Color-aware Regularization for Gradient Domain Image Manipulation

Fanbo Deng[1], Seon Joo Kim[2], Yu-Wing Tai[3], and Michael S. Brown[1]

[1] National University of Singapore
[2] SUNY Korea
[3] Korea Advanced Institute of Science and Technology

Abstract. We propose a color-aware regularization for use with gradient domain image manipulation to avoid color shift artifacts. Our work is motivated by the observation that colors of objects in natural images typically follow distinct distributions in the color space. Conventional regularization methods ignore these distributions which can lead to undesirable colors appearing in the final output. Our approach uses an anisotropic Mahalanobis distance to control output colors to better fit original distributions. Our color-aware regularization is simple, easy to implement, and does not introduce significant computational overhead. To demonstrate the effectiveness of our method, we show the results with and without our color-aware regularization on three gradient domain tasks: gradient transfer, gradient boosting, and saliency sharpening.

1 Motivation and Related Work

Gradient domain manipulation is the cornerstone of many image processing algorithms from image editing to texture transfer to image fusion. For an overview of gradient domain algorithms and applications we refer readers to [1]. As the name implies, gradient domain algorithms do not operate in the 0th order domain (i.e. color domain), but instead impose changes to the 1st order derivatives of the input image, i.e. the image gradient. When left unchecked, gradient domain processing can result in noticeable color shifts in the 0th domain output image. To ameliorate color-shifting artifacts, most gradient domain approaches impose an additional 0th order constraint either at the boundary of the processed region or over the entire region.

Early gradient domain processing approaches (e.g. [2–5]) were formulated using the Poisson equation (see [6]) which incorporates a 0th order boundary constraint on the solution, i.e. the Dirichlet boundary condition. While generally sufficient for most processes, this method can, from time to time, exhibit very noticeable color shifts inside the processed region. As a result, other approaches, especially more recent ones (e.g. [1, 7–11]) impose a regularization over the entire 0th order solution. This is typically done using an L_2 norm regularization on one or more of the 0th order image channels. This solution results in a bi-objective function that tries to manipulate the image gradient while minimizing

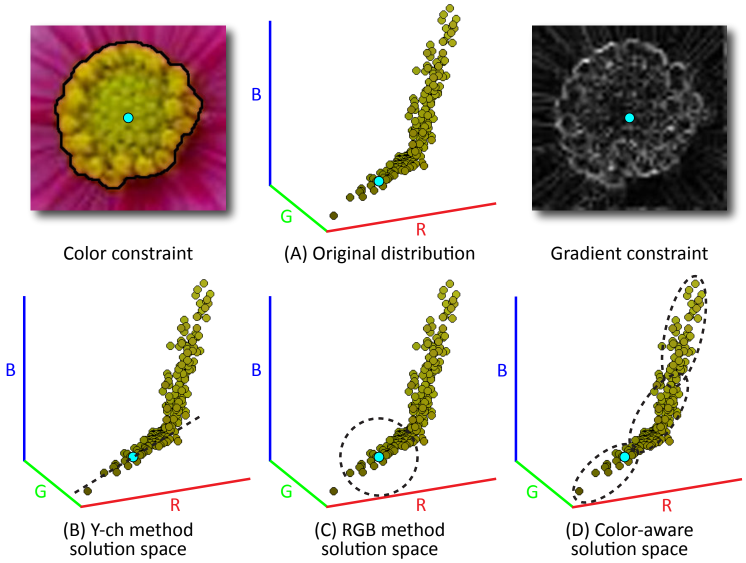


Fig. 1. Solution spaces (denoted by the dotted line) of the marked pixel using different 0th domain regularization methods.

the Euclidean error (i.e. L_2) between the original and output 0th order domains.

This paper targets this latter regularization strategy which is applied in one of two manners, either by 1) first converting the input space (presumably RGB) to a new color space (e.g. YUV or LAB) that separates the luminance and chrominance components and processing only the luminance channel (we refer to this manner as Y-ch method in the rest of this paper); or by 2) applying the L_2 regularization to all three channels separately (we refer to this manner as RGB method). When only one channel is processed, the regularization effectively constrains the output solution so that each pixel is restricted to a 1-D space (Figure 1(B)). Although this approach does not shift the chromacity, it can produce outputs that appear flat and less vivid. This can be seen in Figure 2(B). When all three channels are processed, the per pixel solution space is constrained to lie within the sphere about its original value as shown in Figure 1(C). This conventional regularization is applied irrespective to how the scene colors are distributed in the original input. As a result, satisfying the regularization constraint may also introduce colors that are quite different than those in the original image. This can be seen in Figure 2(C) where the solution of the gradient boosting has resulted in noticeable color shifts.

Our work is motivated by the observation that objects' RGB colors in natural images follow unique distributions. For example, in Figure 1(A), the pixel marked in cyan is plotted with all other pixels belonging to the same object. It is easy to see that the pixel belongs to a distinct color distribution in the RGB space.

