

# Probabilistic Moving Least Squares with Spatial Constraints for Nonlinear Color Transfer Between Images

Youngbae Hwang<sup>a</sup>, Joon-Young Lee<sup>b</sup>, In-So Kweon<sup>c</sup>, Seon Joo Kim<sup>d</sup>

<sup>a</sup>*Korea Electronics Technology Institute, Seongnam-si, Gyeonggi-do, Korea*

<sup>b</sup>*Adobe*

<sup>c</sup>*KAIST, Daejeon, Korea*

<sup>d</sup>*Yonsei University, Seoul, Korea*

---

## Abstract

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 align the color of a scene between images, we introduce a novel color transfer framework based on a scattered point interpolation scheme. Compared to the conventional color transformation methods that use a parametric mapping or color distribution matching, we solve for a full nonlinear and nonparametric color mapping in the 3D RGB color space by employing the moving least squares framework. We further strengthen the transfer with a probabilistic modeling of the color transfer in the 3D color space as well as spatial constraints to deal with mis-alignments, noise, and spatially varying illumination. 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.

**Keywords:** Color Transfer, Color Correction, Moving Least Squares

---

## 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 (Kim et al., 2012)) (Fig. 1). Photographs of a scene may also vary due to different photographic adjustment styles of the users (Bychkovsky et al., 2011).

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.<sup>1</sup> 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 (Brainard and Freeman, 1997; Gijssen et al., 2011). It is also used to generate color consistent image panoramas and 3D texture-maps (Kim and Pollefeys, 2008; Xiong and Pulli, 2010a; Xu and Mulligan, 2010), as well as to enhance and manipulate images by emulating the tone and the color style of other images (Huang and Chen, 2009; HaCohen et al., 2013; Tsai et al., 2016).

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 parametric 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 (Kim et al., 2012). 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. We further include spatial constraints to the probabilistic moving least squares framework to deal with local variations of a color due to local illumination changes, viewpoint changes, etc. 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. Note that this work focuses on transferring color between images of a same scene, different from works that transfer color between different scenes Reinhard et al. (2001).

A preliminary version of this work was presented in (Hwang et al., 2014). On top of adding significantly more experiments, we improved the previous algorithm in (Hwang et al., 2014) by introducing a new weight to account for spatially

---

<sup>1</sup>The process is also called as color correction, color mapping, or photometric alignment.

varying color transfer (3.3). By considering the distance from the location of control points, color transfer for the target pixel is dominated by closer and more similar control points. While the previous method could only work for global color transfers (one-to-one color mappings), the new framework can be applied for local color transfers (one-to-many color mappings) due to spatially varying illumination, non-Lambertian scenes, etc. We have also added more implementation details and included results using different registration algorithm, whereas the previous work only showed results with registration by homographies.

## 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  $3 \times 3$  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 (Brainard and Freeman, 1997). 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 (Ilie and Welch, 2005):  $\mathbf{x}' = \mathbf{M}' \cdot [r^n, g^n, b^n, \dots, r, g, b, 1]^\top$ . By introducing higher degree terms, the polynomial model can achieve more expressive mapping than the linear model. To handle the nonlinearity of the color mapping more effectively, HaCohen et al. (2011) presented a piecewise cubic spline with 7 breaks for each color channel. Faridul et al. (2013) achieved the cross-channel color mapping by performing a nonlinear channel-wise mapping and a linear cross-channel mapping consecutively. Sajadi et al. (2010) presented a color transfer function using multiple higher-dimensional Bézier patches.

Another popular color transfer method is based on the statistics of the color distribution in images, first proposed by Reinhard et al. (2001). 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 (Tai et al., 2007; Pitie et al., 2005; Pitie and Kokaram, 2007; Oliveira et al., 2011). Tai et al. (2007) presented a soft-segmentation technique and transferred the statistics of color distributions between each segment



Figure 1: Changes in image color due to different factors (1st to 4th rows show image changes due to different camera types (iPhone vs. Cannon), different camera settings, illumination changes, and photographic retouches). 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.

pair. Bonneel et al. (2013) combined three different affine chrominance transforms in different luminance bands with a non-linear tone mapping curve. Pitie and Kokaram (2007) modeled the statistics of color distributions using a multi-variate Gaussian distribution and adopted the Monge-Kantorovitch solution that minimizes the amount of color changes. Lee et al. (2016) presented an improved technique which avoids the visual artifacts of the Monge-Kantorovitch solution with an additional regularization and achieved more expressive results with a non-linear luminance transfer. Oliveira et al. (2011) first computed color distribution mapping functions from the output of the mean shift segmentation, then presented the final color transfer result by merging the local mapping functions according to the color influence maps. Oliveira et al. (2015) further extended the method by modelling the local color distributions with a collection of truncated Gaussians and computing the local color palette mapping functions from the Gaussians. Pitie et al. (2005) presented a nonlinear mapping algorithm for transferring multi-dimensional color histograms instead of transferring Gaussian statistics linearly. Bonneel et al. (2016) proposed to compute the optimal transport barycenter of histograms and presented a color transfer without large color distortions. Freedman and Kisilev (2010) presented a Stretch-Minimizing Structure Preserving (SMSP) transformation applied to a discrete color flow between target and source histograms. While these statistical approaches are 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 (Kim and Pollefeys, 2008), a 2D tensor voting scheme (Jia and Tang, 2005), and a color homography (Gong et al., 2016). Also, there are several studies to enhance the visual quality of color transfer with special assumptions in content (Wu et al., 2013), illuminant (Nguyen et al., 2014; Frigo et al., 2014), and style (Hristova et al., 2015). For more detailed comparisons and evaluations of different color transfer methods, we refer the readers to (Xu and Mulligan, 2010; Faridul et al., 2014).

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 (Kim et al., 2012), 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 (Lancaster and Salkauskas, 1981) 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 (Schaefer et al., 2006), surface reconstruction (Fleishman et al., 2005), and image super-resolution and denoising (Bose and Ahuja, 2006). Recently, an image stitching method based on the Moving DLT (Direct Linear Transformation), which is a variant of the MLS, was proposed for seamlessly bridging image regions that are inconsistent with the projective model (Zaragoza et al., 2014).

