Color Transfer Using Probabilistic Moving Least Squares

Youngbae Hwang*
KETI
ybhwang@keti.re.kr

Joon-Young Lee*
KAIST
jylee@rcv.kaist.ac.kr

In So Kweon
KAIST
iskweon@kaist.ac.kr

Seon Joo Kim
Yonsei University
seonjookim@yonsei.ac.kr

(* denotes co-first authors.)

Abstract

This paper introduces a new color transfer method which is a process of transferring color of an image to match the color of another image of the same scene. The color of a scene may vary from image to image because the photographs are taken at different times, with different cameras, and under different camera settings. To solve for a full nonlinear and nonparametric color mapping in the 3D RGB color space, we propose a scattered point interpolation scheme using moving least squares and strengthen it with a probabilistic modeling of the color transfer in the 3D color space to deal with mis-alignments and noise. Experiments show the effectiveness of our method over previous color transfer methods both quantitatively and qualitatively. In addition, our framework can be applied for various instances of color transfer such as transferring color between different camera models, camera settings, and illumination conditions, as well as for video color transfers.

1. Introduction

Color of a scene may vary from image to image because the photographs are taken at different times (illumination change), with different cameras (camera spectral sensitivity change), and under different camera settings (in-camera imaging parameter change [10]) (Fig. 1). Photographs of a scene may also vary due to different photographic adjustment styles of the users [4].

In general, color transfer refers to the process of transforming color of an image so that the color becomes consistent with the color of another image.[1] Color transfer is applied to many computer vision and graphics problems. One main application is the computational color constancy in which the color is transferred to remove the color cast by the illumination [2]. It is also used to generate color consistent image panoramas and 3D texture-maps [11, 14, 23], as well as to enhance and manipulate images by emulating the tone and the color style of other images [6, 16, 18].

[1]The process is also called as color correction, color mapping, or photometric alignment.

Different camera types (iPhone vs. Canon) (Building)

Different camera settings (Sculpture)

Illumination changes (Campus)

Photographic retouches (Gangnam1)

Figure 1: Changes in image color due to different factors. The images in the third column are our results of transferring the color of the images in the first column to that of the second column.

The goal of this paper is to introduce a new mechanism for transferring color between images. We are particularly interested in employing a full nonlinear and nonparametric color mapping in the 3D RGB color space instead of using a linear color transformation, modeling color channels separately, or matching statistical color measures (mean and variance) between images in an uncorrelated color space. Utilizing a full 3D color transformation is especially useful for explaining the in-camera imaging pipeline which was recently introduced in [10]. To solve the nonparametric 3D color transfer problem, we employ a scattered point interpolation scheme based on moving least squares and make it more robust by combining it with a probabilistic modeling of the color transfer. Our framework can be applied for

various instances of color transfer such as transferring color between different camera models (e.g. iPhone and a Canon DSLR) and camera settings (e.g. white balance and picture styles), illumination conditions, and photographic retouch styles as shown in Fig. 1.

2. Related Work

2.1. Color Transfer

Given an RGB value $\mathbf{x} = [r, g, b]^\top$, the most commonly used method for transferring the color is to apply a linear transformation: $\mathbf{x}' = \mathbf{M}\mathbf{x}$, where $\mathbf{M}$ is a 3×3 matrix describing the mapping of the three color channel values. Although the matrix $\mathbf{M}$ can be of any arbitrary form, a simple diagonal model is used more often than not, especially in the computational color constancy work [2]. While the linear transformation model provides a simple yet effective way to transform colors, it shows clear limitations in explaining the complicated nonlinear transformations in the imaging process. Another method for transferring color is to use a general polynomial model [7]: $\mathbf{x}' = \mathbf{M}' \cdot [r^n, g^n, b^n, ..., r, g, b, 1]^\top$. By introducing higher degree terms, the polynomial model can compensate for the nonlinearities better than the linear model.

Another popular color transfer method is based on the statistics of the color distribution in images, first proposed by Reinhard et al. in [16]. The images are transformed to the uncorrelated $l\alpha\beta$ space and the color transform is computed by matching the means and the standard deviations of the global color distributions of the images. This approach served as the baseline for other following color transfer works such as in [14, 15, 18]. While this statistical approach is effective in transferring the look and the feel of the image color (which is good enough for some applications), it may not be practical for photometrically aligning different color values accurately.

Some other color transfer methods include computing the brightness transfer from 2D joint histograms of registered images [11] and a 2D tensor voting scheme [8]. For more detailed comparisons and evaluations of different color transfer methods, we refer the readers to [23].

