

Illumination Robust Color Naming via Label Propagation

Yuanliu Liu, Zejian Yuan, Badong Chen, Jianru Xue, Nanning Zheng
 Institute of Artificial Intelligence and Robotics
 Xi'an Jiaotong University

liuyuanliu88@gmail.com, {yuan.ze.jian, chenbd, jrxue, nnzheng}@mail.xjtu.edu.cn

Abstract

Color composition is an important property for many computer vision tasks like image retrieval and object classification. In this paper we address the problem of inferring the color composition of the intrinsic reflectance of objects, where the shadows and highlights may change the observed color dramatically. We achieve this through color label propagation without recovering the intrinsic reflectance beforehand. Specifically, the color labels are propagated between regions sharing the same reflectance, and the direction of propagation is promoted to be from regions under full illumination and normal view angles to abnormal regions. We detect shadowed and highlighted regions as well as pairs of regions that have similar reflectance. A joint inference process is adopted to trim the inconsistent identities and connections. For evaluation we collect three datasets of images under noticeable highlights and shadows. Experimental results show that our model can effectively describe the color composition of real-world images.

1. Introduction

Color is a basic characteristic of visual objects. We use colors to distinguish one object from another in our daily life. Each human language has many words for describing the colors. According to [5], several color names are shared by most of the languages, which are called basic color terms. In English, they are black, blue, brown, grey, green, orange, pink, purple, red, white, and yellow. Color naming usually maps the observed colors to these basic color terms. Color names are widely used in computer vision, e.g., Google Image Search and object detection [22]. Recently, automatic color name labeling becomes important for online shopping [4] and online art galleries.

Visual objects look quite different under varying illuminations and view angles. See Fig. 1 for an example. Highlights make some parts of the strawberry appear white, while shadows make some regions nearly black. Conse-

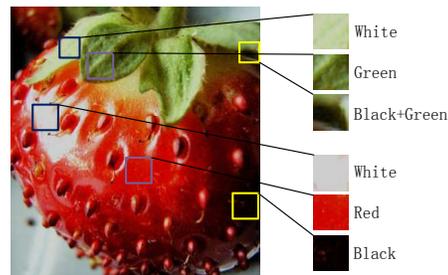


Figure 1: The regions appear very different from their original reflectance (in purple boxes) under highlights (in blue boxes) or shadows (in yellow boxes). If we know which regions have the same material (e.g., bottom three boxes), we can infer the color names of highlights or shadows regions by propagating the labels of normal regions to them. (Best viewed in color.)

quently, the color naming results will be totally different for different parts of the same surface. This will limit its usage in computer vision tasks like object detection, which desires consistent description of the same object. Traditional chip-based methods [3][14] generate flat color chips under controlled lighting environment. These color chips are totally unaware of the light condition and the view angle, so the annotation of them cannot be migrated to natural images directly.

Barrow and Tenenbaum proposed to decompose an image into a set of intrinsic images, each containing a single physical characteristic [2]. Typical intrinsic images include the reflectance, shading and specular reflection [7]. Especially, the reflectance describes how the light is reflected from the body of the object, which is invariant to the illumination and the view angle. In this paper we label the color of the intrinsic reflectance instead of the raw image under shadows and highlights.

Given a natural image, our goal is to tell the color composition of the intrinsic reflectance for each image region. However, decomposing a single image into several intrinsic components is an ill-posed problem. Instead of using some prior knowledge to get the reflectance first

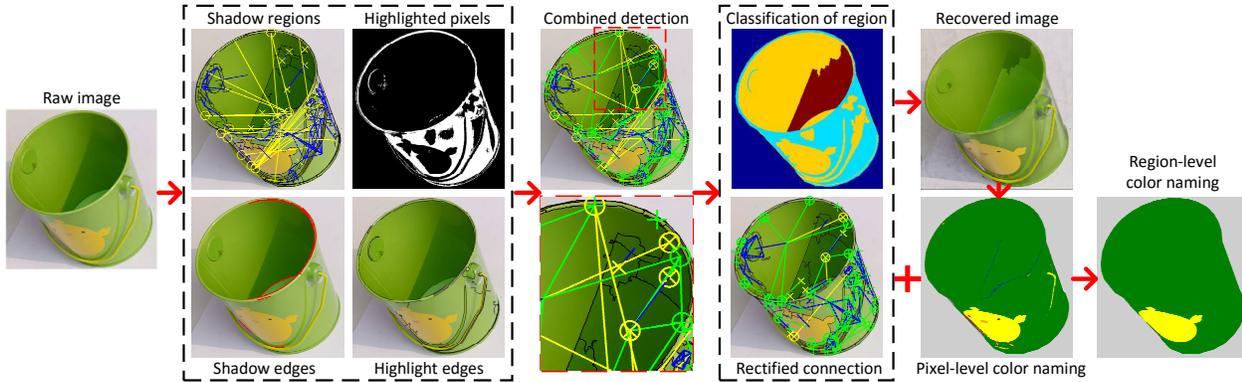


Figure 2: The flow chart of our method. For the detection results, the highlighted regions are marked with green cross, and the shadowed regions are indicated by yellow “x”. Highlight/diffuse pairs are connected by green lines, with a green circle on highlight. Shadow/non-shadow pairs are joined by yellow lines, with a yellow circle on shadow. The similar pairs are coupled by blue lines. For the classification of regions, normal regions are covered by yellow, highlighted regions are marked by cyan, and shadows are shown in red.