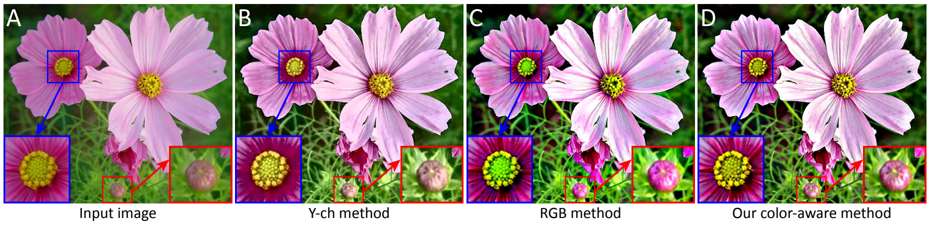


Fig. 2. This figure compares conventional 0th domain regularization applied to an image that has had its gradient boosted. A) Input image. B) Result using L_2 regularization over the Y channel only. C) Result using L_2 regularization over all three channels of the RGB input. D) Our color-aware regularization result. Note the flat output colors exhibited by Y-ch method in B, and the subtle color-shifting exhibited by RGB method in C.

Such unique distributions observed by Omer and Werman [12] have shown that colors in natural images tend to form elongated clusters (referred to as lines) in the RGB space. Our color-aware approach constrains the solution space to more tightly follow the original distribution in order to avoid color shifting as shown in Figure 1(D) and Figure 2(D).

Contribution Our contribution is the introduction of a regularization term that more faithfully follows the distribution of colors in the input image. Our approach applies a simple segmentation to the input image to assign each pixel to a color distribution represented as a Gaussian mixture model (GMM). Using these GMMs we can formulate the color-aware regularization as an anisotropic Mahalanobis distance [13] which can be expressed as a linear system. This color-aware regularization constrains the output solution to better fit the original input color distributions thereby avoiding color shifts. Our approach can be easily incorporated into existing gradient-domain formulations. We demonstrate the effectiveness of this regularization on a variety of inputs using three selected applications, gradient transfer, gradient boosting and saliency sharpening. We compare our results with conventional L_2 regularization approaches (Y-ch method and RGB method) as used by [1, 7–11].

2 Color-aware Regularization Framework

2.1 Overview

An overview of our framework is shown in Figure 3. Each pixel is first assigned to a color distribution via segmentation. We found that a simple superpixel segmentation [14] followed by k-means clustering [13] is enough to find the underlying color distributions. These individual color distributions are then fit with a series of 3D Gaussian distributions in the RGB color space. The input to our algorithm is an image where each pixel is assigned to a single distribution represented by a series of Gaussians, i.e. $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m$.

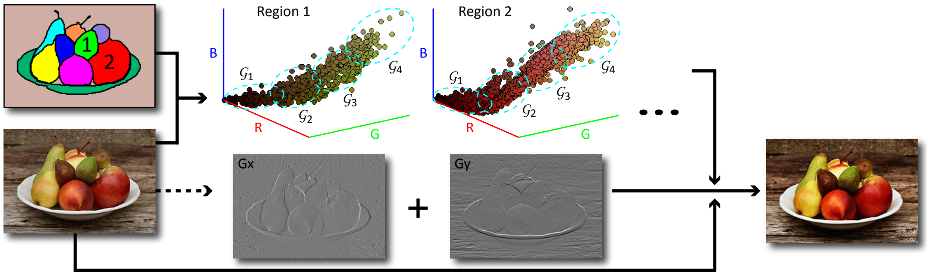


Fig. 3. The overall workflow of our color-aware regularization framework.

A bi-objective function is then used to transfer the new gradients to the input while regularizing each output pixel to lie within a minimum distance from one of the Gaussian distributions used to model its associated color distribution. This regularization is formulated as an optimization problem similar to [1, 7–11].

2.2 Conventional optimization framework

Taking gradient transfer application as example, we review the conventional optimization framework based on an L_2 regularization term. The purpose of gradient transfer is to transform gradients from the source image to the target image while preserving the original look-and-feel of the target image.

Given two images f and g, we seek a new image u whose colors (from one or more color channels) are as close as possible to f, and at the same time, has gradients that are as close as possible to g. More formally, the final result u is generated by minimizing the following bi-objective cost function

$$E(u) = \sum_{p \in u} (\lambda E_d(p) + E_s(p)), \tag{1}$$

where p is the pixel index of image u; E_d is the 0th order color constraint term and E_s is the 1st order gradient constraint term; λ is used for the balance between E_d and E_s. These two terms are defined as:

$$E_d(p) = (u_p - f_p)^2, \tag{2}$$

$$E_s(p) = \left(\frac{\partial u}{\partial x} - c \cdot \frac{\partial g}{\partial x}\right)_p^2 + \left(\frac{\partial u}{\partial y} - c \cdot \frac{\partial g}{\partial y}\right)_p^2, \tag{3}$$

where $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ denote the partial derivative operators in x- and y-direction; c is a scaling factor used in gradient boosting and saliency sharpening application and is set to 1.0 for gradient transfer application.