In this paper, we apply the MLS method to the color transfer problem. We chose the MLS as it is easier to incorporate other weighting terms into the framework compared to the other interpolation schemes such as the radial basis function as used in (Kim et al., 2012). 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. We additionally add a weighting term for spatial constraints in order to deal with locally varying color transformation. To the best of our knowledge, this is the first attempt to employ the MLS framework that operates in the color domain.

## 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  $3 \times 3$  mapping and the distribution matching (Reinhard et al., 2001).

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

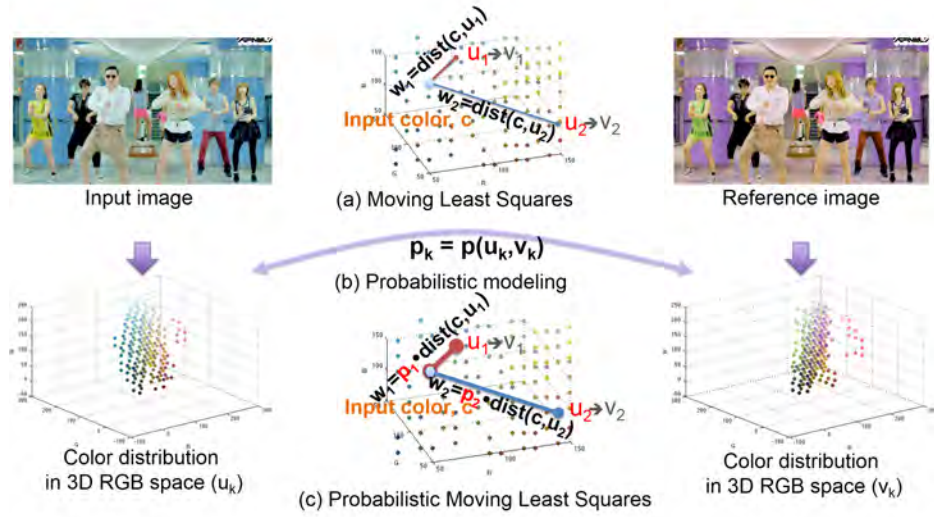


Figure 2: The PMLS concept. (a) The original MLS framework, the thickness of the line indicates the distance weight. (b) The probability of the bidirectional color transfer between the corresponding control points (Eq. 13). (c) The proposed PMLS framework. The thickness of the line indicates the distance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

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

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

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

$$w_k = \frac{1}{|\mathbf{u}_k - \mathbf{x}|^{2\alpha}}. \quad (2)$$

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 (Schaefer et al., 2006). Therefore, the transformation  $T_{\mathbf{x}}$  also varies for each  $\mathbf{x}$ .

For image deformation work (Schaefer et al., 2006), a rigid transformation  $T_{\mathbf{x}}$  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  $T_{\mathbf{x}}$ :

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

where  $\mathbf{A}_{\mathbf{x}}$  is a full  $3 \times 3$  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

$$\mathbf{b}_{\mathbf{x}} = \bar{\mathbf{v}} - \mathbf{A}_{\mathbf{x}}\bar{\mathbf{u}}, \quad (4)$$

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

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

Eq. 3 can then be rewritten as

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

and Eq. 1 becomes

$$\sum_k w_k |\mathbf{A}_{\mathbf{x}}\hat{\mathbf{u}}_k - \hat{\mathbf{v}}_k|^2, \quad (6)$$

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

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

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



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

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:

$$\begin{aligned} \mathbf{A}_x &= \text{ivec}(\mathbf{w}^\top \mathbf{y}_1)^{-1} \times \text{ivec}(\mathbf{w}^\top \mathbf{y}_2) \\ \text{s.t. } \mathbf{y}_1 &= [\text{vec}(\hat{\mathbf{u}}_1 \hat{\mathbf{u}}_1^\top)^\top, \dots, \text{vec}(\hat{\mathbf{u}}_k \hat{\mathbf{u}}_k^\top)^\top]^\top, \\ \mathbf{y}_2 &= [\text{vec}(\hat{\mathbf{u}}_1 \hat{\mathbf{v}}_1^\top)^\top, \dots, \text{vec}(\hat{\mathbf{u}}_k \hat{\mathbf{v}}_k^\top)^\top]^\top, \end{aligned} \quad (8)$$

where  $\mathbf{w}^\top$  is a row vector of  $m$  elements constructed by stacking the weights for the  $m$  control points,  $\text{vec}(\cdot)$  denotes a vectorization operation that converts a  $3 \times 3$  matrix to a row vector of size  $1 \times 9$ , and  $\text{ivec}(\cdot)$  denotes an *inverse-vectorization* operator that converts a  $1 \times 9$  row vector to a  $3 \times 3$  matrix. Instead of the summations in Eq. 7, we can now compute transformation matrix  $T_x$  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:

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

Then, we define the probability of the mappings as:

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

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:

$$p(I(i)|J(j)) = \frac{p(I(i), J(j))}{p(J(j))} = \frac{p(I(i), J(j))}{\sum_{k=1}^n p(I(k), J(j))}, \quad (11)$$

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

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

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

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  $n = 13 \times 13 \times 13 = 2,197$ . When we compute the probability of each mapping as in Eq. 9, we set the bin of low probability to zero for reducing the effect of outliers.

### 3.3. Bilateral weights for 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



Figure 3: Dealing with spatially varying color transfer. We synthesize spatially varying color transfer by a weighted sum between two color transformed images according to pixel height  $j$ ,  $T(I(j)) = \frac{j}{h} \times T_1(I(j)) + \frac{(h-j)}{h} \times T_2(I(j))$ ,  $j = 1 \dots h$ . Thus, color transformation for each row is changed gradually as pixel height is changed. The proposed B-PMLS method can deal with the local change effectively.

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 :

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

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  $\epsilon$  is added because there may exist other points with the same color as the control point  $\mathbf{u}_k$  in our color transfer method. By combining the probability term to the weight, the probabilistic moving least squares (PMLS) can now deal with the registration errors and noise more effectively.