[20], we propose to propagate the color labels from normal regions to those shadowed or highlighted regions with the same albedo. Here the normal regions refer to those areas of a surface that are illuminated by the natural light and viewed at a angle that is not close to the reflection angle. The observed colors from the normal regions are qualified representatives of the reflectance, which are suitable sources for label propagation.

The flow chart of our method is shown in Fig. 2. We begin with segmenting the foreground of the image by MeanShift [6]. Then we detect the shadows and highlights, and identify the pairwise relations between regions sharing the same reflectance (Sec. 2). The shadow and highlight detection results are often noisy and inconsistent. We use a joint inference to combine the local detection results into a consistent labeling of the status of the regions, and rectify the pairwise relations accordingly (Sec. 3). To ease the color propagation, we recover the color and intensity of the shadowed and highlighted regions roughly (Sec. 4). We get the region-level color compositions from the pixel-level color naming results supplied by off-the-shelf methods [3]. Finally we build a MRF model for label propagation (Sec. 5). While MRF models are undirected, we desire a one-way propagation of color labels, i.e., from normal regions to the shadowed or highlighted ones but not reversely. We achieve this by setting the normal regions to be the anchor nodes and keeping the labels of the other regions adaptive.

The main contributions of this paper are: (1) We build a MRF model to propagate the color labels from normal regions to shadowed or highlighted regions, which can robustly estimate the color names of the intrinsic reflectance of natural images; (2) we use a joint inference process to make the shadow and highlight detection results consistent, which gives reliable paths and directions for label propagation; and (3) we collect three datasets of natural

images under shadows and highlights, and annotate them with region-level intrinsic color labels for evaluation.

1.1. Related Work

Serra et al. [20] proposed to infer the pixel-wise color names of the intrinsic image from label propagation. Their method is based on the segmentation results of the Ridge Analysis of the color Distributions (RAD) [25], which is robust to shadows and highlights. They built a MRF model to encourage the pixels connected by a ridge to have same color name. But their model did not specify the direction of the propagation. It may fail when the shadow and highlight cover a bigger portion of the surface than the normal regions. In contrast, our model is aware of which regions are shadowed or highlighted, and the labels will be propagated in the desired way.

Liu et al. [10] labeled the color composition of visual objects. They inferred the region-level and image-level color distribution from pixel-level color naming results, taking the human preference and color assimilation effect into consideration. Van de Weijer et al. [24] learned color names from natural images. They used a PLSA model to capture the color composition of the image and the color name distribution over pixels. The learned color names show good robustness to shadows and shadings since the training data contain such variations. But these variations made the color name distribution flat, i.e., there are several reasonable explanations for one observed pixel. Mojsilović [14] presented a computational model for color naming. She built a syntax for multi-level color description. For extracting the color composition of an image, she computed the color name histogram of the pixels in uniform or textured regions. This work addressed the issues of color constancy, image smoothing and segmentation, but not the shadows or highlights.

Shadow detection [1, 29, 8] and highlight removal [23, 13, 27, 11] have already been widely studied. Their results are the basis of identifying shadowed and highlighted regions as well as constructing the connections between regions. Especially, Guo et al. detect unary shadowed regions as well as pairs of shadow/non-shadow regions [8], and determine the status of regions jointly from the detection results. We extend their inference of shadows to incorporate highlights inside.

2. Shadows and highlights

In this section, we calculate the probability of being normal, shadowed or highlighted for each region. We also extract the shadow/non-shadow pairs E_{sn} , the highlight/diffuse pairs E_{hd} and the similar pairs E_{sim} . A region pair is regarded as a shadow/non-shadow pair if they have the same reflectance but only the former region is blocked from the direct illumination. A highlight/diffuse pair is composed of a highlighted region and a diffuse region that have the same diffuse reflectance. Two nearby regions have similar appearance can form a similar pair. They are the basis to determine the path and direction of label propagation in Sec. 5.

2.1. Shadow detection

The shadows cause some regular changes to the color and texture, and we call them “shadow jump”. Intuitively, the shadowed regions get lower intensity and weaker texture than the lit regions [29]. In the penumbra area the illumination changes continuously, which results in a soft shadow edge. We use these properties to detect shadows and shadow/non-shadow pairs.