2.3 Color-aware regularization term

As shown in Figure 1, the solution space of each pixel in the resulting image u is constrained either to lie on a 1-D solution space if only a single channel is processed (Y-ch method), or to lie within a sphere centered at each pixel if applied to all three channels (RGB method). Since the Euclidean distance is blind to the inherent correlation among variables, neither of these methods is able to take into account the color distribution information of the input image f. This can lead to flattened colors or noticeable color shifts in the output image. To solve this problem, we change the conventional L_2 regularization to an anisotropic Mahalanobis distance that more tightly fits the original color distribution. By using the Mahalanobis distance, 0th domain solutions along the shorter axis of each pixel's associated Gaussian model are penalized. This forces the solution to move along the longer axis, thus constraining the solution to lie closer to the original color distribution.

Single Gaussian Model We first consider the case where we can model a color distribution using a single Gaussian distribution. We define our color-aware 0th order regularization term as:

$$E_{mdd}(p) = (\boldsymbol{u}_p - \boldsymbol{f}_p)^T S_p^{-1} (\boldsymbol{u}_p - \boldsymbol{f}_p), \tag{4}$$

where p is the pixel index; both $\boldsymbol{u}_p$ and $\boldsymbol{f}_p$ are the RGB pixel values represented by 3D column vectors; S_p is a 3×3 covariance matrix of the Gaussian that pixel p is assigned to. The term E_{mdd} is the squared Mahalanobis distance, which is a dissimilarity measure between the two vectors $\boldsymbol{u}_p$ and $\boldsymbol{f}_p$. The benefit of the Mahalanobis distance is that, unlike the Euclidean distance, it considers the correlation of data elements in the vector, in our case the pixels' RGB values.

Combining Eq. 3 and Eq. 4 using matrix notation we can write our quadratic form bi-objective cost function as

$$\begin{aligned} E(u) &= \lambda E_{mdd}(u) + E_s(u) \\ &= \lambda (u - f)^T \Sigma (u - f) \\ &\quad + (G_x u - c \cdot G_x g)^T (G_x u - c \cdot G_x g) \\ &\quad + (G_y u - c \cdot G_y g)^T (G_y u - c \cdot G_y g), \end{aligned} \tag{5}$$

where u, f and g are RGB images reshaped into the column vector form (e.g. $[R_1 G_1 B_1 \ldots R_N G_N B_N]^T$); Σ is a 3N×3N (N is the number of pixels) block-diagonal matrix containing the 3×3 inverse covariance matrices of Gaussian models that each pixel is assigned to; the matrices G_x and G_y are discrete forward differentiation operators. Note that gradient constraint g does not necessarily form a 3-channel image since we may transfer gradients of a grayscale image to a color image (see Section 3). In that case, image g is extended to an RGB image by copying itself three times. Minimizing Eq. 5 amounts to taking its derivative, setting it to zero, and solving for vector u that is uniquely defined as the solution of the linear system:

$$(\lambda \Sigma + G_x^T G_x + G_y^T G_y) u = \lambda \Sigma f + c \cdot (G_x^T G_x g + G_y^T G_y g). \tag{6}$$

To solve this linear system, we use the conjugate gradient (CG) method [15] that is also used by [16] and [1]. Note that further improvement can be made

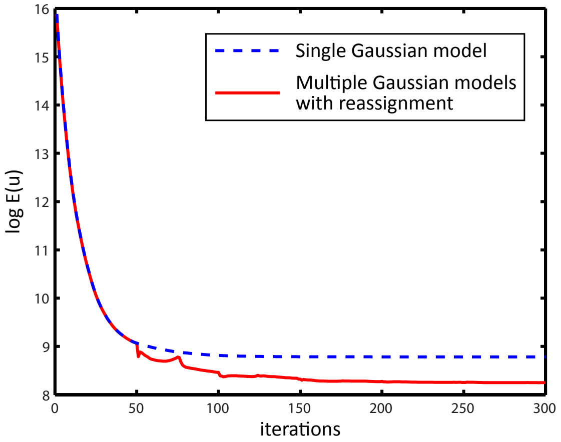


Fig. 4. Comparison of cost values (with spatial-varying weights applied) when using single Gaussian model (blue dashed line) and multiple Gaussian models (red solid line). For multiple Gaussian models, the reassignment operation is carried out every 50 iterations ($t = 50$ in CG solver).

to the 1st order term $E_s(u)$ in Eq. 5 since the L_2 norm is known to be sensitive to noise and may result in haloing artifacts in the output image. To solve this problem, we add two spatial-varying weights to $E_s(u)$ using the same weighting scheme presented in [1]:

$$w^x(p) = \left(\left| \frac{\partial f}{\partial x} - \frac{\partial g}{\partial x} \right|_p + 1 \right)^{-\alpha} \tag{7}$$

$$w^y(p) = \left(\left| \frac{\partial f}{\partial y} - \frac{\partial g}{\partial y} \right|_p + 1 \right)^{-\alpha} \tag{8}$$

where the parameter α (typically $1.2 \leq \alpha \leq 3$) determines the sensitivity of $E_s(u)$ to noise. By using this per-pixel weighting scheme halo artifacts are effectively reduced.

