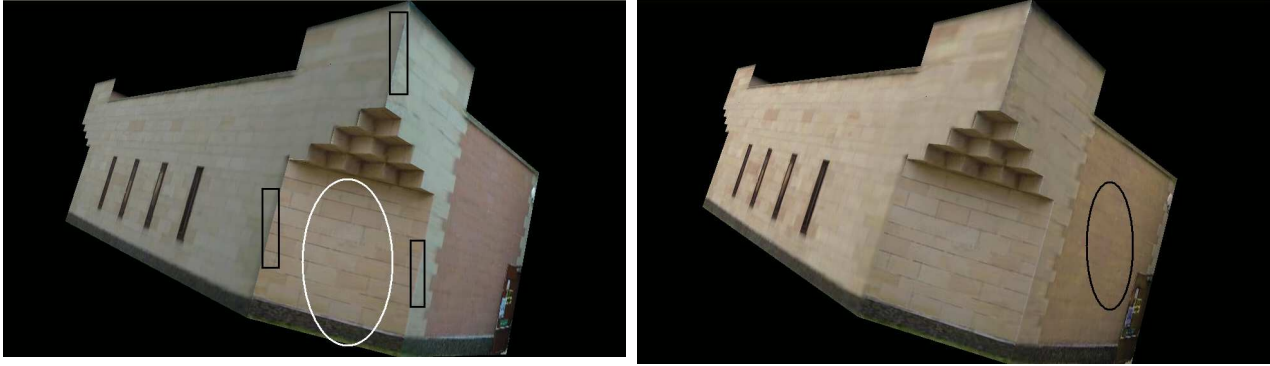


Figure 7. A VRML model textured with three original images (left) and color corrected images (right). The squares in the left image show color discrepancies at image boundaries. The circle shows the overlapping region in which pixels were used to estimate color transformation matrices. Color discrepancies between images were almost removed after the color correction, though the red color region in the right side (the circle in the right image) was a bit lighter.

Figure 7) was a bit lighter.

Figure 8 shows a VRML model textured with 30 original images and color corrected images. The color correction method was applied to remove color discrepancies. Color discrepancies were almost removed and a good appearance model was constructed.

3.3 Comparing the pairwise with multiple image color correction

To compare effects of the pairwise image color correction with those of the multiple image color correction, Figure 9 shows another VRML model textured with eight original images for all sides of a small tower. Image numbering starts from the top leftmost image in a clockwise direction. The front face of the tower (the top leftmost image in Figures 9, 10 and 11) is the target color space and the other image color spaces were transformed to the front face's by using the color correction methods. All object face edges are overlapping regions. The overlapping surfaces used are the thin faces of the tower. Overlapping region 1 is seen in images 1, 2 and 3. Similarly overlapping region 2 is in images 3, 4 and 5, overlapping region 3 is in images 5, 6 and 7 and overlapping region 4 is in images 7, 8 and 1. The results with the multiple and the pairwise image color correction are in Figures 10 and 11 respectively. Figure 12 shows RGB color difference histograms between the front face of the tower and overlapping region 4 after the pairwise (dashed line) and multiple (solid line) image color corrections. As you can see in Figures 10 and 11, color discrepancies at image boundaries were almost removed after the color corrections. However, the pairwise image color correction gradually increased errors and there is a large dissimilarity between the front face and overlapping region 4 after the pairwise color image correction. On the other hand, the multiple image color correction prevented the error from accumulating and there is less dissimilarity between the front face and overlapping region 4 after the multiple image color correction, though the white color region in image 5 (the square in the bottom rightmost image in Figure 10) became a bit grayish. In addition, the histograms show that RGB color differences with the multiple image color correction are smaller than those of the pairwise image color correction.

It takes about 20 minutes to compute the color transformation matrices for the multiple color image correction with 500 overlapping pixels with a laptop computer (Intel Pentium III 1 GHz CPU, 256MB RAM, 40GB HDD) running MATLAB under Microsoft Windows XP Home Edition. Our algorithm is linear in the number of overlapping pixels. In theory it is quadratic in the number of images, but in practice it is nearly linear because each patch only overlaps with a few others.

The result is not perfect because the image rectification and luminosity adjustment may not be enough. Moreover, the subsampled pixels which were chosen to estimate color transformation matrices and the number of them may not be appropriate. However, we could decrease the subsampling rate and color discrepancies are almost removed across all images.