Region-based shadow detection. We use the framework of Guo et al. [8] to detect shadowed regions and shadow/non-shadow pairs of regions. Especially, we propose a new shadow invariant feature, which measures the color/texture similarity robustly under shadow jump. For the color/texture histograms of a pair of regions, we calculate their Earth Mover’s Distance (EMD) [17]. The shadow jump patterns are embedded into the ground distance between histogram bins. First we count the frequency of one color/texton bin being transported to another colors/texton bin by shadows in a training set. We assume that the transportation with higher frequency is more probably caused by a shadow jump, so we assign a smaller distance to it. In implementation, the distance is set to be inversely proportional to the frequency of the transportation. The resulted EMD captures the patterns of shadow jump on both color and texture. It is also robust to small variations of the image. Since the shadow jump is asymmetric, it can separate shadow/non-shadow pairs from similar pairs as well as non-shadow/shadow pairs. We use Pele and Werman’s version of EMD with non-symmetric ground

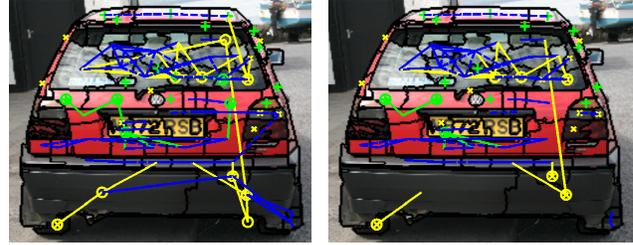


Figure 3: The shadow and highlight detection results. (a) The original detection results. (b) The joint inference result. See Fig. 2 for the meaning of the marks and lines.

distance [15]. The texture distance are very useful for achromatic surfaces. We use texton histograms [12] to represent the texture.

We translate the output scores S_{sn} of the shadow/non-shadow pair detectors into probabilities by sigmoid functions $p_{sn} = 1/(1 + \exp(-\gamma_{sn} * S_{sn} + \eta_{sn}))$ [16], where γ_{sn} and η_{sn} are parameters. Similarly we can get the probability p_s of being shadowed with parameters γ_s and η_s .

Edge-based shadow detection. We use the method of Lalonde et al. [9] to detect shadow edges. An oriented gaussian derivative filter is convolved with the image area near the region boundaries. The filter responses are fed into a decision tree classifier to decide if the boundary is a shadow edge. Since the shadow edges are not closed in most cases, we still do not know which regions are shadowed. Here we label the regions on the darker side of the edges as being shadowed with a probability of $p_s = 1$. We further extend the shadow label to similar regions nearby. In addition, if two regions with similar chromaticity are separated by a shadow edge, they probably come from the same surface. Therefore the probability of a shadow/non-shadow pair p_{sn} can be determined by the similarity of chromaticity.

The outputs of the region-based method and the edge-based method are fused by taking the maximum of corresponding values. Note that, most highlighted regions will be classified as non-shadow regions, and they may also appear in the shadow/non-shadow pairs.

2.2. Highlight Detection

In this section, we measure the probability p_{hd} of two adjacent regions being a highlight/diffuse pair. We also calculate the probability of highlights p_h for chromatic and achromatic regions separately.

Highlight/diffuse pairs. According to the Dichromatic Reflection Model [21], the reflectance R is calculated by

$$\mathbf{R}(p, \gamma) = w_d(p)\mathbf{R}_d(p) + w_s(p, \gamma)\mathbf{R}_s \quad (1)$$

where $R_d(p)$ is the chromaticity, i.e., the normalized RGB, of diffuse body reflectance at the pixel p . R_s is the

chromaticity of the specular interface reflectance, which is assumed to be always white ($[1/3, 1/3, 1/3]$). $w_d(p)$ and $w_s(p, \gamma)$ are the intensities of diffuse reflectance and specular reflectance, respectively. γ is the viewing angle. The diffuse intensity w_d does not vary with the viewing angle. In contrast, the specular intensity w_s is very high in certain viewing angles, where the highlights occur. It should be mentioned that there are still specular reflections in other viewing angles, although their intensities are much lower.

A highlight/diffuse pair is formed by two adjacent regions that have the same hue but different saturation. The specular reflections are achromatic, so the highlights will retain the hue of the body reflectance but reduce the saturation [23]. Based on these properties, we measure the probability p_{hd} of a highlight/diffuse pair by the similarity of hue histograms and the difference of saturation. For achromatic regions, the hue is unstable. Instead we measure the strength of the boundary between the highlighted regions and their diffuse neighbors. The weaker the boundary is, the higher the probability of being a highlight/diffuse pair will be. The underlying assumption is that the reflection angle changes smoothly over the surface, so does the intensity of specular reflection.

Chromatic highlights. To remove the specular reflection on chromatic surfaces, we use the method of [23]. It iteratively reduces the achromatic component w_s of each pixel until all the surfaces become specular-free. That is, the saturation of the highlighted pixels gets close to that of the diffuse pixels. See [23] for details. We regard a pixel in the original image as highlighted if its saturation drops more than a threshold T_h after highlight removal.