Compared to the previous methods described above, the main contribution of this paper is that we present a nonlinear and nonparametric color transfer framework that operates in a 3D color space. This new framework for color transfer fits well with the in-camera imaging process recently introduced in [10], where it was shown that the color values are processed in a highly nonlinear fashion in the 3D color space due to components such tone-mapping and gamut mapping. Experiments show that our method can align colors much more accurately than the other frameworks and the proposed method is general enough to be used for many different applications.

2.2. Moving Least Squares

Moving least squares (MLS) is a scattered point interpolation technique first introduced in [12] to generate surfaces. Using the MLS method, one can reconstruct a continuous function from a set of point samples by incorporating the weighted least squares scheme, which gives more weights to those samples that are closer to the point being reconstructed (see Fig. 2(a)). The MLS approach has been successfully used for image deformation [17], surface reconstruction [5], and image superresolution and denoising [1]. In this paper, we apply the MLS method to the color transfer problem. To fit the MLS to the color transfer problem, we further incorporate a probabilistic measure to the MLS to strengthen the performance and add a parallel processing scheme to increase the computational efficiency. To the best of our knowledge, this is the first attempt to employ the MLS framework for transferring color.

3. Color Transfer Algorithm using Probabilistic Moving Least Squares

We introduce a mechanism for transforming color given a set of correspondences between a pair of images I and J. By employing a nonlinear and nonparametric method, we can model various sources of color changes between images without targeting a specific form of the color change (e.g. exposure change, illumination change, etc.) in addition to modeling the color change more accurately compared to parametric methods such as the linear 3×3 mapping and the distribution matching [16].

Fig. 2 sketches the idea behind our Probabilistic Moving Least Squares (PMLS) framework for color transfer. Using the MLS framework, the color transfer can be computed for each input color by considering the distance of the input color to the control points, which are in our case a set of corresponding color points (Fig. 2(a)). A set of corresponding points given by an image registration algorithm can include many outliers, which can in turn adversely influence the MLS results. To deal with mismatches and noise, we employ a probabilistic approach by computing the probability of the bidirectional color transfer between the corresponding control points (Fig. 2(b)) and combining it with the MLS. The weights for the scattered point interpolation using PMLS now depends on not only the distance but also the probability of the color transfer between the corresponding points (Fig. 2(c)). We now explain the details of our algorithm in the following subsections.

3.1. Moving Least Squares Framework

Let $\mathbf{u}$ and $\mathbf{v}$ be the sets of the corresponding pixel values ($[r, g, b]^\top$) for images I and J respectively, and they will serve as the control points of the MLS algorithm. Given an RGB value $\mathbf{x}$ in image I, we solve for the transformation

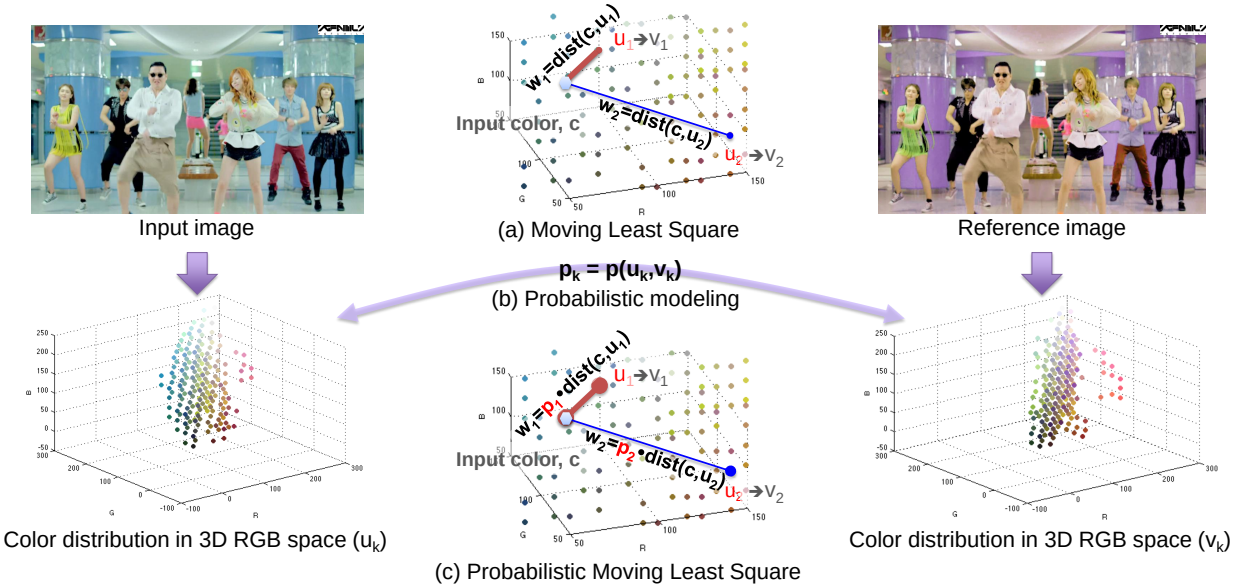


Figure 2: The PMLS concept. (a) The original MLS framework, the thickness of the line indicates the distance weight. (b) The probability of the bidirectional color transfer between the corresponding control points (Eq. 13). (c) The proposed PMLS framework. The thickness of the line indicates the distance weight and the size of the circle is proportional to the probability of the color transfer between the corresponding points.