So far, the color transfer model can only be applied for a global color transfer as the model produces the same RGB for the pixels with the same RGB inputs. However, there may be cases where the color variation is local due to local illumination change or shading, non-Lambertian scenes, etc. To take the local color transfer into account, we propose Bilateral weights for the Probabilistic Moving Least Squares (B-PMLS) as:

$$w_k = \exp\left(-\frac{1}{2}\left(\frac{\|\mathbf{P}_{\mathbf{u}_k} - \mathbf{P}_{\mathbf{x}}\|^2}{\sigma_s^2} + \frac{\|\mathbf{u}_k - \mathbf{x}\|^2}{\sigma_r^2}\right)\right) \times p(\mathcal{M}\{I(i), J(j)\}), \quad (15)$$

where  $\mathbf{P}_{\mathbf{u}_k}$  and  $\mathbf{P}_{\mathbf{x}}$  are the locations of the control point  $\mathbf{u}_k$  and an input pixel  $\mathbf{x}$ , respectively. The intuition behind the new weight is that the color of a pixel is transferred by its distance to the control points in the color space as well as in the spatial domain.  $\sigma_s^2$  and  $\sigma_r^2$  are used to control the balance between the two. Fig. 3 shows the advantage of the B-PMLS over the PMLS and the MLS in case of the local color variations.

For our B-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 ( $p(\mathcal{M}\{I(i), J(j)\})$ ). In this work, we register two images by a planar homography (Brown and Lowe, 2007) or a non-rigid dense correspondence (NRDC) scheme (HaCohen et al., 2011) depending on the given images. Note that other image matching schemes can also be used such as dense stereo matching (Tola et al., 2010) and SIFT flow (Liu et al., 2008). After the image registration, we randomly select a small portion of the matching points (1% in this work) as the control points of our algorithm. Fig. 4 shows the performance improvement by using the B-PMLS(Fig. 4(e)) over the MLS scheme(Fig. 4(c)) and PMLS(Fig. 4(d)).

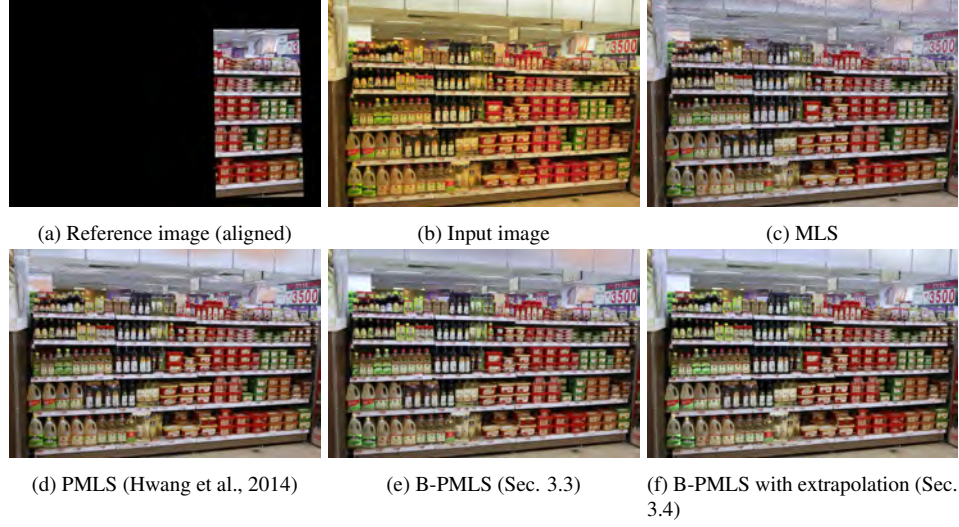
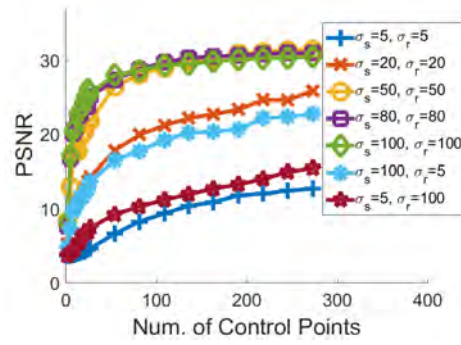


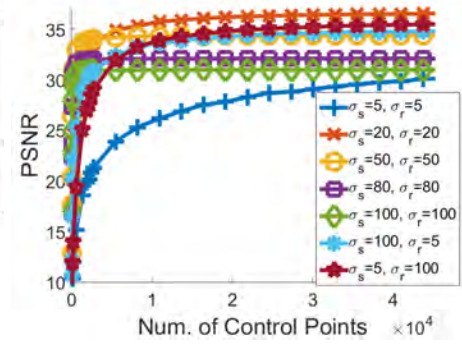
Figure 4: Comparing the performance of MLS, PMLS, and B-PMLS. Matching errors adversely affect the performance of the MLS algorithm, while the PMLS can absorb the matching errors due to the probabilistic modeling (d). B-PMLS shows better transferred results than PMLS especially in non-fronto-parallel regions (e.g., ceiling) by using the spatial constraints (e). Color transfer in the non-overlapping region is also effectively modeled with our extrapolation scheme (f). We recommend the readers to zoom-in to see the difference clearly.



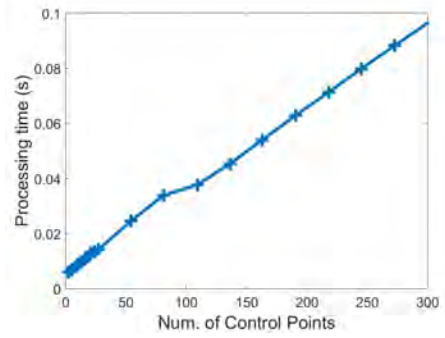
Figure 5: Color correction results with various numbers of control points. Corresponding PSNRs from (a) to (d) are 17.95, 26.69, 29.04 and 31.46dB, respectively.