Figure 8. A VRML model textured with 30 original images (left) and color corrected images (right). The color correction method was applied several times. The appearance of all stone surfaces is much more consistent than the original.



Figure 9. A VRML model textured with eight real images before the color correction. Image 1 is the top leftmost image. The image numbering starts from the top leftmost in a clockwise direction. There are some color discrepancies at image boundaries.



Figure 10. A VRML model textured with eight real images after the multiple color correction. Image 1 is the top leftmost image. The image numbering starts from the top leftmost in a clockwise direction. Color discrepancies at image boundaries were almost removed. There is less dissimilarity between the front face of the tower and overlapping region 4, though the white color region (the square) in image 5 became a bit grayish.



Figure 11. A VRML model textured with eight real images after the pairwise color correction. Image 1 is the top leftmost image. The image numbering starts from the top leftmost in a clockwise direction. Color discrepancies at image boundaries were almost removed. However, there is a large dissimilarity between the front face of the tower and overlapping region 4 due to the error accumulation.

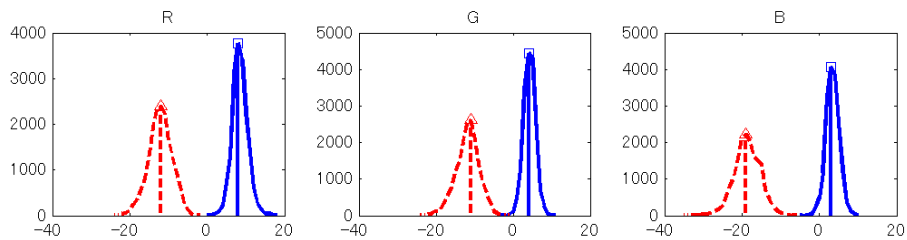


Figure 12. RGB color difference histograms between the front face of the tower and overlapping region 4 after the pairwise (dashed line) and multiple (solid line) image color correction. As you can see in the histograms, RGB color differences with the multiple image color correction (solid line) are smaller than those of the pairwise image color correction (dashed line).

3.4 Summary and Conclusions

We summarize and conclude this work as follows:

- A color correction method for multiple view textures having overlapping regions was proposed and some results were shown.
- The proposed method works as supported by both the visual evidence and histogram analysis.
- To obtain a good result, a good image registration and high resolution images are necessary.
- Highlights and shadows are still problems when removing color discrepancies correctly. Shadows may be removed by using Finlayson's technique [7].
- Overlapping regions should have a full range of colors. Otherwise, the color corrected images may include strange colors outside the overlapping regions because these colors are unconstrained.

To improve this method, precise pixel matchings in overlapping regions (a good image rectification) and a good luminosity adjustment may be required. Moreover, good overlapping region selection may be needed. So, developing these methods are future areas of this work.

Acknowledgements

Many thanks to *CANON INC.* for support on this project.

References

- [1] A.Agathos and R.B.Fisher. *Colour Texture Fusion of Multiple Range Images*. Proc. 4th Int. Conf. on 3-D Digital Imaging and Modeling, Banff, pp 139-146, 2003.
- [2] A.Baumberg. *Blending images for texturing 3D models*. Proc. Conf. on British Machine Vision Association, Cardiff, pp 404-413, 2002.
- [3] E.Beauchesne and S.Roy. *Automatic Relighting of Overlapping Textures of a 3D Model*. Proc. Conf. on Computer Vision and Pattern Recognition, Madison, Vol. 2, pp 139-146, 2003.
- [4] I.Zoghalmi, O.Faugeras and R.Deriche. *Using geometric corners to build a 2D mosaic from a set of images*. Proc. Conf. on Computer Vision and Pattern Recognition, pp 420-425, 1997.
- [5] P.Heckbert. *Projective Mappings for Image Warping*. Master's thesis, University of California, Berkeley, 1989. Retrieved: December 2003, available from: <http://www-2.cs.cmu.edu/ph/869/www/notes/proj/proj.pdf>. School of Computer Science, Carnegie Mellon University, 2001.
- [6] K.Barnard, G.Finlayson and B.Funt. *Colour Constancy for Scenes with Varying Illumination*. A survey, Computer Vision and Image Understanding, Vol. 65(2), pp 311-321, 1997.
- [7] G.D.Finlayson, M.S.Drew and C.Lu. *Intrinsic Images by Entropy Minimization*. Proc. 8th European Conf. on Computer Vision, Prague, pp 582-595, 2004.