$T_{\mathbf{x}}$ that minimizes

$$\sum_{k=1}^{m} w_k \left| T_{\mathbf{x}}(\mathbf{u}_k) - \mathbf{v}_k \right|^2, \tag{1}$$

where m is the number of control points and w_k is the weight defined as

$$w_k = \frac{1}{\left| \mathbf{u}_k - \mathbf{x} \right|^{2\alpha}}. \tag{2}$$

The name *Moving Least Squares* comes from the fact that the weights w_k in this least squares problem change depending on the color $\mathbf{x}$ to be evaluated [17]. Therefore, the transformation $T_{\mathbf{x}}$ also varies for each $\mathbf{x}$.

For image deformation work [17], a rigid transformation $\mathbf{T}_{\mathbf{x}}$ was chosen as the most realistic transformation since it maintains the geometric properties with less degrees of freedom. However, for our color transfer work, the transformation should be general as to model different elements of the color deformation such as illumination change, nonlinearities in the camera pipeline, or photo-editing by a user. Therefore, we choose the following affine transformation for $T_{\mathbf{x}}$:

$$T_{\mathbf{x}}(\mathbf{x}) = \mathbf{A}_{\mathbf{x}}\mathbf{x} + \mathbf{b}_{\mathbf{x}}, \tag{3}$$

where $\mathbf{A}_{\mathbf{x}}$ is a full 3×3 matrix and $\mathbf{b}_{\mathbf{x}}$ represents a translation.

By taking the partial derivative of Eq. 1 with respect to $\mathbf{b}_{\mathbf{x}}$, we get

$$\mathbf{b}_{\mathbf{x}} = \overline{\mathbf{v}} - \mathbf{A}_{\mathbf{x}}\overline{\mathbf{u}}, \tag{4}$$

with $\overline{\mathbf{u}}$ and $\overline{\mathbf{v}}$ being the weighted centroids,

$$\overline{\mathbf{u}} = \frac{\sum_k w_k \mathbf{u}_k}{\sum_k w_k}, \quad \overline{\mathbf{v}} = \frac{\sum_k w_k \mathbf{v}_k}{\sum_k w_k}.$$

Eq. 3 can then be rewritten as

$$T_{\mathbf{x}}(\mathbf{x}) = \mathbf{A}_{\mathbf{x}}(\mathbf{x} - \overline{\mathbf{u}}) + \overline{\mathbf{v}}, \tag{5}$$

and Eq. 1 becomes

$$\sum_k w_k \left| \mathbf{A}_{\mathbf{x}}\hat{\mathbf{u}}_k - \hat{\mathbf{v}}_k \right|^2, \tag{6}$$

where $\hat{\mathbf{u}}_k = \mathbf{u}_k - \overline{\mathbf{u}}$ and $\hat{\mathbf{v}}_k = \mathbf{v}_k - \overline{\mathbf{v}}$.

$\mathbf{A}_{\mathbf{x}}$ can be computed by minimizing Eq. 6 as

$$\mathbf{A}_{\mathbf{x}} = \left(\sum_k w_k \hat{\mathbf{u}}_k \hat{\mathbf{u}}_k^\top \right)^{-1} \sum_k w_k \hat{\mathbf{u}}_k \hat{\mathbf{v}}_k^\top. \tag{7}$$

After computing $\mathbf{A}_{\mathbf{x}}$ and $\mathbf{b}_{\mathbf{x}}$, the color $\mathbf{x}$ in image I to be evaluated is transformed to the color $\mathbf{A}_{\mathbf{x}}(\mathbf{x} - \overline{\mathbf{u}}) + \overline{\mathbf{v}}$.