Multiple Gaussian Models Instead of using a single Gaussian model per color distribution, we use several Gaussian models to represent each color distribution more precisely. As shown in Figure 3, each pixel is first assigned to a color distribution (region) via segmentation. Each color distribution is represented by a series of 3D Gaussian models $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m$ and each pixel is initially assigned to its nearest Gaussian model $\mathcal{G}_i$ via Eq. 4. All pixels in the same region (color distribution) share the same set of Gaussian models and can be reassigned to any Gaussian models within this set when iteratively solving the output image. We integrate this reassignment scheme with the conjugate gradient algorithm and show that it can further decrease the objective cost function (see Figure 4).

Assume that we divide the input image into k color distributions and each distribution is represented by m 3D Gaussian models, resulting in $k \times m$ Gaussian models in total; $\mathcal{G}_{i,j}$ is the jth Gaussian model of the ith color distribution

Algorithm 1 Gaussian model reassignment

```
Require: input image $f$ and $g$, initial assignment map of all pixels $ASG$ (a matrix),
    maximum number of reassignment operations $T$, number of CG solver iterations $t$,
    small tolerance $\epsilon > 0$
$u = f$
for $reselect = 1$ to $T$ do
    $u =$ conjugate_gradient_solver$(f, g, u, ASG, t)$; $ASG\_old = ASG$;
    for $i = 1$ to $k$ do
        for all $p \in Region(i)$ do
            $j_0 = \underset{j \in [1,m]}{\operatorname{argmin}} \ (\boldsymbol{u}_p - \boldsymbol{f}_p)^T S_{i,j}^{-1} (\boldsymbol{u}_p - \boldsymbol{f}_p)$
            reassign $p$ to the Gaussian model $\mathcal{G}_{i,j_0}$ (one element of $ASG$ is updated)
        end for
    end for
    if $\|ASG - ASG\_old\|_F < \epsilon$ then
        break
    end if
end for
return the output image $u$
```

$(1 \leq i \leq k; 1 \leq j \leq m)$ and $S_{i,j}$ is the covariance matrix of $\mathcal{G}_{i,j}$. The expression $\|A\|_F$ denotes the Frobenius norm of matrix A. We now outline the overall algorithm of our reassignment approach as shown in Algorithm 1.

Convergency Analysis Without using spatial-varying weights on the 1st order constraint term, minimizing the conventional bi-objective cost function reviewed in Section 2.2 is known to be a convex problem. Our color-aware optimization framework (using single Gaussian model) differs from the conventional formulation by only introducing a block-diagonal matrix Σ on both sides of the linear system $Au = b$ (see Eq. 6). We know that the covariance matrix Σ is positive-semidefinite. As a result, introducing the matrix Σ does not violate the convex property of this optimization problem and a global optimal solution exists.

When using multiple Gaussian models and the reassignment scheme, the convex property remains intact. As shown in Algorithm 1, the reassignment scheme is actually a combination of several independent conjugate gradient solving procedures. After each reassignment step is done, the cost value is guaranteed to be decreased (or at least remain unchanged) by reassigning each pixel to the Gaussian model whose covariance matrix can minimize the 0th order term E_{mdd} while keeping the 1st order term E_s unchanged.

However, the optimization problem is no longer convex once the spatial-varying weights are used. In this case, the global optimum solution may not exist, but we can still use conjugate gradient method to find an appropriate solution. In practice, we find our framework works well to minimize Eq. 5 within 250 iterations (see Section 3.1). Two plots of the cost values during conjugate gradient iterations are shown in Figure 4. As we can see, with the help of multiple Gaussian models and the reassignment scheme, the cost value has been further decreased compared to the result achieved by the single Gaussian model. Note that the cost values are shown in *log* scale.

3 Experiments

We compare results obtained by our color-aware regularization against those obtained using a conventional optimization framework [1, 7–11] based on L_2 0th order regularization in the two manners previously discussed (i.e Y-ch method and RGB method). The fast deconvolution algorithm presented by [17] is used to perform the conventional optimization. Comparisons are conducted on three selected tasks including gradient transfer, gradient boosting and saliency sharpening. Before carrying out experiments we briefly explain the parameters we used for these tasks.

3.1 Experiment setups

For all the three methods, the gradient scaling factor c is set to 1.0 in gradient transfer task and 2.0 or 3.0 in gradient boosting/saliency sharpening tasks. To keep the comparisons fair, we adjust the balancing factor λ for each method to make sure that a comparable amount of gradient has been transferred or boosted for each example (see quantitative comparison in Section 3.3).

For our color-aware regularization method, we use over segmentation algorithm followed by k-means clustering to detect underlying color distributions of an image and k is chosen from [10, 15] range (see Section 3.3 for detail explanation). The number of Gaussian models used to represent each color distribution is fixed to $m = 5$. We restrict the number of Gaussian reassignment operations within 5 times ($T = 5$) and set 50 iterations for the CG solver ($t = 50$). With the above settings, the running time for an 800×600 image is around 3 minutes (Matlab implementation on an Intel Core 2 Duo 2.8GHz computer). We note using more than 5 Gaussian models does not significantly improve the results.