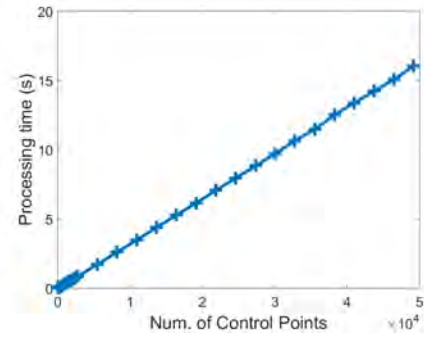
(a) PSNR with varying number of control points



(b) PSNR with varying number of control points (larger values)



(c) Processing time with varying number of control points



(d) Processing time with varying number of control points (larger values)

Figure 6: PSNR with varying number of control points



### 3.4. Extrapolation Scheme

Since the B-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. 4. 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 computed with the control points in the overlapping region. We can use any parametric methods for extrapolation such as RGB to RGB transform (Ilie and Welch, 2005), general polynomial transform (Ilie and Welch, 2005), and statistical mapping (Reinhard et al., 2001; Pitie and Kokaram, 2007). We used the second order polynomial transformation (Ilie and Welch, 2005) for generating the control points for the extrapolation in this paper. The effectiveness of our extrapolation scheme is shown in Fig. 4(f), which shows the improvement in the color consistency in the non-overlapping regions (e.g. ceiling).

### 3.5. Implementation issues

Our B-PMLS framework can be efficiently implemented using the GPGPU based parallel processing because the process can be run independently for each pixel. Each thread in GPU have a run for each pixel with the same control points and the probabilities. Note that because each pixel has different weights as in Eq. 15 according to its RGB value, the internal memory in GPU should be allocated separately. To avoid the limitation of the number of control points, we compute the weights for the control points one more time instead of storing them in the internal memory in GPU.

Important parameters for the proposed method are  $\sigma_s$  and  $\sigma_r$  in Eq. 15, and the number of control points  $m$ . Fig. 5 shows an example of the effect of the number of control points  $m$ . On an example that has diverse colors, we can transfer most colors by only using 13 control points as shown in Fig. 5(b). With 27 control points, we get a very good transfer result that is close to the reference image with the PSNR value being 29.04dB. As expected, the visual quality and the PSNR improve with the growth in  $m$ , although increasing  $m$  would also increase the computational time.

Fig. 6 shows the effect of varying  $\sigma_s$ ,  $\sigma_r$  and  $m$  simultaneously. Because small  $\sigma_s$  and  $\sigma_r$  values allow highly similar control points in the narrow neighborhood, color transfer results are worse than other parameter settings. In contrast, since

Scene	PSNR									
	Baseline	Reinhard	3×3	2nd Poly	IDT	BTF	Tai	CIM	PMLS	B-PMLS
building	12.366	19.565	17.614	20.994	20.201	18.468	20.173	20.898	21.608	23.568
flower1	15.405	19.949	22.936	25.507	25.334	21.747	19.719	22.846	26.023	26.316
flower2	18.823	22.554	23.431	25.372	25.456	24.453	22.410	24.604	25.565	25.808
gangnam1	21.575	21.973	28.511	30.853	25.196	22.625	24.009	29.164	35.167	35.654
gangnam2	19.654	24.256	27.558	30.711	28.775	24.807	24.169	27.958	36.064	35.864
gangnam3	18.250	23.124	25.237	26.742	22.301	20.824	21.874	25.379	34.357	35.429
illum	14.581	18.595	18.181	19.386	18.874	18.370	18.715	19.180	19.578	21.430
mart	18.460	21.204	22.989	23.639	23.425	22.675	21.986	22.405	23.917	24.140
playground	20.601	23.770	24.509	25.398	25.849	23.225	25.204	25.858	26.473	27.010
sculpture	16.152	26.170	27.783	30.011	30.106	29.538	25.956	29.011	30.241	30.866
tonal1	17.642	25.382	25.589	32.498	35.086	33.203	24.772	32.842	35.302	35.917
tonal2	18.043	25.566	27.327	28.375	28.929	23.393	23.051	27.001	30.141	30.775
tonal3	15.963	25.327	26.284	28.729	31.174	28.192	24.437	28.274	33.483	34.215
tonal4	14.274	24.402	24.370	29.430	31.900	28.350	23.806	29.878	33.302	34.547
tonal5	18.429	25.227	24.780	29.813	32.593	29.249	21.673	29.824	34.068	35.604

Scene	SSIM									
	Baseline	Reinhard	3×3	2nd Poly	IDT	BTF	Tai	CIM	PMLS	B-PMLS
building	0.684	0.816	0.800	0.858	0.849	0.844	0.822	0.861	0.854	0.890
flower1	0.942	0.958	0.965	0.970	0.968	0.952	0.901	0.966	0.970	0.971
flower2	0.890	0.903	0.907	0.929	0.931	0.927	0.864	0.927	0.933	0.935
gangnam1	0.944	0.950	0.969	0.980	0.933	0.944	0.887	0.974	0.990	0.991
gangnam2	0.957	0.949	0.957	0.969	0.970	0.965	0.934	0.967	0.992	0.986
gangnam3	0.960	0.918	0.971	0.972	0.968	0.952	0.849	0.973	0.992	0.993
illum	0.532	0.594	0.580	0.612	0.608	0.601	0.591	0.613	0.601	0.687
mart	0.892	0.881	0.895	0.896	0.895	0.893	0.882	0.887	0.895	0.901
playground	0.885	0.905	0.906	0.910	0.911	0.880	0.904	0.914	0.920	0.923
sculpture	0.953	0.969	0.972	0.974	0.973	0.973	0.926	0.972	0.972	0.974
tonal1	0.902	0.944	0.956	0.980	0.991	0.989	0.917	0.988	0.990	0.991
tonal2	0.966	0.985	0.983	0.987	0.988	0.983	0.978	0.986	0.993	0.995
tonal3	0.890	0.964	0.956	0.978	0.991	0.983	0.940	0.985	0.996	0.997
tonal4	0.853	0.947	0.955	0.982	0.988	0.982	0.873	0.984	0.990	0.994
tonal5	0.933	0.955	0.967	0.975	0.989	0.988	0.925	0.985	0.991	0.996

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



higher  $\sigma_s$  and  $\sigma_r$  values reduce distinctiveness of control points, they also cannot obtain good results. The proposed color transfer method shows parameter insensitivity for a sufficient number of control points (more than 1%), when we select intermediate values ( $20 \leq \sigma_s, \sigma_r \leq 80$ ). In this paper, we set  $\sigma_s = 20, \sigma_r = 20$  in all of our experiments.