In the MLS framework, the accuracy of the color transfer would depend on the number of control points. Since increasing the number of control points would also increase the computational load, we devised a parallel processing scheme to speed up the color transfer computation. The summations in Eq. 7 are not well suited for the parallelization, so we rewrite the equation as follows:

$$\begin{aligned} \mathbf{A}_{\mathbf{x}} &= \mathrm{ivec}(\mathbf{w}^\top \mathbf{y_1})^{-1} \times \mathrm{ivec}(\mathbf{w}^\top \mathbf{y_2}) \\ \text{s.t.} \quad \mathbf{y_1} &= [\mathrm{vec}(\hat{\mathbf{u}}_1 \hat{\mathbf{u}}_1^\top)^\top, \cdots, \mathrm{vec}(\hat{\mathbf{u}}_k \hat{\mathbf{u}}_k^\top)^\top]^\top, \\ \mathbf{y_2} &= [\mathrm{vec}(\hat{\mathbf{u}}_1 \hat{\mathbf{v}}_1^\top)^\top, \cdots, \mathrm{vec}(\hat{\mathbf{u}}_k \hat{\mathbf{v}}_k^\top)^\top]^\top, \end{aligned} \tag{8}$$

(a) Reference image (aligned) (b) Input image (c) MLS (d) PMLS (Sec. 3.3) (e) PMLS with extrapolation (Sec. 3.4)

Figure 3: MLS vs. PMLS. Matching errors adversely affect the performance of the MLS algorithm, while the PMLS can absorb the matching errors due to the probabilistic modeling (d). Color transfer in the non-overlapping region is also effectively modeled with our extrapolation scheme (e). We recommend the readers to zoom-in to see the difference clearly.

where $\mathbf{w}^T$ is a row vector of m elements constructed by stacking the weights for the m control points, $\mathrm{vec}(\cdot)$ denotes a vectorization operation that converts a 3×3 matrix to a row vector of size 1×9, and $\mathrm{ivec}(\cdot)$ denotes an *inverse-vectorization* operator that converts a 1×9 row vector to a 3×3 matrix. Instead of the summations in Eq. 7, we can now compute transformation matrix $T_{\mathbf{x}}$ more efficiently by matrix multiplications, which can be easily parallelized. We implement the parallel processing using a GPGPU.

3.2. Probabilistic Modeling of the Color Transfer

The MLS framework is built upon the control points, which are basically corresponding colors in a given image pair in our color transfer problem. Finding correspondences between two images is not an easy task and matching errors do exist in many registration algorithms. Since the outliers in the matching can severely affect our MLS framework, we propose to use a probabilistic modeling of color transfer in order to gain robustness against the outliers and noise by taking into account the reliability of the computed correspondences.

We determine the reliability of each corresponding color pair by considering both the probability of the forward mapping (input image I to reference image J) and the reverse mapping (reference image J to input image I). Let $I(r,g,b)$ and $J(r,g,b)$ indicate the RGB values in the input image I and the reference image J respectively. To simplify the formulation, we define an indexing function in the input image from r, g, b values to a single index i as $\mathcal{I}\{(r,g,b) \leftrightarrow i\}$ (j is the index of the reference image). From a set of corresponding color values computed through an image registration algorithm, we can first compute the probability of each mapping as follows:

$$p(I(i), J(j)) = \frac{\#\ matches(i,j)}{\#\ total\ matches}. \tag{9}$$

Then, we define the probability of the mappings as:

$$p(\mathcal{M}\{I(i), J(j)\}) \triangleq p(I(i)|J(j))p(J(j)|I(i)), \tag{10}$$

where $\mathcal{M}\{\cdot\}$ denotes a mapping function between i and j. $p(I(i)|J(j))$ and $p(J(j)|I(i))$ can be simply computed using the Bayes' theorem and the marginalization as follows:

$$\begin{aligned} p(I(i)|J(j)) &= \frac{p(I(i), J(j))}{p(J(j))} \\ &= \frac{p(I(i), J(j))}{\sum_{k=1}^{n} p(I(k), J(j))}, \end{aligned} \tag{11}$$

$$p(J(j)|I(i)) = \frac{p(I(i), J(j))}{\sum_{k=1}^{n} p(I(i), J(k))}. \tag{12}$$

Note that considering both directions in Eq. 10 has the effect of reducing the outliers.

The final color mapping probability between the two images can now be computed as

$$\begin{aligned} &p(\mathcal{M}\{I(i), J(j)\}) \\ &= \frac{p(I(i), J(j))^2}{\sum_{k=1}^{n} p(I(i), J(k)) \sum_{k=1}^{n} p(I(k), J(j))}. \end{aligned} \tag{13}$$

Since the space of possible color is much bigger (256^3) than the color distribution of an image, we divide the color space by fixed-size(n) bins for computing $p(\mathcal{M}\{I(i), J(j)\})$. The mapping function $\mathcal{M}\{\cdot\}$ is essentially a $n \times n$ matrix with ($I(i)$,$J(j)$) representing the bin index. We set the bin size as $20 \times 20 \times 20$ in all of our experiments, in which case the number of bins $n = 13 \times 13 \times 13 = 2197$.