3.2 Image gradient manipulation tasks

Gradient transfer The first two examples demonstrate gradient transfer of the gradients from a near-infrared (NIR) image to an ordinary RGB image. Such gradient transfer has been demonstrated to improve some forms of photography [8, 18] since NIR often contain more details that cannot be seen in the visible spectrum. In the first example, we show an example of an outdoor scene of a castle where the clouds and other textures are notably stronger in the NIR image. Two input images (NIR and RGB) are shown in Figure 5(A-a) and Figure 5(A-b). Figure 5(A-c) shows the result generated by the Y-ch method. While the desired gradients (clouds) are transferred, the color of the green plants below the castle change to cyan. Figure 5(A-d) shows the result produced by the RGB method. Note that the red color of the plants and rocks changes to green. Our result is shown in Figure 5(A-e). The colors of both the red rocks and green plants are

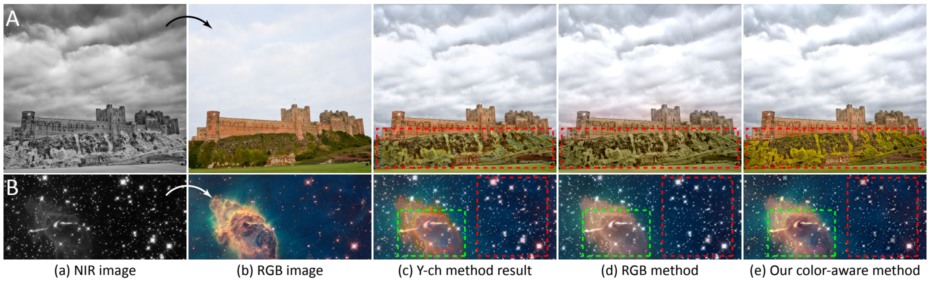


Fig. 5. Examples of gradient transfer: (a) input NIR image; (b) input RGB image; (c) result using L_2 regularization over the Y channel only; (d) result using L_2 regularization over R/G/B channels; (e) our color-aware regularization result. Regions with color-shifting problem have been highlighted in red and green dashed boxes.

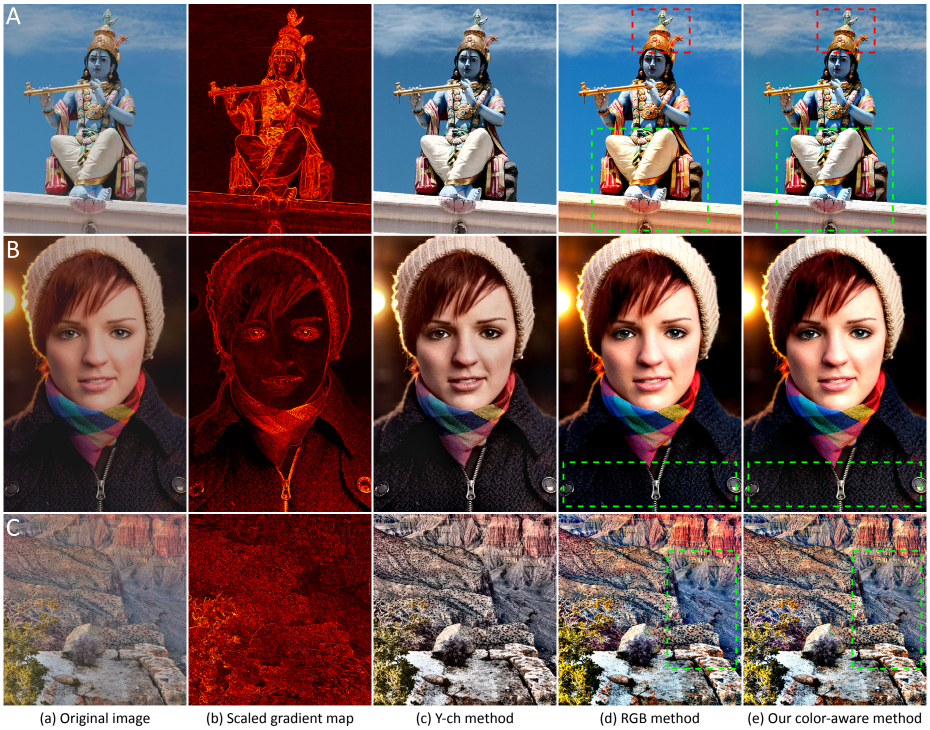


Fig. 6. Examples of gradient boosting: (a) input RGB image; (b) scaled gradient map providing target gradients; (c) result using L_2 regularization over the Y channel only; (d) result using L_2 regularization over R/G/B channels; (e) our color-aware regularization result. Regions with color-shifting problem have been highlighted in green dashed boxes.

