Image Recoloring Based on Object Color Distributions

Mahmoud Afifi[1], Brian Price[2], Scott Cohen[2], and Michael S. Brown[1]

[1]York University, Toronto [2]Adobe Research, San Jose

Figure 1: *(A) An input image and its semantic segmentation (object mask) obtained by RefineNet [LMSR17]. (B) Recolored images produced by Photoshop's variation tool. (C) Recolored images from our method that considers the color distribution of objects in the image.*

Abstract

We present a method to perform automatic image recoloring based on the distribution of colors associated with objects present in an image. For example, when recoloring an image containing a sky object, our method incorporates the observation that objects of class 'sky' have a color distribution with three dominant modes for blue (daytime), yellow/red (dusk/dawn), and dark (nighttime). Our work leverages recent deep-learning methods that can perform reasonably accurate object-level segmentation. By using the images in datasets used to train deep-learning object segmentation methods, we are able to model the color distribution of each object class in the dataset. Given a new input image and its associated semantic segmentation (i.e., object mask), we perform color transfer to map the input image color histogram to a set of target color histograms that were constructed based on the learned color distribution of the objects in the image. We show that our framework is able to produce compelling color variations that are often more interesting and unique than results produced by existing methods.

1. Introduction and Related Work

Image recoloring aims to manipulate an image's RGB color values to give it a new appearance that conveys a different look and feel. Color manipulation for this purpose is achieved in different ways, such as color transfer (e.g., [FPC*16]), appearance transfer (e.g., [LRT*14]), and style transfer (e.g., [LPSB17]). Image editing software, such as Photoshop, provides tools for automatic image recoloring as a way to provide users with interesting variations on an input image. Fig. 1 shows a typical case of image recoloring, where an input image is manipulated automatically to produce several recolored variations.

The vast majority of existing recoloring methods require a target image that is specified by the user (e.g., [FPC*16, GEB16, LPSB17]). There are several methods that perform automatic transfer (e.g., [HZMH14, LSL*16]). To the best of our knowledge, these existing methods do not explicitly consider the objects present in the image in the recoloring process. Such object-level semantic information can be useful in guiding the recoloring effort to produce interesting and plausible variations on the input image. For example, if the input image has a *sky object*, the recolored image should exploit observations from the training data that a sky object can be blue, but not green. Moreover, if the original input image already has a sky object with a blue color appearance, we can use a dissimilar color associated with the sky object class (e.g., a reddish appearance) to provide more variety in the recolored results.

Contribution We propose a data-driven framework to automatically recolor an input image that incorporates information about the color distributions of objects present in the image. Specifically, we show how to model the color distributions of different object classes from thousands of images with labeled objects. Given a new input image and its object segmentation, we outline a procedure to produce a diverse set of recolored images. While this short paper represents work in progress, we show that our results produce compelling examples that provide more interesting variations than existing methods.

© 2019 The Author(s)
Eurographics Proceedings © 2019 The Eurographics Association.