Achromatic highlights. We use the method of [13] to detect the achromatic highlights, including those near-white highlights on chromatic surfaces. There are two assumptions: (1) Highlights on a smooth shining surface tend to have a profile of a sharp spike overlaying on a smooth line, and (2) on each side of the spike, the intensity profile is close to a straight line. Accordingly, when we use different thresholds T for the intensity of highlights, the perimeters of the resulted highlight regions will approximate a straight line. In contrast, the perimeters of a bright diffuse region under different thresholds will be roughly piecewise constant. Since the reflectance ratio varies across surfaces, the threshold T should be different. We first find out bright achromatic regions using an intensity threshold and a saturation threshold. Then we find a threshold T for each region.

After detecting both achromatic and chromatic highlights, we count the proportion of the highlighted pixels in each region, which is taken to be the probability p_h of being a highlighted region.

2.3. Similar pairs

We use the detectors of Guo et al. to find out similar pairs [8]. Similar to the shadow/non-shadow pairs in Sec 2.1, the output scores S_{sim} are translated into probabilities p_{sim} by a sigmoid function with parameters γ_{sim} and η_{sim} .

3. Joint Inference of Shadows and Highlights

The detection results of shadows and highlights are often noisy. In addition, detections based on different features may conflict. For example, in the left part of Fig. 3 two shadowed regions on the car rear form a shadow/non-shadow pair since they have the same color but different intensity. Also, a highlighted region on the right side of the car roof was connected to a normal region of the window since they are both nearly achromatic. Here we compromise the detection results in a MRF model.

We label the image regions $X = \{x_i\}_{i=1}^N$ with labels $Y = \{y_i\}_{i=1}^N$, where each y_i could be shadowed (-1), highlighted (1), or normal (0). Here we assume that a region cannot be both shadowed and highlighted at the same time. Note that the shadowed regions may also contain weak specular reflection that are negligible for the color naming task. We further enforce the consistency between pairwise relationship and labels of individual regions, and the rules are: (1) A similar pair of regions should have the same label; (2) a shadow/non-shadow pair is formed by a shadowed region and a normal or highlighted region in the right order; and (3) a highlight/diffuse pair begins with a highlighted region and ends in a normal or highlighted one.

The score function is defined as follows:

$$S(X, Y) = \sum_i \Phi^u(x_i, y_i) + \sum_{(i,j)} \Phi_{i,j}^b(y_i, y_j) \quad (2)$$

Here the unary term is defined as follows:

$$\begin{aligned} \Phi^u(x_i, -1) &= p_s(i) \\ \Phi^u(x_i, 1) &= p_h(i) \\ \Phi^u(x_i, 0) &= 1 - \max(p_s(i), p_h(i)) \end{aligned} \quad (3)$$

where p_s and p_h are the probabilities of being shadowed and highlighted, respectively. The binary term is designed based on the rules of pairwise relationships:

$$\Phi_{i,j}^b(y_i, y_j) = \begin{cases} p_{sim}(i, j) & \text{if } (i, j) \in E_{sim}, y_i = y_j \\ p_{sn}(i, j) & \text{if } (i, j) \in E_{sn}, y_i = -1, y_j \neq -1 \\ p_{hd}(i, j) & \text{if } (i, j) \in E_{hd}, y_i = 1, y_j \neq 1 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where p_{sim} , p_{sn} and p_{hd} are the probabilities of being a similar pair, a shadow/non-shadow pair, and a highlight/diffuse pair, respectively. All these probabilities are got from shadow and highlight detection in Sec. 2.

The score function in Eq. 2 can be converted to an energy function by simply adding a negative sign to all the terms. Then we use the method of Alpha-Beta swap implemented by the UGM software [19] to find out the optimal label Y^* . After that, we prune the region pairs that are inconsistent to the new labels. The remaining shadow/non-shadow pairs, the highlight/diffuse pairs and the similar pairs are denoted by E_{sn}^* , E_{hd}^* , and E_{sim}^* , respectively. The new labels and connections are more reliable than the initial detection in Sec. 2. See Fig. 3 for an example. The optimal label sequence and edge sets are used for color propagation in Sec. 5.

4. Color and intensity recovery

The connections between regions are sparse after we prune the inconsistent connections in Sec. 3. Fig. 4 gives an example. In this image there is only a very small portion of the object exhibiting canonical lighting. If we apply the label propagation described in Eq. 5 shortly, it might fail because the labels of normal regions cannot be sufficiently propagated to the highlighted and shadowed regions. This problem has been well-addressed in the domain of constrained segmentation [28]. To alleviate the problem we roughly recover the colors according to the shadow and highlight detection results, as follows.

For chromatic surfaces, the highlight removal process in Sec. 2.2 removes the specular reflection to generate diffuse surfaces. If there are some highly saturated diffuse pixels, which happens a lot in natural images, the achromatic component of adjacent pixels will be totally eliminated after highlight removal (See Fig. 5 for an example). That is, the intensity will be reduced until one color channel nearly drops to 0. It will appear darker than those normal regions. Here we propose to transfer the specular reflection from the normal regions to the highlighted region when they form a highlight/diffuse pair. In implementation, we take the median value if there are more than one normal regions connected to the highlighted region. Under strong specular reflection, some channels of the camera sensors may saturate. It will decrease the differences between channels and thus change the hue. Fortunately, most highlighted regions still have several unsaturated pixels on their smooth borders. We propagate the color from the unsaturated pixels to the saturated ones. The totally saturated regions cannot be handled here, since nearly all the color information has lost. We left these regions to label propagation in Sec. 5.