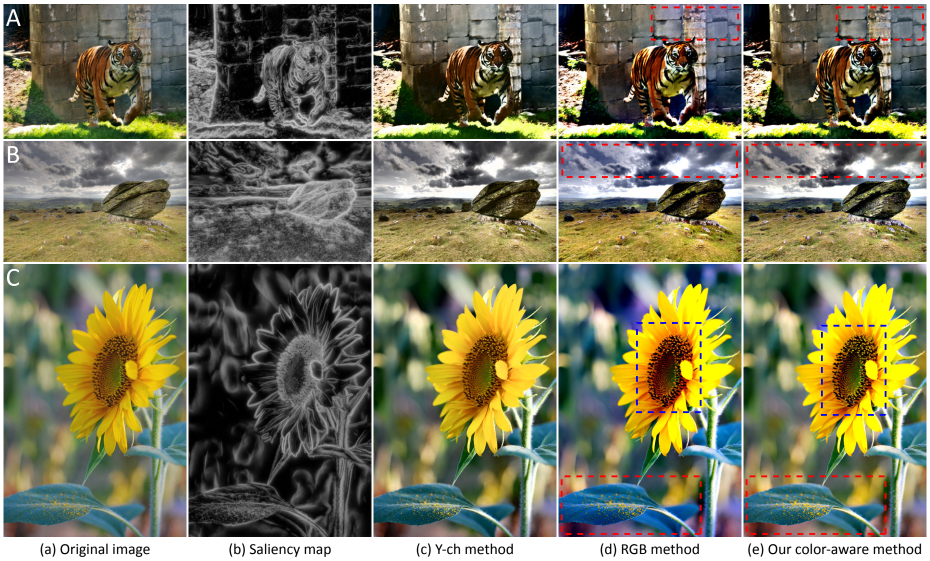
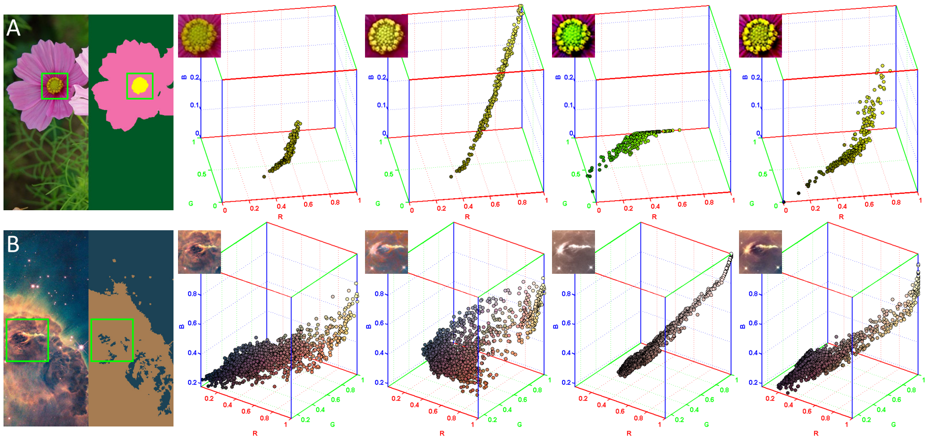


Fig. 7. Examples of saliency sharpening: (a) input RGB image; (b) saliency map of the input image; (c) result using L_2 regularization over the Y channel only; (d) result using L_2 regularization over R/G/B channels; (e) our color-aware regularization result. Regions with color-shifting problem have been highlighted in red and blue dashed boxes.

preserved well. Another example is shown in the second row of Figure 5. Note that the color of the nebula (highlighted by a green dashed box) changes significantly in Figure 5(B-c) and the color of the stars (highlighted by a red dashed box) is washed out in Figure 5(B-d). Our method achieves a better result in Figure 5(B-e) with colors that are more similar to the input RGB image.

Gradient boosting The second example targets gradient boosting that is aimed to enhance image contrast. In Figure 6, column (a) shows original input images; column (b) shows the scaled gradient magnitudes after boosting (rendered as a *hot map* for better visualization); column (c), (d) and (e) are the results generated by the three different methods. We can see that when using the RGB method (column (d) in Figure 6), the results suffer from noticeable color-shifting in some regions. For instance, the color of the wall and the Buddha's legs in example A become yellowish; the color of the woman's clothing in example B changes from brown to blue; the color of the valley in example C also shifts to blue. Although less color shifts is noticeable when using the Y-ch method (column (c) in Figure 6), the overall color of these images seems to be flattened and less vivid. Our results (column (e) in Figure 6) show the images with boosted contrast and without color shifts. In addition, our results are more vivid and colorful compared to the Y-ch method.