3.3. Probabilistic Moving Least Squares

Probabilistic modeling of the color transfer in the previous subsection provides the reliability measure of the control points. We now combine this reliability measure to the MLS framework by considering the probability of the color transfer of the control points in addition to the distance of the color to be evaluated to the control points. The weight in Eq. 2 becomes :

$$w_k = \frac{1}{\left|\mathbf{u}_k - \mathbf{x}\right|^{2\alpha} + \epsilon} \times p(\mathcal{M}\{I(i), J(j)\}) \quad \text{s.t.} \quad \mathbf{u}_k \in I(i), \mathbf{v}_k \in J(j). \tag{14}$$

	PSNR (dB)								SSIM							
	Base	Rein[16]	3×3	Poly [7]	BTF[11]	Tai[19]	CIM[14]	Ours	Base	Rein[16]	3×3	Poly[7]	BTF[11]	Tai[19]	CIM[14]	Ours
building	12.366	19.565	17.614	20.994	18.468	20.173	20.898	21.502	0.684	0.816	0.800	0.858	0.844	0.822	0.861	0.850
flower1	15.405	19.949	22.936	25.507	21.747	19.719	22.846	26.219	0.942	0.958	0.965	0.970	0.952	0.901	0.966	0.973
flower2	18.823	22.554	23.431	25.372	24.453	22.410	24.604	25.988	0.890	0.903	0.907	0.929	0.927	0.864	0.927	0.937
gangnam1	21.575	21.973	28.511	30.853	22.625	24.009	29.164	34.724	0.944	0.950	0.969	0.980	0.944	0.887	0.974	0.982
gangnam2	19.654	24.256	27.558	30.711	24.807	24.169	27.958	37.793	0.957	0.949	0.957	0.969	0.965	0.934	0.967	0.989
gangnam3	18.250	23.124	25.237	26.742	20.824	21.874	25.379	35.350	0.960	0.918	0.971	0.972	0.952	0.849	0.973	0.991
illum	14.581	18.595	18.181	19.386	18.370	18.715	19.180	19.567	0.532	0.594	0.580	0.612	0.601	0.591	0.613	0.611
mart	18.460	21.204	22.989	23.639	22.675	21.986	22.405	23.701	0.892	0.881	0.895	0.896	0.893	0.882	0.887	0.897
playground	20.601	23.770	24.509	25.398	23.225	25.204	25.858	26.742	0.885	0.905	0.906	0.910	0.880	0.904	0.914	0.918
sculpture	16.152	26.170	27.783	30.011	29.538	25.956	29.011	30.244	0.953	0.969	0.972	0.974	0.973	0.926	0.972	0.973
tonal1	17.642	25.382	25.589	32.498	33.203	24.772	32.842	35.245	0.902	0.944	0.956	0.980	0.989	0.917	0.988	0.990
tonal2	18.043	25.566	27.327	28.375	23.393	23.051	27.001	30.592	0.966	0.985	0.983	0.987	0.983	0.978	0.986	0.994
tonal3	15.963	25.327	26.284	28.729	28.192	24.437	28.274	33.866	0.890	0.964	0.956	0.978	0.983	0.940	0.985	0.996
tonal4	14.274	24.402	24.370	29.430	28.350	23.806	29.878	33.726	0.853	0.947	0.955	0.982	0.982	0.873	0.984	0.991
tonal5	18.429	25.227	24.780	29.813	29.249	21.673	29.824	34.169	0.933	0.955	0.967	0.975	0.988	0.925	0.985	0.991

Table 1: Quantitative evaluations. Red, blue, and green indicate 1^{st}, 2^{nd}, and 3^{rd} best performance respectively. The proposed PMLS method outperforms other methods in terms of PSNR and SSIM when applied to various test image sets such as the tonal adjustment database [4], captured image pairs with different camera settings, cameras, illumination and different photo retouch styles.

Again, $\mathbf{u}_k$ and $\mathbf{v}_k$ denote the color of the k^{th} corresponding points in image I and J respectively. The term ϵ is added because there may exist other points with the same color as the control point $\mathbf{u}_k$ in our color transfer problem. We set ϵ to 1 in all of our experiments. By combining the probability term to the weight, our probabilistic moving least squares (PMLS) can now deal with the registration errors and noise more effectively. The parameter α in Eq. 14 controls the weighting on the control points in the scattered point interpolation. A large α value places a hard constraint on the given color matches while a smaller α would allow for a smoother interpolation with a soft constraint. In our experiments, we set α to 2.