The processing time with vary  $m$  is shown in Fig. 6(c)(d). We use nVidia GTX Titan with 6GB GDDR5. Because the increase of  $m$  yields the increase of weight computation in Eq. 15, the processing is proportional to  $m$ . If we use the 1% of 1M pixel image (10,000 control points), the processing time is approximately 5 seconds. However, using only 100 control points would still achieve good results ( $> 30dB$ ) in real time as shown in Fig. 6(a)(c). Note that in terms of the PSNR value, the next best method to our work would result in the PSNR of 29.824dB (See *tonal5* in Table 1).

#### 4. Experiments

In this section, we provide a variety of experiments to validate our B-PMLS algorithm for color transfer. We first provide quantitative evaluations of different color transfer algorithms (Reinhard (Reinhard et al., 2001),  $3 \times 3$ , 2nd Poly (Ilie and Welch, 2005), IDT (Pitie et al., 2005), BTF (Kim and Pollefeys, 2008), Tai (Tai et al., 2005), CIM (Oliveira et al., 2011), PMLS (Hwang et al., 2014)) in Table 1. Note that Reinhard, IDT, and Tai do not require explicit color matches between two images. The evaluation datasets 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) (Wang et al., 2004) are computed over the corresponding pixels as the measures of the performance<sup>2</sup>. As for the SSIM which is a measure of structural similarity, our B-PMLS scheme outperforms other method in most cases because of the partial filtering with the Bilateral weights. The PSNR measures the difference in color after the transfer and maximizing this measure is the ultimate goal of this paper. In terms of the 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.

---

<sup>2</sup>Note that although the registration errors may affect the evaluation measures, the measures for different methods were computed using the same registration information.

In Fig. 7, 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 columnwise. 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 B-PMLS framework outperforms other state-of-the-art methods both qualitatively and quantitatively. Both PMLS and B-PMLS show very good performance on GANGNAM3 and TONAL4 because the scenes have global color changes. However, when a scene contains large local color variations due to local illumination changes as in the case of BUILDING and ILLUM, B-PMLS shows much better results than PMLS because B-PMLS can effectively handle local color transfer.

For more detailed comparisons between PMLS and B-PMLS, we show the enlarged result of the BUILDING in Fig. 8. We also compare the zoomed-in views of the same patch locations. Both results look good as a whole, but PMLS suffers from artifacts in local regions since it cannot handle local color changes. On the other hand, B-PMLS achieves accurate color transfer results even with local illumination changes.

We further show experimental results in Fig. 9, this time applying the partial dense matching for the correspondence problem. Recently, a method called the Non-Rigid Dense Correspondence (NRDC) (HaCohen et al., 2011) has shown effective results in finding dense correspondence between partially overlapped images. Using the NRDC for the matching, we present color transfer results between images with severe scene changes in Fig. 9 with comparison to the color transfer results in (HaCohen et al., 2011). Because the approach in (HaCohen et al., 2011) transfers each color channel separately similar to the Brightness Transfer Function in (Kim and Pollefeys, 2008), they are less effective in representing the color channel mixes. As shown in the figure, our B-PMLS method shows better color transfer results especially for more saturated color in the sea, grasses, and the clothing.

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. 10. 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. 11. 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. 12. It is worthy note that our video color transfer can be applicable for the other video frames that have the same scene as the retouched frame. If there are new scene contents that have severely different colors from the retouched frame, there may be erroneous color transfer for those regions. Otherwise, the video transfer results show convincing performance even for human movement, colored highlight changes, and colorful scene contents thanks to enough control points and our B-PMLS based interpolation. The datasets used for the evaluation and our video transfer results are open to public on our website<sup>3</sup>.

## 5. Discussion

We have presented a new mechanism for transferring color between images using a probabilistic moving least squares framework. Our color transfer framework can be applied to many instances of color variation such as different camera, different camera setting, and global/local tonal retouch very well as seen in the paper. 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. In the future, we would like to explore extending our framework for N-view case where there are more than 2 overlapping images (Xiong and Pulli, 2010b) by employing a joint optimization scheme. For image-based large-scale 3D reconstruction (Agarwal et al., 2011; Snavely et al., 2008), they simply took the average of pixel colors in the visible images for associating a color of 3D points. The proposed method can be used to generate the optimal color values by considering multiple color matches among the visible images.

## Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2016R1A2B4014610), by Ministry of Culture, Sports and Tourism(MCST) and Korea Content Agency(KOCCA) in the Culture Technology(CT) Research & Development Program 2015, and the Cross-Ministry Giga KOREA Project (MSIT) No. GK18P0200.