Saliency sharpening Saliency sharpening is similar to gradient boosting application. The only difference is that the gradient boosting globally enhances gradients by a factor c, while saliency sharpening strengthens gradients in a



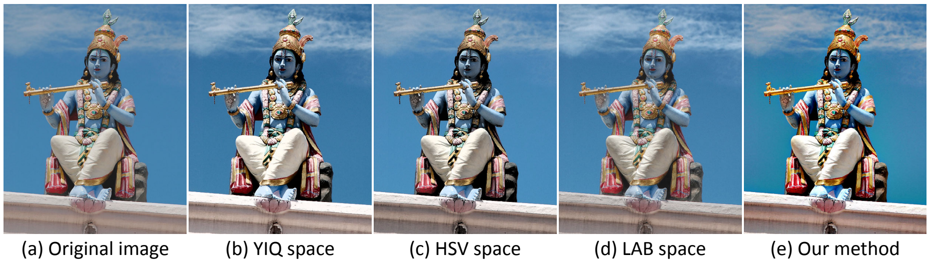
(a) Input RGB image and its segmentation map (b) Original distribution (c) Y-ch method (d) RGB method (e) Our color-aware mehtod

Fig. 8. Distributions of the solutions using different 0th domain regularization methods: (a) input RGB image and its segmentation map; (b) original color distribution of the selected region (highlighted in green solid boxes); (c) resulting distribution using L_2 regularization over the Y channel only; (d) resulting distribution using L_2 regularization over R/G/B channels; (e) our color-aware regularization distribution. Note that our distribution better maintains the shape and trend of the original.

spatially varying manner based on the image saliency map. We adopted the gradient attenuation function proposed in [2] to generate a grayscale saliency map M (brighter regions indicate larger scale factors and stronger boosting). In this case, the global scale factor c in Eq. 5 will be replaced by $(1+c \cdot M)$. As shown in Figure 7, our method produces results visually more appealing compared to the other two methods. Note the visible color-shifting on the wall behind the tiger (Figure 7(A-d)), the cloud above the rock (Figure 7(B-d)) and the sunflower (Figure 7(C-d)). Again, results from the Y-ch method (column (c) in Figure 7) appear flat similar to the examples in gradient boosting application. However, our results (column (e) in Figure 7) successfully preserve the original color of input images after saliency sharpening process.

3.3 Evaluation and analysis

In order to show how our color-aware regularization method preserves the original color distribution more faithfully than the other two methods, we plot the original color distribution of a selected region in the input image and compare it with color distributions of the same region in three output images. In Figure 8, column (a) shows the input image and its color-coded segmentation map; column (b) plots the color distribution (data points are randomly sub-sampled for better visualization) of the selected region in the RGB space; column (c), (d) and (e) plot different results generated by the Y-ch method, the RGB method and our method respectively. The plots show that the color distribution of our



(a) Original image (b) YIQ space (c) HSV space (d) LAB space (e) Our method

Fig. 9. Comparison of other color spaces: (a) input RGB images; (b), (c) gradient boosted results using L_2 regularization over the luminance/brightness channel of YIQ/HSV color spaces; (d) result of L_2 regularization over all channels of LAB color space; (e) our color-aware regularization result.

output image (selected regions) preserves the original distribution much more faithfully than the other two in terms of shape and trend.

Other than YUV and RGB spaces, we also compared our method with traditional L_2 regularization applied on other commonly-used color spaces. Similar to the Y-ch method, we convert the input image into YIQ/HSV color space and regularize the luminance/brightness channel only. As shown in Figure 9(b, c), the results are similar to that of the Y-ch method and also suffer from flattened colors due to the limitation that the output pixels are restricted to a 1-D space (refer to Figure 1(B)). Similar to the RGB method, we convert the input image into LAB color space and regularize three channels separately. Using LAB color space we get the result (Figure 9(d)) that also appears flat and less colorful compared to our result (Figure 9(e)).

Table 1. This table shows the overall amount of gradient transferred by each method (average L_2 difference between output and input gradients) is similar for all example images shown in Figure 5(A, B), Figure 6(A, B, C) and Figure 7(A, B, C).

Methods	Figure 5		Figure 6			Figure 7		
	A	B	A	B	C	A	B	C
Y-ch method	0.0040	0.0047	0.0369	0.0148	0.0899	0.0849	0.1344	0.0518
RGB method	0.0041	0.0036	0.0372	0.0123	0.0591	0.0757	0.1244	0.0467
Our method	0.0044	0.0046	0.0340	0.0113	0.0533	0.0747	0.1182	0.0453

We also want to examine the amount of gradient effectively transferred by each method. To do so, we compare the average per-pixel Euclidean distance of the gradient maps of three output images with the constrained gradient map. Table 1 lists the amount of gradient transferred for each example. Note that all methods transfer a comparable amount of gradient. This verifies that 1) our approach is able to transfer gradient as effective as the other methods; and 2) the results shown are fairly compared because they have each transferred approximately the same amount of gradient.