For our PMLS framework, we need a sufficient amount of color matches between an image pair in order to cover a large range of color as well as to compute the probability of the color transfer ($p(\mathcal{M}\{I(i), J(j)\})$. We register two images by a planar homography [3] although other image matching schemes can also be used such as dense stereo matching [20], SIFT flow [13], etc. After the image registration, we randomly select a small portion of the matching points (1% in this work) as the control points of our PMLS algorithm. Fig. 3 shows the performance improvement by using the PMLS(Fig. 3(d)) over the conventional MLS scheme(Fig. 3(c)).

3.4. Extrapolation Scheme

Since the PMLS is essentially an interpolation method, problems may arise when the overlapping area between the input and the reference is small as shown in Fig. 3. For a color point outside the overlapping region, there may not be control points nearby to effectively transfer the color. To deal with this extrapolation issue, we additionally add control points in the color range that is not covered in the overlapping region. If a bin in the color mapping bins computed in Sec. 3.2 does not include any points, we add the center point of this bin as a control point. Destination of this color point is determined by a parametric color transfer model (2nd order polynomial [7]) computed with the control points in the overlapping region. The effectiveness of our extrapolation scheme is shown in Fig. 3 (e), which shows the improvement in the color consistency in the non-overlapping regions (e.g. ceiling).

4. Experiments

In this section, we provide a variety of experiments to validate our PMLS algorithm for color transfer. The computational time of our method is proportional to image size and the number of control points. With our CUDA implementation using nVIDIA Quadro 4000, it takes 4.5 seconds to color transfer an 1M pixel image.

We first provide quantitative evaluations of different color transfer algorithms in Table 1. The evaluation datasets[2] include many sources of the color change including different cameras, different camera settings, different illuminations, and different photo retouch styles. Given a registered image pair, the color of the first image is transferred to match the color of the second image, and then PSNR and SSIM (Structural SIMilarity) [21] are computed over the corresponding pixels as the measures of the performance [3]. As for the SSIM which is a measure of structural

[2]The datasets used for the evaluation is shown in the supplementary material.

[3]Note that although the registration errors may affect the evaluation measures, the measures for different methods were computed using the same registration information.

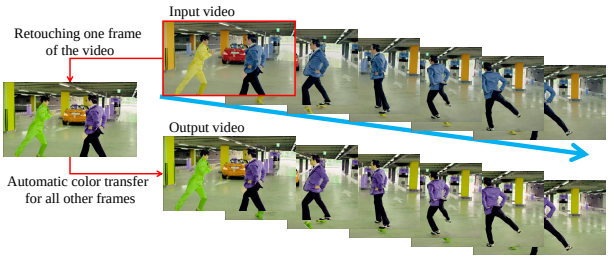


Figure 6: Workflow of the video color transfer. The user retouches the color of just one frame and the color transfer is computed between the original image and the edited image. The same color transfer is applied to all other frames for the video color transfer.

similarity, our PMLS scheme outperforms other method in most cases although the difference is relatively small. The PSNR measures the difference in color after the transfer and maximizing this measure is the ultimate goal of this paper. In terms of PSNR, our framework outperforms all other methods for all datasets especially for different camera cases and photo retouch examples since they are highly nonlinear which is difficult to model parametrically.

In Fig. 4, we show more qualitative and quantitative evaluations. For the qualitative evaluation, the results are shown by creating an image mosaic by switching between the reference image and the transformed image column-wise. When the color transfer is accurate, the resulting mosaic should look like a single image as is the case with our method. Errors maps are also provided for quantitative evaluations. Again, it can be seen that our PMLS framework outperforms other state-of-the-art methods both qualitatively and quantitatively.

As for the application of our color transfer framework, we first apply it to align color of different images to create a color consistent image panorama as shown in Fig. 5. With our color transfer, we can create a color consistent panorama without any blending as can be seen in the figure.

We further apply our method for video color transfer as shown in Fig. 6. In the video color transfer application, a user only has to retouch or edit one frame of the video to change the color scheme of the whole video clip instead of retouching all other frames. The user picks a frame in the video clip and edits the color of the frame. The color transfer is computed between the original frame and the edited version of the frame. Then the computed color transfer is applied to all other frames in the sequence, resulting in a consistent color transferred video as shown in Fig. 7.

5. Discussion

We have presented a new mechanism for transferring color between images using a probabilistic moving least squares framework. Through numerous experiments, we have shown that our method can transfer color between images more accurately than the previous color transfer methods and be used for interesting applications such as video color transfers.

Currently, our color transfer framework is applied globally in the spatial domain, acting as a one-to-one color mapping function. This color model explains many instances of color variation such as different camera, different camera setting, and global tonal retouch very well as seen in the paper. Our model can also explain illumination variations such as global lighting color variations (as in the computational color constancy) and even directional illumination change as long as the mapping is one-to-one. However, our method will have problems for one-to-many color mappings, e.g. local illumination variations due to shadows, specularities, and etc. In the future, we would like to take this locality into consideration by looking for piecewise-consistent color mappings between images as in [9]. Furthermore, we would like to explore extending our framework for N-view case where there are more than 2 overlapping images [22] by employing a joint optimization scheme.