The ratio of intensity r between shadow and non-shadow regions are needed for shadow removal [8]. We get an estimate of r from the ratio of the average intensity between each pair of shadow/non-shadow regions. The median of these ratios is taken to be the final estimation \tilde{r} . The shadows are removed by dividing the intensity by \tilde{r} .

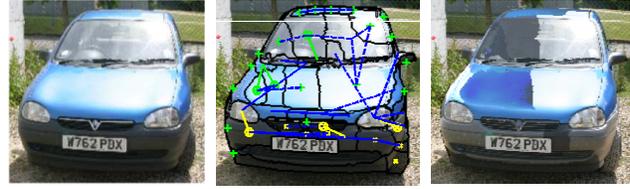


Figure 4: An example of color and intensity recovery. **Left:** The original image. **Middle:** The shadow and highlight detection results, which are sparse. See Fig. 2 for the meaning of the marks and lines. **Right:** The image after color and intensity recovery.



Figure 5: Color and Intensity recovery for highlighted regions. **Left:** The original image. The top left corner of Region 1 turns achromatic due to highlight. Region 2 is a highlighted region. Region 3 is a normal region. Region 1 and Region 3 form a highlight/diffuse pair. Region 2 and Region 3 form another highlight/diffuse pair. **Middle:** The specular-free image after highlight removal. All three regions are darker than the original image. Moreover, the achromatic pixels in Region 1 become black. **Right:** The recovered image.

Although the recovered image in this section is not exactly the intrinsic reflectance, it gets much closer to the real color of the surfaces. Using the recovered image instead of the raw image to calculate the region-level color distribution (Sec. 5) will ease the color label propagation.

5. Color Labeling

In this section we infer the region-level color compositions through label propagation. Formally, the input is a set of image regions $X = \{x_i\}_{i=1}^N$, and the output is their color distribution $Z = \{z_{i,c}\}_{(i,c)=(1,1)}^{(N,K)}$. Here K is the number of basic color terms. A region x_i is specified by its color distribution $f(x_i)$ got from averaging the pixel-level color distribution. We use the method of Parametric Fuzzy Sets (PFS) [3] to get the pixel-level color naming results, i.e., the probability of each pixel belonging to the 11 basic color terms. To make the label space tractable, we quantize $z_{i,c}$ to be within $\{0, 0.25, 0.5, 0.75, 1\}$. We further limit the number of colors in a single region to be no more than 2, since people tend to use a small number of colors to label a uniform region¹. There are 176 feasible states for the region-level color label z_i . Among them, there are 11 states that only one color occurs in the region. The number of combinations of two colors is $3 \cdot C_{11}^2 = 165$, where 3 is the

¹Some researchers [20] also use 3 colors to describe a region, but we found that it will be hard to estimate the proportion of them consistently by human annotators.

number of choices among $\{0.25, 0.5, 0.75\}$.

Illumination robust color naming requires that all the regions of the same reflectance should be labeled with the same color names, no matter how different the illumination and the specular reflection are from place to place. We consider 3 types of region pairs: the highlight/diffuse pairs in E_{sn}^* , the shadow/non-shadow pairs in E_{hn}^* , and the similar pairs in E_{sim}^* (Sec. 3). All these pairs of regions are composed of regions with the same reflectance, so we can propagate the labels between them. We combine different types of region pairs into $E = E_{sn}^* \cup E_{hn}^* \cup E_{sim}^*$.

We build a MRF model for color name propagation. The energy of labeling regions X with color distribution Z is defined as follows:

$$P(X, Z; Y^*) = \sum_i \Psi^u(x_i, z_i; y_i^*) + \sum_{(i,j) \in E} \Psi_{i,j}^b(z_i, z_j) \quad (5)$$

with

$$\begin{aligned} \Psi^u(x_i, z_i; y_i^*) &= \alpha(y_i^*)d(f(x_i), z_i) \\ \Psi_{i,j}^b(z_i, z_j) &= \beta(i, j)d(z_i, z_j) \end{aligned}$$

where $\alpha(y_i^*)$ is the weight of each region decided by its type. We assign much smaller weights to the highlighted or shadowed regions than the normal ones. Therefore the highlighted regions and the shadowed regions are more adaptable, which means the binary terms of MRF will change their labels to those of the normal regions more often than in the reverse way. In addition, we set the weights of achromatic highlighted regions to be the lowest, since the observation on them are the most unreliable. $\beta(i, j)$ is the strength of the connection between node i and j , which is set to be the probability of the pairwise connection $p_t(i, j)$ of type t . We adopt the histogram intersection to measure the similarity of different labels, so $d(z_i, z_j) = 1 - \sum_c \min(z_{i,c}, z_{j,c})$. We use Alpha-Beta swap of UGM [19] to minimize the energy in Eq. 5.