DOI: 10.2312/egs.20191008

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

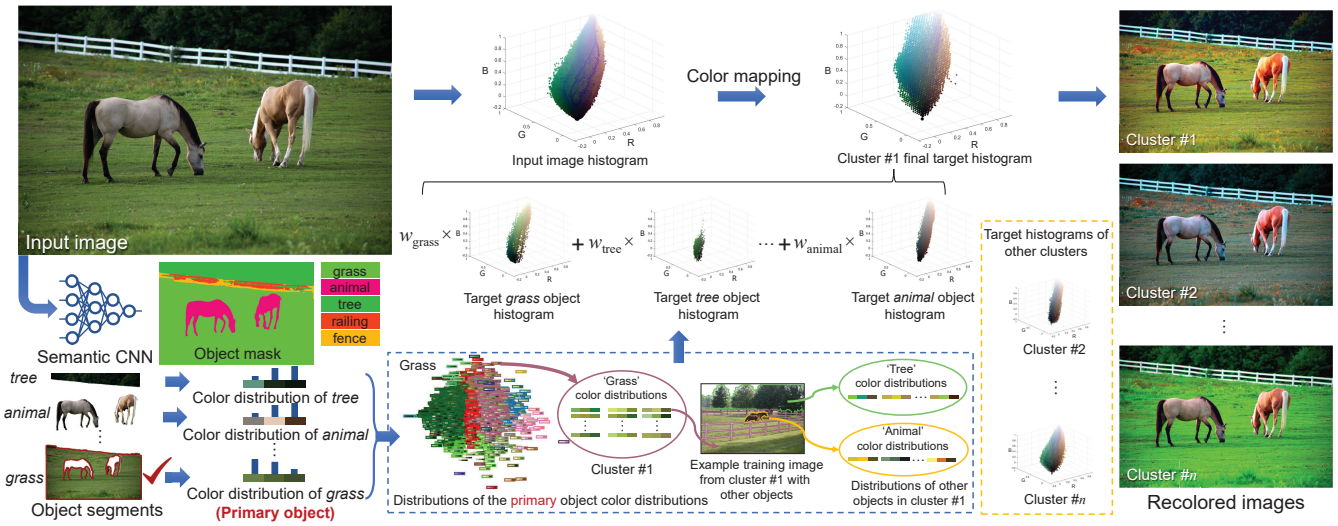

Figure 2: *Our image recoloring framework. We use a deep-learning method to obtain the input image's semantic segmentation as an object mask. We generate a color distribution for each object as a color palette and select one object in the image as the primary object. Next, we visit each cluster in the training images of the primary object class. For each cluster, we select an object instance with the most dissimilar colors to our input object's colors and add its colors to our target histogram. Within this same cluster, we search for the other (non-primary) objects found in the input image and use the most dissimilar instance in the target histogram. Lastly, color transfer is applied to map the input image to our generated target histogram. This is repeated for each cluster, producing several variations on the input image. The term w_{obj} refers to each object's pixel ratio in the input image.*

2. Our Method

Fig. 2 shows a diagram of our procedure. We start by describing the data preparation followed by the details of our method.

2.1. Training Data

Our method requires a large source of images with labeled object masks in order to build a color distribution for each class (type) of objects. To this end, we use the MIT Scene Parsing Benchmark (SPB) [ZZP*17], which contains 20,210 images with pixel masks for 150 different object classes.

For each object class b, we extract the RGB pixel values for all object instances for this class in the training images. The object class is represented as a set of color distributions $\mathcal{C}_b = \{\mathbf{c}_b^{(m)}\}_{m=1}^{M}$, where M is the number of training images containing an instance of the object of class b. An individual object instance's color distribution $\mathbf{c}_b^{(i)}$ is modeled as a color palette, with k colors, generated by the method proposed in [CFL*15]. The color palette representation also maintains the ratio of the number of pixels associated with each color in the palette.

We use the set $\mathcal{C}_b$ to build a "distribution of color distributions" (DoD) for each object class b. The DoD of an object b is generated by calculating the earth mover's distance (EMD) [RTG00] between each pair of color palettes $\mathbf{c}_b^{(i)}$ and $\{\mathbf{c}_b^{(j)}\}_{j=1}^{M}$ of object instances, where $i = [1,...,M]$ and $j \neq i$. Based on the EMDs between all color distributions in $\mathcal{C}_b$, we generate n clusters of the color distributions using an agglomerative (bottom-up) hierarchical clustering with Ward's minimum variance [WJ63].

By clustering all individual objects in the training images of object class b based on their color palettes, each object instance associated with a cluster shares a similar overall color appearance. Fig. 3 shows a visualization of this clustering procedure. Within each cluster, we maintain a list of all other objects that appeared in the training images. This latter point is important as it gives us a way to

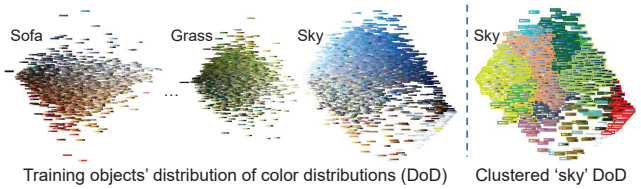

Figure 3: *For each object class in our training data, we extract each instance of an object of that class that appears in the training images and represent its color appearance as a distribution of k colors in the form of a color palette. We then construct a distribution of color distributions for each object class, termed a DoD (distribution of color distributions). The DoD is computed by clustering the color palettes into n clusters.*