Acknowledgements This research was partially supported by the Center for Integrated Smart Sensors as Global Frontier Project (CISS-2011-0031868), the IT R&D program of MSIP/KEIT (No. 10043450), the National Research Foundation grant funded by the Korea government (MSIP) (No. 2010-0028680), and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2013R1A1A1005065).

References

[1] N. K. Bose and N. A. Ahuja. Superresolution and noise filtering using moving least squares. *IEEE Transactions on Image Processing*, 15(8):2239–2248, 2006. 2

[2] D. H. Brainard and W. T. Freeman. Bayesian color constancy. *Journal of the Optical Society of America A*, 14:1393–1411, 1997. 1, 2

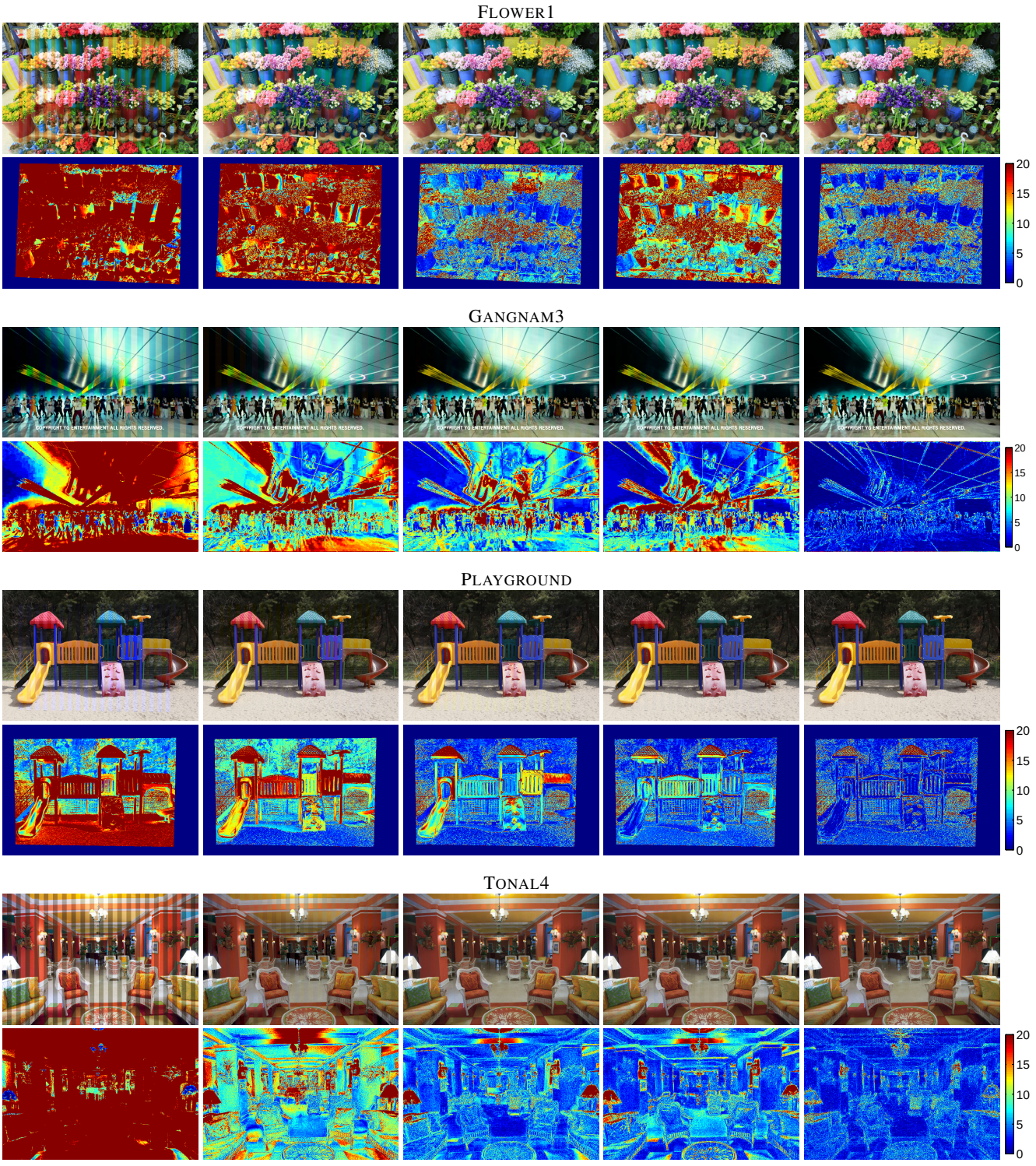
[3] M. Brown and D. G. Lowe. Automatic Panoramic Image Stitching using Invariant Features. *International Journal of Computer Vision*, 74(1):59–73, Aug. 2007. 5, 8