6. Experiments

In our experiments, we evaluate the accuracy of region-level color naming performance. We also analyze the impact of joint inference of shadows and highlights, and the color and intensity recovery.

6.1. Datasets

To our best knowledge, there is no public dataset available for region-level color naming. So we built 3 datasets for evaluation. The first one is a set of 30 car images. The original images are from the MSRC Object Categorization v2 dataset [26]. We add the region-level color annotation using the Labelme toolbox [18]. For each surface with a uniform color, we draw its boundary and add a label of the proportions of the basic color terms. To facilitate the annotation, the number of colors on a single region is

limited to be no more than 2. The color proportion is chosen from 5 discrete values $\{0, 0.25, 0.50, 0.75, 1\}$. To make the label consistent, all the images are annotated by one author. These images contains intensive highlights due to glass, plastic and metal surfaces. Shadows are also common due to self-occlusion.

To test the robustness on different object class and materials, we built another dataset called Multiclass Color Naming dataset (MCN). It contains 17 object categories e.g., horse, jewelry and shoes. It has 20 images in total. Most of the images are collected from the internet, and the others are from the dataset of [8]. This dataset covers many materials including glass, stone, wood, fruit, cloth, fur and so on. In addition, the strength of the highlights and shadows vary a lot across images.

We build a larger dataset of small objects. It contains 300 images taken by the Canon 7D camera in raw mode. Each color channel is stored in 16 bits when we export the images in TIF format. The images are cropped and resized to be 1M pixels each. The materials of the objects range from paper, plastic, metal, leather, wood, leaf and so on. We put the objects on the table and arrange them in different layout. For some images we intentionally block the sunshine partially to generate shadows. We then capture the images from different view angles.

6.2. Evaluation

We evaluate the region-level color naming results by their histogram intersection with the ground-truth:

$$S_r = \sum_i a_i \sum_c \min(z_{i,c}^{pred}, z_{i,c}^{gt}) \quad (6)$$

where $z_{i,c}^{pred}$ and $z_{i,c}^{gt}$ are the prediction and ground-truth annotation of the proportion of color c on region i , respectively. The weight a_i is the area of region i . The evaluation is based on the ground-truth segmentation. Since the predicted color names are labeled on regions from the Meanshift segmentation, we need to reorganize them at first. We distribute the region-level prediction to the pixel-level, and calculate the color name histograms of the regions from the ground-truth segmentation.

To reduce the influence of segmentation, we also evaluate the pixel-level performance S_p . Both the ground-truth annotation and the predicted labels are distributed to the pixels. We calculate the pixel-level histogram intersection and then average them over the pixels of the whole object.

We set the weights α in Eq. 5 of normal regions, shadows, chromatic highlights and achromatic highlights to be 1, 0.2, 0.5 and 0.1, respectively. The parameters of the sigmoid functions in Sec. 2 are set to be: $\gamma_{ns} = 5, \eta_{ns} = 3, \gamma_{sim} = 5, \eta_{sim} = 6, \gamma_s = 5, \eta_s = 2$. The threshold of chromatic highlight is set to be $T_h = 0.2$.

Datasets	Car Dataset						MCN Dataset	
	\bar{S}_r^1	\bar{S}_r^2	\bar{S}_r	\bar{S}_p^1	\bar{S}_p^2	\bar{S}_p	\bar{S}_r	\bar{S}_p
RAD [20]	22.3	23.8	22.5	22.0	23.1	22.2	19.4	19.0
PLSA [24]	37.6	29.0	36.2	37.1	28.7	35.7	41.7	40.9
PFS [3]	60.5	50.5	58.8	55.1	46.1	53.6	54.0	50.1
Ours(W/O JID)	62.2	48.4	60.0	60.0	46.5	57.7	53.8	51.4
Ours(W/O CIR)	62.8	49.0	60.5	60.8	47.1	58.6	54.6	52.2
Ours	63.8	47.4	61.1	61.4	45.8	58.8	56.5	55.4

Table 1: The average percentage of corrected labeled regions \bar{S}_r and pixels \bar{S}_p over the images of the car and the MCN dataset. The superscript 1 indicates group $G1$ and 2 for $G2$. Ours W/O CIR is the version without color and intensity recovery. Ours W/O JID is the version without both CIR and joint inference of shadow and highlight detection results.

Table 2: The results on the small object dataset. Ours-PFSfeat refers to our model using PFS for the pixel-level color naming, while Ours-PLSAfeat uses PLSA as the baseline.