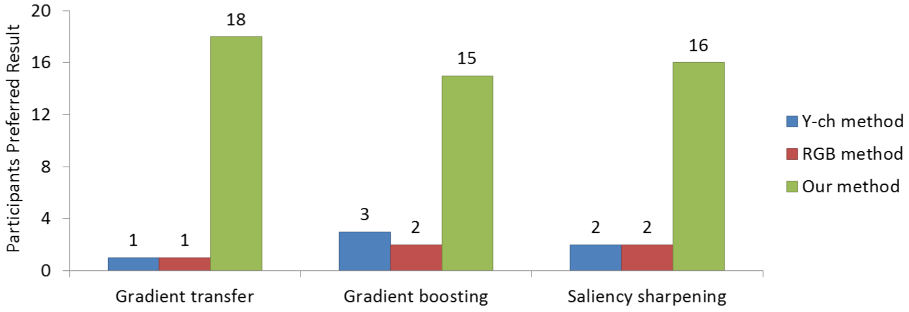


Fig. 10. Participants preferred results of three different methods.

Our color distributions are determined by over segmentation followed by k-means clustering, resulting in k distributions, each of which is further decomposed into GMMs. To determine the sensitivity of our results to the choice of k, we performed experiments with k ranging from 5 to 40. We found the results dose not vary too much for values of k greater than 15. As a result, we advocate using the range [10, 15].

Lastly, since our approach is subjective in nature, we performed a simple user study on user's preference of the results on 14 examples (3 for gradient transfer, 7 for gradient boosting and 4 for saliency sharpening). Twenty participants (average age around 25) were asked to choose their preferred results out of the outputs of the three different methods. Participants were not trained before the experiment, but over half of them had experience with image editing software such as Photoshop. Our user study showed that 18 participants preferred our results for the gradient transfer application, and 15 participants preferred our results for the gradient boosting application. For saliency sharpening application, 16 participants preferred the results produced by our color-aware regularization method. Figure 10 shows a graph of these results.

4 Conclusion

We have presented a straight-forward approach to perform 0th domain regularization in a manner that more faithfully follows the original input color distribution. This results in gradient transfer that avoids color shifting while still producing vivid results. While our approach requires an initial segmentation to determine the distinct color distributions in the image, we found that the segmentation stage is not a crucial issue and any basic over segmentation algorithm (e.g. watershed [19] or superpixel [14]) gave good results. More sophisticated segmentation algorithm like Ridge-based Distribution Analysis [20] were tried but generated similar results. We also note that our approach is not significantly slower than conventional techniques and can be easily incorporated into existing image gradient manipulation methods.

Acknowledgement This work was supported by the NUS Young Investigator Award, R-252-000-379-101, and the IT Consilience Creative Program of the Ministry of Knowledge Economy, Korea.

References

1. Bhat, P., Zitnick, C., Cohen, M., Curless, B.: Gradientshop: a gradient-domain optimization framework for image and video filtering. ACM Transactions on Graphics **29** (2010) 1–14
2. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. ACM Transactions on Graphics **21** (2002) 249–256
3. Jia, J., Sun, J., Tang, C., Shum, H.: Drag-and-drop pasting. ACM Transactions on Graphics **25** (2006) 631–637
4. Raskar, R., Ilie, A., Yu, J.: Image fusion for context enhancement and video surrealism. ACM SIGGRAPH Courses (2005)
5. McCann, J., Pollard, N.: Real-time gradient-domain painting. ACM Transactions on Graphics **27** (2008) 1–7
6. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Transactions on Graphics **22** (2003) 313–318
7. Zeng, Y., Chen, W., Peng, Q.: A novel variational image model: Towards a unified approach to image editing. Journal of Computer Science and Technology **21** (2006)
8. Krishnan, D., Fergus, R.: Dark flash photography. ACM Transactions on Graphics **28** (2009)
9. Yang, W., Zheng, J., Cai, J., Rahardja, S., Chen, C.: Natural and seamless image composition with color control. IEEE Transactions on Image Processing **18** (2009)
10. Xiao, X., Ma, L.: Gradient-preserving color transfer. In: Computer Graphics Forum. Volume 28. (2009) 1879–1886
11. Shibata, T., Iketani, A., Senda, S.: Image inpainting based on probabilistic structure estimation. In: Asian Conference on Computer Vision. (2010) 109–120
12. Omer, I., Werman, M.: Color lines: Image specific color representation. IEEE Computer Vision and Pattern Recognition (2004)
13. Duda, R., Hart, P., Stork, D.: Pattern classification. Volume 2. Wiley New York (2001)
14. Ren, X., Malik, J.: Learning a classification model for segmentation. IEEE International Conference on Computer Vision (2003)
15. Avriel, M.: Nonlinear programming: Analysis and methods. Dover Publishing (2003)
16. Levin, A., Fergus, R., Durand, F., Freeman, W.: Image and depth from a conventional camera with a coded aperture. ACM Transactions on Graphics **26** (2007)
17. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. Advances in Neural Information Processing Systems **22** (2009) 1–9
18. Zhang, X., Sim, T., Miao, X.: Enhancing photographs with near infra-red images. IEEE Computer Vision and Pattern Recognition (2008)
19. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence **13** (1991) 583–598
20. Vazquez, E., Baldrich, R., van de Weijer, J., Vanrell, M.: Describing reflectances for color segmentation robust to shadows, highlights, and textures. IEEE Transactions on Pattern Analysis and Machine Intelligence (2010)