[4] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *CVPR*, pages 97–104, 2011. 1, 5

[5] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. In *ACM SIGGRAPH 2005*, pages 544–552, 2005. 2

[6] T.-W. Huang and H.-T. Chen. Landmark-based sparse color representations for color transfer. In *ICCV*, pages 199–204, 2009. 1

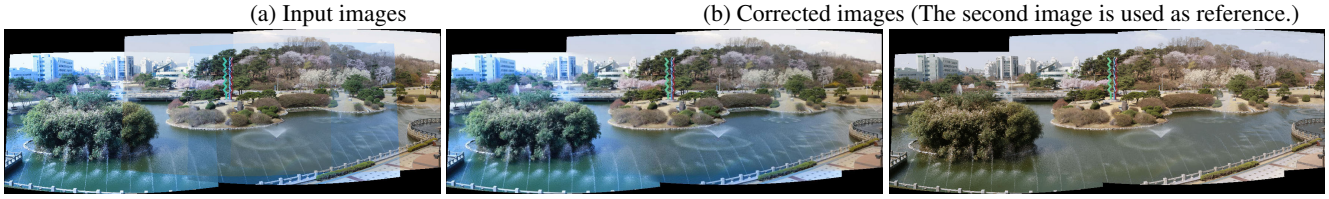
[7] A. Ilie and G. Welch. Ensuring color consistency across multiple cameras. In *ICCV*, pages 1268–1275, 2005. 2, 5, 7



(a) Baseline (b) Reinhard [16] (c) Second Polynomial [7] (d) CIM [14] (e) Ours

Figure 4: More qualitative and quantitative results. The reference image and the color transferred image are overlayed by column blocks. Using our algorithm, the mosaic images look as if they are just a single image, while the results from the other methods show clear discrepancy. We recommend the readers to zoom-in to see the difference clearly.

(a) Input images (b) Corrected images (The second image is used as reference.)



(c) Panorama using the input images (d) Autostitch with multi-band blending [3] (e) Panorama using the corrected images

Figure 5: Photometric alignment using our color transfer method for generating a color consistent panorama. Geometric alignment is obtained from [3].

GANGNAM1

(a) Example pair (b) Frame 2685 (c) Frame 2705 (d) Frame 2725 (e) Frame 2745

Figure 7: Video transfer application. (Top) Original Video (Bottom) Color transfered video where the color transfer is computed from just one example pair.

[8] J. Jia and C.-K. Tang. Tensor voting for image correction by global and local intensity alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):36–50, 2005. 2

[9] S. Kagarlitsky, Y. Moses, and Y. Hel-Or. Piecewise-consistent color mappings of images acquired under various conditions. In *ICCV*, pages 2311–2318, 2009. 6

[10] S. J. Kim, H. T. Lin, Z. Lu, S. Süsstrunk, S. Lin, and M. S. Brown. A new in-camera imaging model for color computer vision and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2289–2302, 2012. 1, 2

[11] S. J. Kim and M. Pollefeys. Robust radiometric calibration and vignetting correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:562–576, 2008. 1, 2, 5

[12] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 87:141–158, 1981. 2

[13] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*, pages 28–42, 2008. 5

[14] M. Oliveira, A. D. Sappa, and V. Santos. Unsupervised local color correction for coarsely registered images. In *CVPR*, pages 201–208, 2011. 1, 2, 5, 7

[15] F. Pitie, A. Kokaram, and R. Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *ICCV*, pages 1434–1439, 2005. 2

[16] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001. 1, 2, 5, 7

[17] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. In *ACM SIGGRAPH 2006*, pages 533–540, 2006. 2, 3

[18] Y.-W. Tai, J. Jia, and C.-K. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *CVPR*, pages 747–754, 2005. 1, 2

[19] Y.-W. Tai, J. Jia, and C.-K. Tang. Soft color segmentation and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1520–1537, 2007. 5

[20] E. Tola, V. Lepetit, and P. Fua. Daisy: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, 2010. 5

[21] 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, 2004. 5

[22] Y. Xiong and K. Pulli. Color matching of image sequences with combined gamma and linear corrections. In *ACM Multimedia*, 2010. 6

[23] W. Xu and J. Mulligan. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In *CVPR*, pages 263–270, 2010. 1, 2