---

<sup>3</sup><https://sites.google.com/site/unimono/pmls>

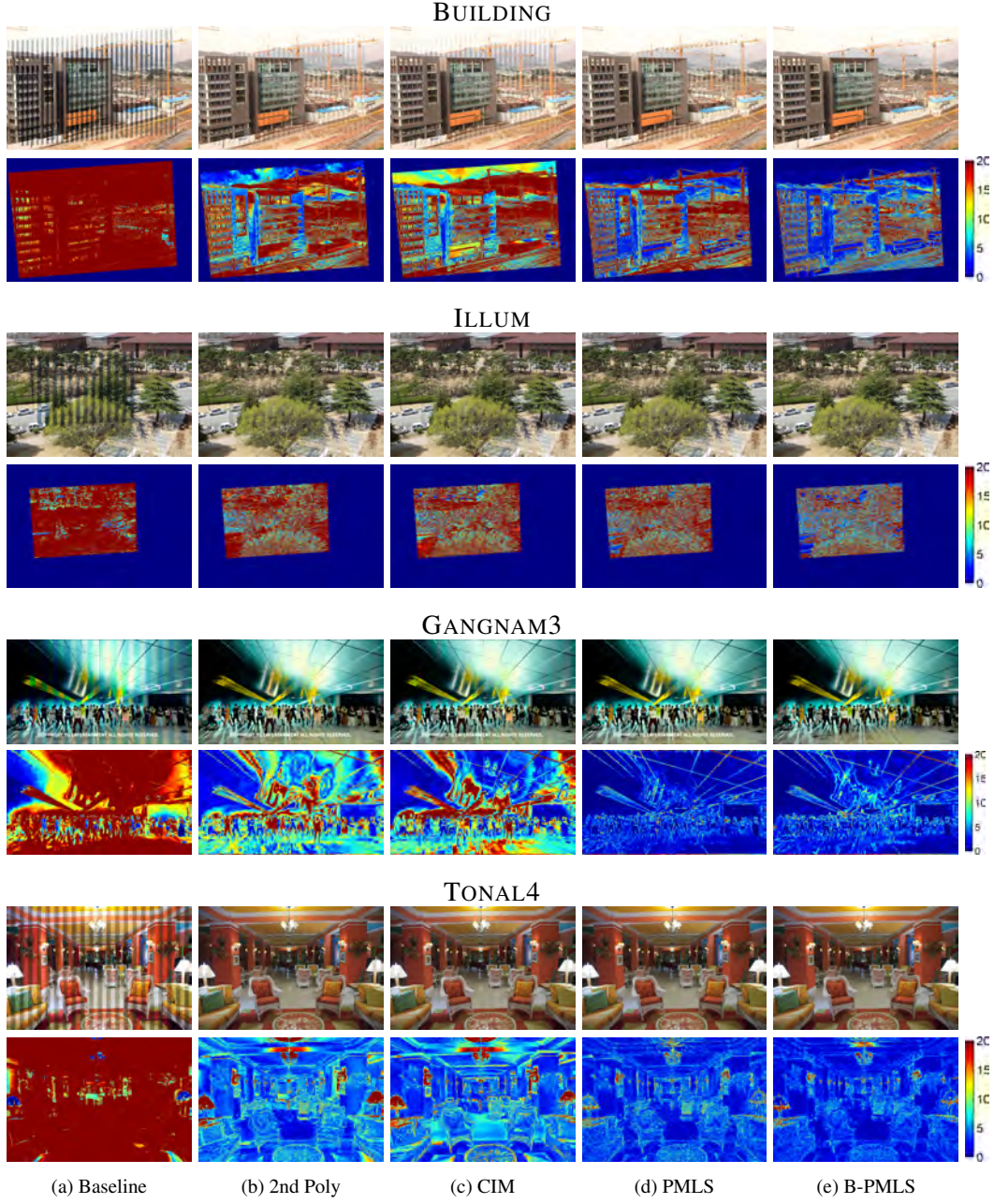


Figure 7: More qualitative and quantitative results. The reference image and the color transferred image are overlaid by column blocks. Using our algorithm, the mosaic images look as if they are just a single image, while the results from the other methods show clear discrepancy. We recommend the readers to zoom-in to see the difference clearly.

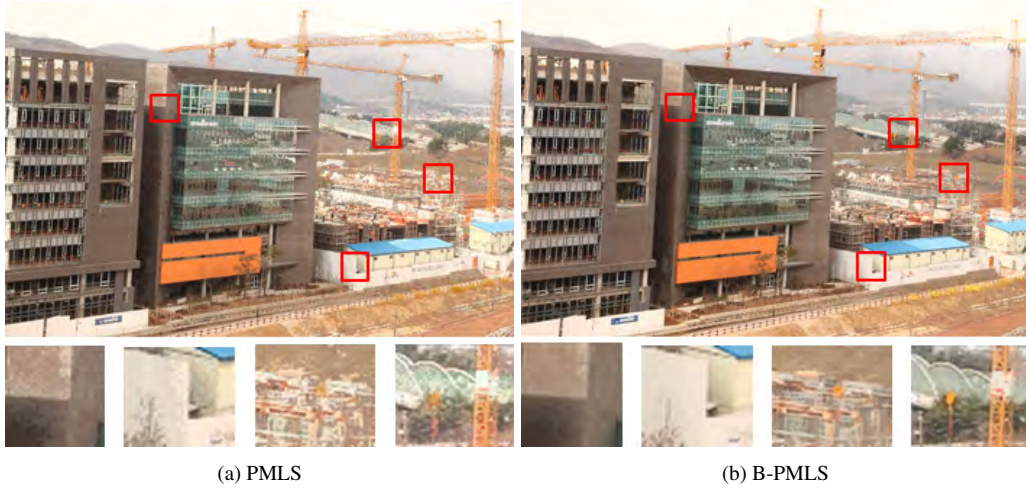


Figure 8: More detailed comparison between PMLS and B-PMLS. Because the input and the reference images have a large baseline, there are many outliers in the control points. By enforcing the spatial constraints, B-PMLS shows more robust color transfer results as shown in the zoomed-in views.