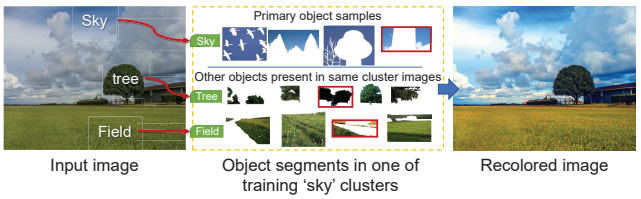

Figure 4: *For each cluster of the primary object, we search for the most dissimilar training samples (highlighted with red borders) of the input image's semantic objects.*

© 2019 The Author(s)
Eurographics Proceedings © 2019 The Eurographics Association.

Algorithm 1 Given an input image $\mathbf{I}$ and its object mask $\mathbf{M}$, we generate n recolored images $\{\mathbf{R}_i\}_{i=1}^{n}$.

procedure RECOLORING
  $p \leftarrow$ select the primary object
  for each object j in $\mathbf{M}$, **do**:
    $\mathbf{c}_{\text{obj}_j}^{\text{input}} \leftarrow$ get color palette (CP) of object j in $\mathbf{I}$
  end
for each training cluster i of primary object p, **do**:
  $\mathbf{H}^{(i)} \leftarrow \{\}$ ▷ Initialize empty target histogram
  for each object j in $\mathbf{M}$, **do**:
    if $\mathcal{C}_{\text{obj}_j}^{\text{clust}(i)}$ is emtpy, **then**
      $\mathbf{S}_{\text{obj}_j} \leftarrow \mathbf{I}_{\text{obj}_j}$ ▷ Original input object pixels
    else
      $\mathbf{d}_{\text{obj}_j} \leftarrow$ calculate EMD between $\mathbf{c}_{\text{obj}_j}^{\text{input}}$ and CPs in $\mathcal{C}_{\text{obj}_j}^{\text{clust}(i)}$
      $\mathbf{S}_{\text{obj}_j} \leftarrow$ retrieve training sample with max value in $\mathbf{d}_{\text{obj}_j}$
    end
    $\mathbf{H}_{\text{obj}_j}^{(i)} \leftarrow$ generate color histogram of $\mathbf{S}_{\text{obj}_j}$.
    $\mathbf{H}^{(i)} \leftarrow$ add $\mathbf{H}_{\text{obj}_j}^{(i)}$ to $\mathbf{H}^{(i)}$.
  end
  $\mathbf{R}_i \leftarrow$ color mapping $\left(\mathbf{H}^{\text{input}}, \mathbf{H}^{(i)}\right)$
end

find other objects that have appeared in the images for a particular object class b.

2.2. Recoloring Procedure

Given an input image $\mathbf{I}$, we compute its semantic object mask $\mathbf{M}$ using RefineNet [LMSR17]. Note that this mask can be noisy (i.e., RefineNet reports ~ 0.79 pixel-wise accuracy rate on the SPB dataset). The objects in $\mathbf{I}$ are ranked based on each object's pixel ratio in $\mathbf{I}$ (w_{obj}) and the ratio of the training samples with that object class (r_{obj}) in the training dataset. Both of these terms are normalized to range between 0 and 1. We select the primary object p as the object with the maximum score $w_{\text{obj}_p} + r_{\text{obj}_p}$.

For the primary object and all other objects in $\mathbf{I}$, we compute their color distributions (i.e., a color palette with associated pixel weights). We then visit each cluster of the primary object's class as described in Sec. 2.1. Within a cluster, we find the most dissimilar object instance based on the EMD of the primary object's color palette and each example in the cluster. The colors of this dissimilar object instance are added to the target histogram. Within the same cluster, this procedure is repeated for all other objects present in the image. Again, we seek the most dissimilar color palette to the input image's object within the cluster using the EMD metric. Fig. 4 provides an illustrative example. These dissimilar colors from the training data are added to the target histogram. If an object found in the input image does not exist in a cluster, we copy the input object's colors to the target histogram. Note that the construction of the target histogram was performed by a weighted summation based on each object's pixel ratio in $\mathbf{I}$. We have purposely chosen to use objects with the most dissimilar colors to provide notably different recolored images; however, we note the criteria for selecting target object instances can be adjusted to employ different strategies.