Algorithm	\bar{S}_r	\bar{S}_p
PLSA [24]	52.6	52.1
PFS [3]	68.3	66.6
Ours-PLSAfeat	61.1	60.7
Ours-PFSfeat	68.7	67.9

6.3. Results

We divide the Car dataset into 2 groups: images with only one car ($G1$) and images with multiple cars ($G2$). The average \bar{S}_r and \bar{S}_p for both groups are given in the left part of Table 1. The chip-based color naming method of [3] is taken as the baseline. It supplies the pixel-level color distribution, which is the source of our region-level color distribution. It reflects the performance of color naming without any color label propagation. We compare our algorithm to the RAD-based method [20], which also adopts color label propagation. We also compare to the PLSA model [24], which learned pixel-level color distribution from weakly labeled real images. For group $G1$, our method achieves an improvement of 5.5% for \bar{S}_r and 11.4% for \bar{S}_p over the baseline of PFS [3]. In detail, the joint inference of detection results contribute 18% and 13% for the improvement of \bar{S}_r and \bar{S}_p , respectively. The numbers for color and intensity recovery are 30% and 10%. For group $G2$, our method without color and intensity recovery gets the best performance.

The results on the MCN dataset are shown in the right part of Table 1. Our method outperforms the other methods in both \bar{S}_r and \bar{S}_p . It suggests that our method is effective for various materials and lighting conditions.

The results on the small objects dataset are shown in Tab. 2. Our method still performs best. Since the images in this dataset have high quality, all the methods achieved much better results than on the other two datasets.

Figs. 6 and 7 give some examples. The chip-based method PFS [3] suffers the shadows and highlights a lot. For example, some part of the red car under highlights are label with pink or even white (in the second and third row

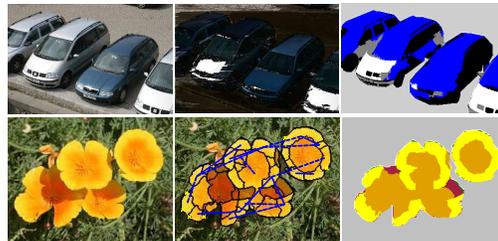


Figure 8: Some failed cases.

of Fig. 6), and the orange wall under shadows is labeled with brown (in the fifth row of Fig. 6). PLSA learned the color names from natural images [24], so the color names cover colors generated from various illuminations and view angles. Hence this method got slightly better robustness to the shadows and highlights. For example, the red surfaces with a few specular reflection are correctly labeled (in the second and third row of Fig. 6), but it fails when the specular reflection gets stronger. Our model explicitly accounts for the shadows and highlights, so the influence of these factors are greatly reduced. Although the blue car in the fourth row of Fig. 6 is under severe highlight, our model still labeled the color names correctly. In the fifth row of Fig. 6, more than half of the wall is shadowed, and the label propagation recovered most regions.

RAD also propagates labels between the pixels sharing the same reflectance, but they do not specify the direction of the propagation [20]. Therefore the results may get worse after label propagation in the wrong directions. Moreover, the RAD segmentation do not consider the spatial relation between pixels, so they may connect the highlighted and shadowed pixels to the bright and dark achromatic surfaces, respectively. Therefore the labels may be propagated from the black/white regions or shadowed/highlighted regions to the normal regions if the former ones are much larger than the later ones. As a result, many regions are mistaken to be black or white by RAD.

Fig. 8 shows some failed cases. In the top row of Fig. 8, the inter-reflection between multiple cars changes the local illumination. In this case the highlight removal will make mistakes since the local chromaticity variations of illumination are not captured by the reflection model in Eq. 1. This also explains why most methods get worse performance on group $G2$ (in which images contain more than one car) than on $G1$ of the Car dataset, and the version without color and intensity recovery achieves better results than our full model. In the bottom row of Fig. 8, some shadowed regions are not identified, so the label propagation does not work for them. Another limitation is that most segmentation algorithms require a minimal size of regions, so small groups of pixels will be merged with their neighbors, e.g., the numbers on the licence plate.