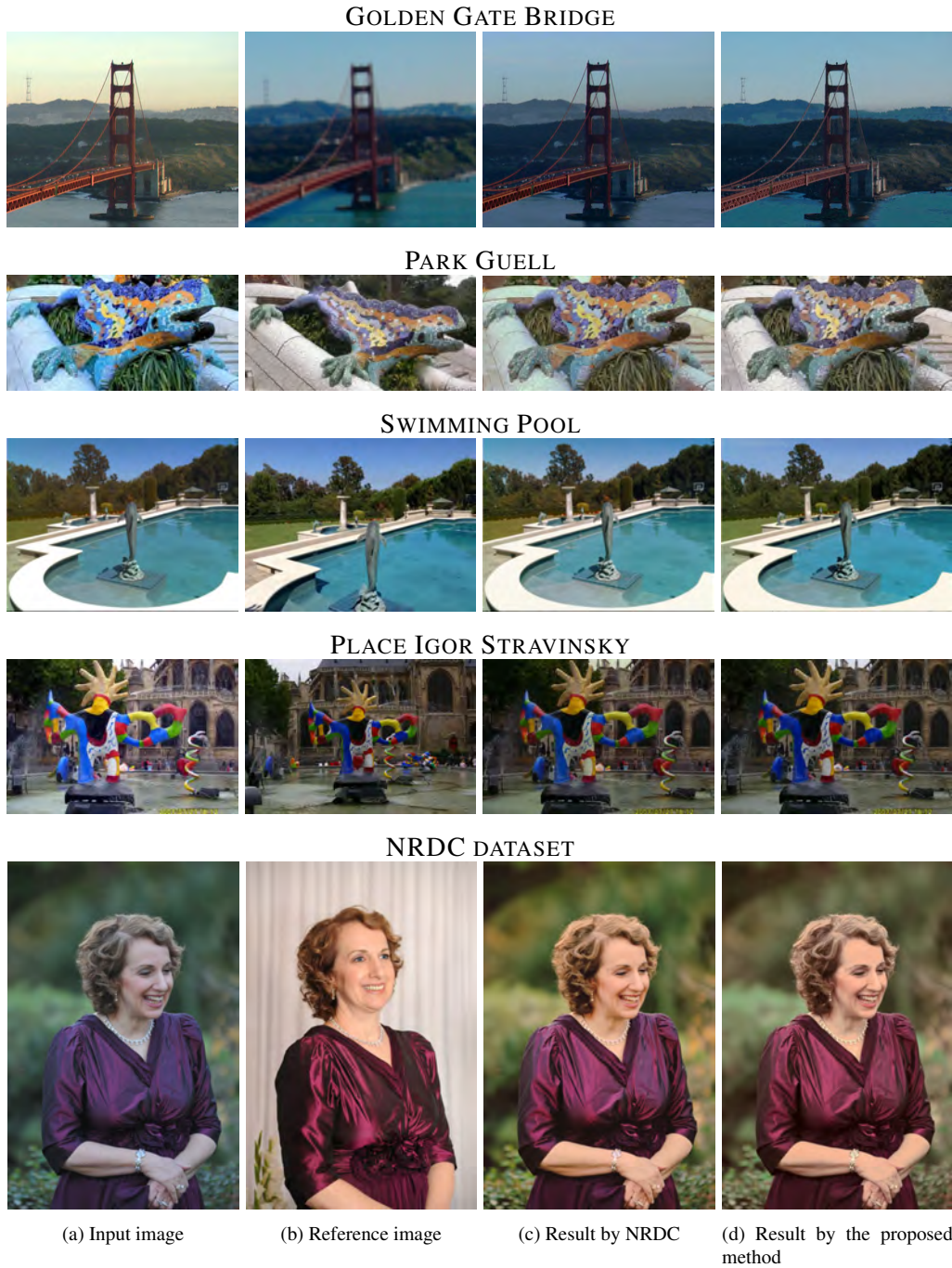


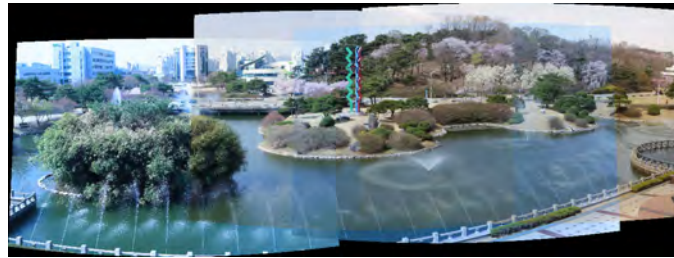
Figure 9: Color transfer results based on using the NRDC for finding the correspondences (HaCohen et al., 2011). Our transfer results are compared to the color transfer results in (HaCohen et al., 2011).



(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 (Brown and Lowe, 2007)



(e) Panorama using the corrected images

Figure 10: Photometric alignment using our color transfer method for generating a color consistent panorama. Geometric alignment is obtained from (Brown and Lowe, 2007).



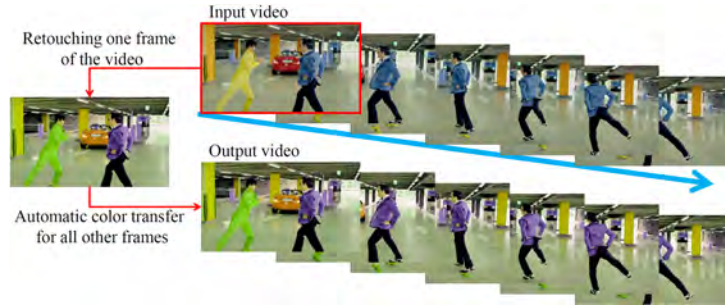


Figure 11: Workflow of the video color transfer. The user retouches the color of just one frame and the color transfer is computed between the original image and the edited image. The same color transfer is applied to all other frames for the video color transfer.



Figure 12: Video transfer application. (Top) Original Video (Bottom) Color transfered video where the color transfer is computed from just one example pair.



## References

- Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., Szeliski, R., Oct. 2011. Building rome in a day. *Commun. ACM* 54 (10), 105–112.  
URL <http://doi.acm.org/10.1145/2001269.2001293>
- Bonneel, N., Peyré, G., Cuturi, M., 2016. Wasserstein barycentric coordinates: Histogram regression using optimal transport. *ACM Transactions on Graphics* 35 (4).
- Bonneel, N., Sunkavalli, K., Paris, S., Pfister, H., 2013. Example-based video color grading. *ACM Trans. Graph.* 32 (4), 39–1.
- Bose, N. K., Ahuja, N. A., 2006. Superresolution and noise filtering using moving least squares. *IEEE Transactions on Image Processing* 15 (8), 2239–2248.
- Brainard, D. H., Freeman, W. T., 1997. Bayesian color constancy. *Journal of the Optical Society of America A* 14, 1393–1411.
- Brown, M., Lowe, D. G., 2007. Automatic Panoramic Image Stitching using Invariant Features. *International Journal of Computer Vision* 74 (1), 59–73.
- Bychkovsky, V., Paris, S., Chan, E., Durand, F., 2011. Learning photographic global tonal adjustment with a database of input / output image pairs. In: *CVPR*. pp. 97–104.
- Faridul, H., Stauder, J., Kerver, J., Trémeau, A., 2013. Approximate cross channel color mapping from sparse color correspondences. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. pp. 860–867.
- Faridul, H. S., Pouli, T., Chamaret, C., Stauder, J., Tremeau, A., Reinhard, E., 2014. A Survey of Color Mapping and its Applications. In: Lefebvre, S., Spagnuolo, M. (Eds.), *Eurographics 2014 - State of the Art Reports*.
- Fleishman, S., Cohen-Or, D., Silva, C. T., 2005. Robust moving least-squares fitting with sharp features. In: *ACM SIGGRAPH 2005*. pp. 544–552.
- Freedman, D., Kisilev, P., June 2010. Object-to-object color transfer: Optimal flows and smp transformations. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. pp. 287–294.