Once a target histogram has been constructed, we map the input histogram to the target histogram. This mapping produces the recolored result. In our experiments, we used the method by Pitie and Kokaram [PK07] to transfer the input image's color histogram to the target color histogram. Then, we scale any out-of-gamut pixels to fall in the range [0-255].

The procedure described above is repeated for each cluster in the primary object's class, producing n output images. Alg. 1 provides pseudocode for our procedure.

3. Results

In our experiments, the number of clusters used in the DoD is set to $n = 20$, the input and target color histogram bins are resized to resolution $32 \times 32 \times 32$, and our color palettes are fixed to have $k = 20$ colors. We show color palettes with three colors in Fig. 2 and Fig. 3 to simplify the visualization.

Fig. 5 shows comparisons between results of our method and three other methods. The first method [LRT*14] was proposed to transfer the appearance of outdoor images to different scene appearances specified by a set of attributes selected by the user. The second method [LSL*16] is an auto color/style transfer method which employs deep features extracted from a pre-trained convolutional neural network (CNN) on the ImageNet dataset [DDS*09] in order to produce content-aware color stylization. The third method [LPSB17] is a CNN-based method which requires a reference image in order to transfer its style to the input image. The results show that our method produces compelling results without requiring any user interaction or reference images. Additional qualitative results are shown in Fig. 6.

4. Concluding Remarks

We have proposed an automated data-driven method to generate recolored images. Our method leverages semantic object information of the input image and these objects' associated color distributions that have been modeled from thousands of training images. We demonstrate the effectiveness of the proposed method on several input images and compare our results with other strategies to produce color variations.

In line with the short paper theme, the proposed framework represents a work in progress. Our plan for future work is to find a more suitable means to perform color transfer based on the generated target histograms. Other directions include increasing the number of training images and the semantic objects and examining other methods to model the color distributions.

References

[CFL*15] CHANG H., FRIED O., LIU Y., DIVERDI S., FINKELSTEIN A.: Palette-based photo recoloring. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 139:1–139:11. 2

[DDS*09] DENG J., DONG W., SOCHER R., LI L.-J., LI K., FEI-FEI L.: Imagenet: A large-scale hierarchical image database. In *CVPR* (2009). 3

[FPC*16] FARIDUL H. S., POULI T., CHAMARET C., STAUDER J., REINHARD E., KUZOVKIN D., TRÉMEAU A.: Colour mapping: A review of recent methods, extensions and applications. In *Computer Graphics Forum* (2016), vol. 35, pp. 59–88. 1

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *CVPR* (2016). 1

© 2019 The Author(s)
Eurographics Proceedings © 2019 The Eurographics Association.