- Frigo, O., Sabater, N., Demoulin, V., Hellier, P., 2014. Optimal transportation for example-guided color transfer. In: Asian Conference on Computer Vision. pp. 655–670.
- Gijsenij, A., Gevers, T., Van De Weijer, J., 2011. Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing* 20 (9), 2475–2489.
- Gong, H., Finlayson, G. D., Fisher, R. B., 2016. Recoding color transfer as a color homography. In: *Proceedings of British Machine Vision Conference*.
- HaCohen, Y., Shechtman, E., Goldman, D. B., Lischinski, D., 2011. Non-rigid dense correspondence with applications for image enhancement. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2011)* 30 (4), 70:1–70:9.
- HaCohen, Y., Shechtman, E., Goldman, D. B., Lischinski, D., 2013. Optimizing color consistency in photo collections. *ACM Transactions on Graphics (TOG)* 32 (4), 38.
- Hristova, H., Le Meur, O., Cozot, R., Bouatouch, K., 2015. Style-aware robust color transfer. In: *Proceedings of the workshop on Computational Aesthetics. Eurographics Association*, pp. 67–77.
- Huang, T.-W., Chen, H.-T., 2009. Landmark-based sparse color representations for color transfer. In: *ICCV*. pp. 199–204.
- Hwang, Y., Lee, J.-Y., Kweon, I. S., Kim, S. J., 2014. Color transfer using probabilistic moving least squares. In: *CVPR*.
- Ilie, A., Welch, G., 2005. Ensuring color consistency across multiple cameras. In: *ICCV*. pp. 1268–1275.
- Jia, J., Tang, C.-K., 2005. Tensor voting for image correction by global and local intensity alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (1), 36–50.
- Kim, S. J., Lin, H. T., Lu, Z., Ssstrunk, S., Lin, S., Brown, M. S., 2012. 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.

- Kim, S. J., Pollefeys, M., 2008. Robust radiometric calibration and vignetting correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 562–576.
- Lancaster, P., Salkauskas, K., 1981. Surfaces generated by moving least squares methods. *Mathematics of Computation* 87, 141–158.
- Lee, J.-Y., Sunkavalli, K., Lin, Z., Shenand, X., Kweon, I. S., 2016. Automatic content-aware color and tone stylization. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W. T., 2008. Sift flow: Dense correspondence across different scenes. In: *ECCV*. pp. 28–42.
- Nguyen, R., Kim, S., Brown, M., 2014. Illuminant aware gamut-based color transfer. In: *Computer Graphics Forum*. Vol. 33. pp. 319–328.
- Oliveira, M., Sappa, A. D., Santos, V., 2011. Unsupervised local color correction for coarsely registered images. In: *CVPR*. pp. 201–208.
- Oliveira, M., Sappa, A. D., Santos, V., 2015. A probabilistic approach for color correction in image mosaicking applications. *IEEE Transactions on Image Processing* 24 (2), 509–523.
- Pitie, F., Kokaram, A., 2007. The linear monge-kantorovitch colour mapping for example-based colour transfer. In: *IEE European Conference on Visual Media production (CVMP'07)*.
- Pitie, F., Kokaram, A., Dahyot, R., 2005. N-dimensional probability density function transfer and its application to color transfer. In: *ICCV*. pp. 1434–1439.
- Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P., 2001. Color transfer between images. *IEEE Computer Graphics and Applications* 21 (5), 34–41.
- Sajadi, B., Lazarov, M., Majumder, A., 2010. Adict: Accurate direct and inverse color transformation. In: *European Conference on Computer Vision*. pp. 72–86.
- Schaefer, S., McPhail, T., Warren, J., 2006. Image deformation using moving least squares. In: *ACM SIGGRAPH 2006*. pp. 533–540.

- Snaveely, N., Seitz, S. M., Szeliski, R., Nov. 2008. Modeling the world from internet photo collections. *Int. J. Comput. Vision* 80 (2), 189–210.  
URL <http://dx.doi.org/10.1007/s11263-007-0107-3>
- Tai, Y.-W., Jia, J., Tang, C.-K., 2005. Local color transfer via probabilistic segmentation by expectation-maximization. In: *CVPR*. pp. 747–754.
- Tai, Y.-W., Jia, J., Tang, C.-K., 2007. Soft color segmentation and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (9), 1520–1537.
- Tola, E., Lepetit, V., Fua, P., 2010. Daisy: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (5), 815–830.  
URL <http://cvlab.epfl.ch/~tola/daisy.html>
- Tsai, Y.-H., Shen, X., Lin, Z., Sunkavalli, K., Yang, M.-H., 2016. Sky is not the limit: semantic-aware sky replacement. *ACM Transactions on Graphics (TOG)* 35 (4), 149.
- Wang, Z., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P., 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions On Image Processing* 13 (4), 600–612.
- Wu, F., Dong, W., Kong, Y., Mei, X., Paul, J.-C., Zhang, X., 2013. Content-based colour transfer. In: *Computer Graphics Forum*. Vol. 32. pp. 190–203.
- Xiong, Y., Pulli, K., 2010a. Color and luminance compensation for mobile panorama construction. In: *Proceedings of the 18th ACM international conference on Multimedia*. ACM, pp. 261–270.
- Xiong, Y., Pulli, K., 2010b. Color matching of image sequences with combined gamma and linear corrections. In: *ACM Multimedia*.
- Xu, W., Mulligan, J., 2010. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In: *CVPR*. pp. 263–270.
- Zaragoza, J., Chin, T.-J., Tran, Q.-H., Brown, M., Suter, D., July 2014. As-projective-as-possible image stitching with moving dlt. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36 (7), 1285–1298.