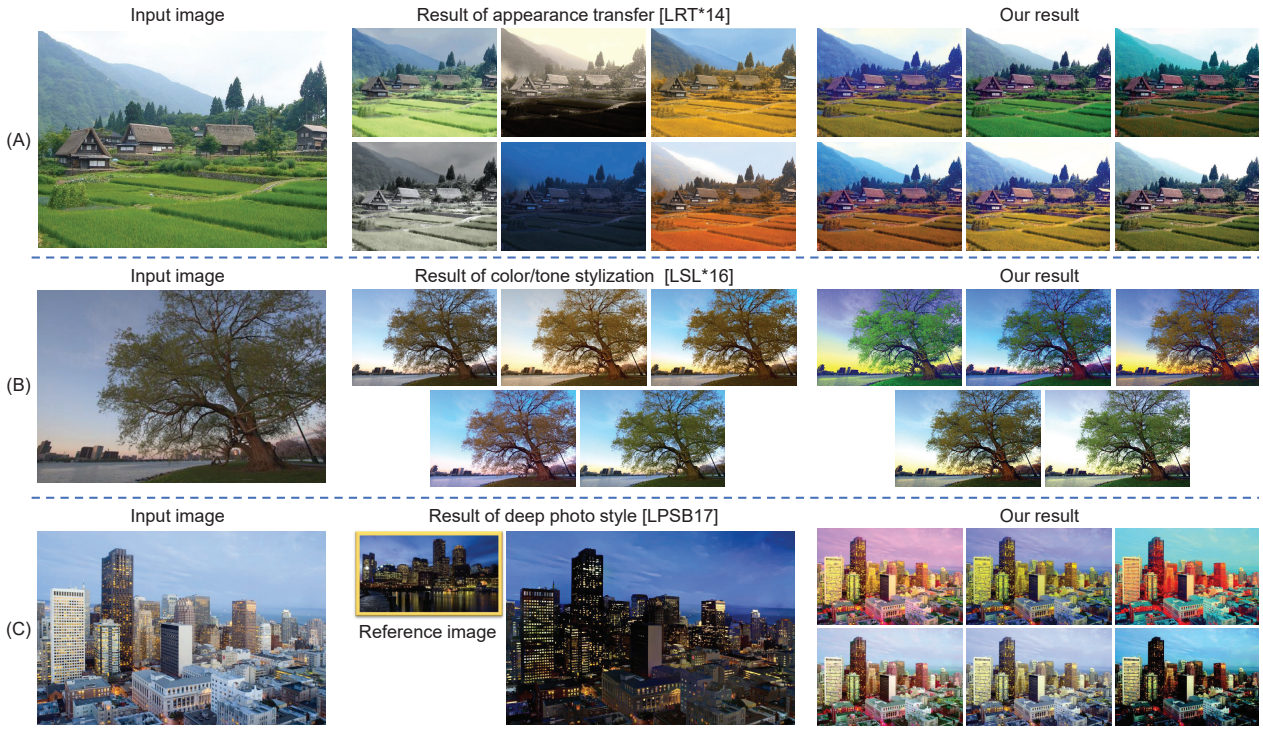

Figure 5: *Comparisons with existing style transfer methods. For each input image (first column), we show results of other methods (second column) and our results (third column). The other methods are: (A) appearance transfer [LRT*14], (B) auto content-aware color and tone stylization [LSL*16], and (C) a reference-based deep style transfer [LPSB17].*

Figure 6: *Qualitative results of the proposed method. (A) Input image. (B) Four examples of our recolored images.*

[HZMH14] Huang H.-Z., Zhang S.-H., Martin R., Hu S.-M.: Learning natural colors for image recoloring. In *Computer Graphics Forum* (2014), vol. 33, pp. 299–308. 1

[LMSR17] Lin G., Milan A., Shen C., Reid I. D.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR* (2017). 1, 3

[LPSB17] Luan F., Paris S., Shechtman E., Bala K.: Deep photo style transfer. In *CVPR* (2017). 1, 3, 4

[LRT*14] Laffont P.-Y., Ren Z., Tao X., Qian C., Hays J.: Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 149:1–149:11. 1, 3, 4

[LSL*16] Lee J.-Y., Sunkavalli K., Lin Z., Shen X., So Kweon I.: Automatic content-aware color and tone stylization. In *CVPR* (2016). 1, 3, 4

[PK07] Pitie F., Kokaram A.: The linear monge-kantorovitch linear colour mapping for example-based colour transfer. In *4th European Conference on Visual Media Production* (2007). 3

[RTG00] Rubner Y., Tomasi C., Guibas L. J.: The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision 40*, 2 (2000), 99–121. 2

[WJ63] Ward Jr J. H.: Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association 58*, 301 (1963), 236–244. 2

[ZZP*17] Zhou B., Zhao H., Puig X., Fidler S., Barriuso A., Torralba A.: Scene parsing through ade20k dataset. In *CVPR* (2017). 2

© 2019 The Author(s)
Eurographics Proceedings © 2019 The Eurographics Association.