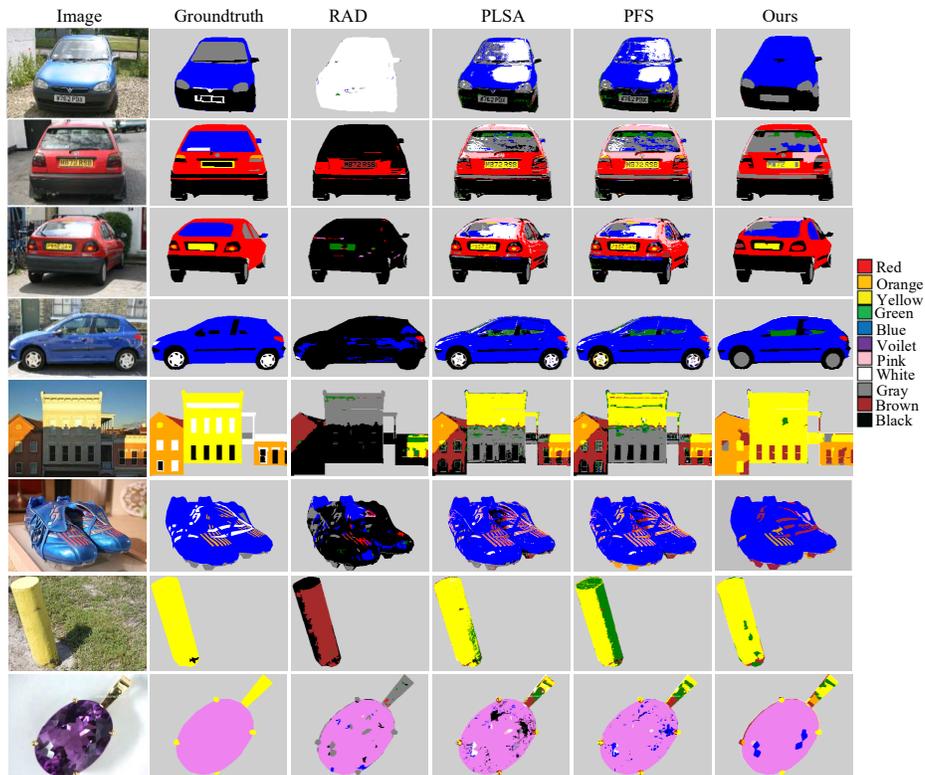


Figure 6: Typical results on the car dataset and the MCN dataset. Each region/pixel is colored with the dominant color name that gets the highest proportion. The background pixels are set to be light gray according to the foreground/background annotation of the datasets.

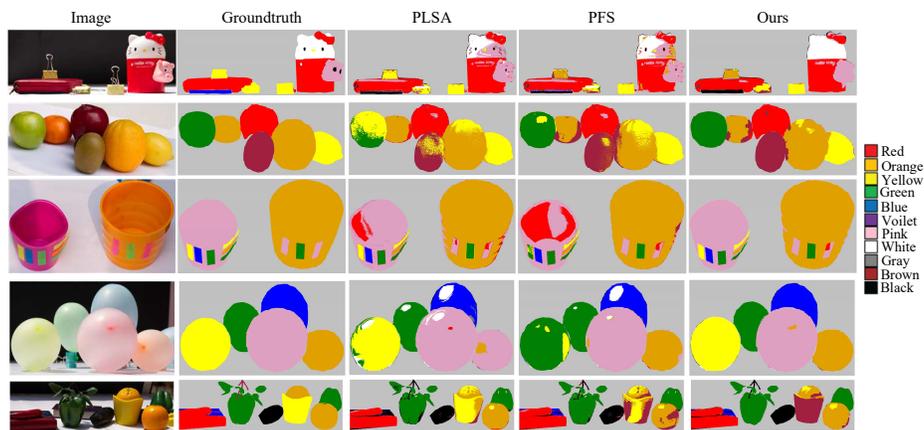


Figure 7: Representative results on the dataset of small objects. See Fig. 6 for the settings of the colors.

7. Conclusion

We built a MRF model to propagate the color labels from normal regions to shadowed or highlighted regions. The resulted color names are robust to illumination variation and specular reflection. Our method relies on shadow and highlight detection, which may fail sometimes. We use a joint inference process to infer the optimal status of the regions and trim the pairwise connections.

The traditional reflection model in Eq. 1 and the highlight detection methods based on it cannot handle the inter-reflections. We leave it for future work.

Acknowledgement. This work was supported by National Basic Research Program of China (No.2015CB351703), National Natural Science Foundation of China (No.61573280, No.61231018), and 111 Project (No.B13043).

References

- [1] K. Barnard and G. Finlayson. Shadow identification using colour ratios. In *Colour Imaging Conference: Colour Science, Systems and Appl*, pages 97–101, 2000. 3
- [2] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer Vision Systems*, pages 3–26, 1978. 1
- [3] R. Benavente, M. Vanrell, and R. Baldrich. Parametric fuzzy sets for automatic color naming. *J. Opt. Soc. Am.*, 25(10), 2008. 1, 2, 5, 7
- [4] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. *ECCV*, 2010. 1
- [5] B. Berlin and P. Kay. *Basic color terms: their universality and evolution*. University of California Press, 1969. 1
- [6] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002. 2
- [7] R. Grosse, M. Johnson, E. Adelson, and W. Freeman. Ground-truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, pages 2335–2342, 2009. 1
- [8] R. Guo, Q. Dai, and D. Hoiem. Single-image shadow detection and removal using paired regions. In *CVPR*, pages 2033–2040, 2011. 3, 4, 5, 6
- [9] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *ECCV*, pages 322–335, 2010. 3
- [10] Y. Liu, Y. Liang, Z. Yuan, and N. Zheng. Learning to describe color composition of visual objects. In *ICPR*, pages 3337–3340, 2012. 2
- [11] Y. Liu, Z. Yuan, N. Zheng, and Y. Wu. Saturation-preserving specular reflection separation. In *CVPR*, 2015. 3
- [12] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004. 3
- [13] K. McHenry, J. Ponce, and D. Forsyth. Finding glass. In *CVPR*, volume 2, pages 973–979, 2005. 3, 4
- [14] A. Mojsilovic. A computational model for color naming and describing color composition of images. *IEEE TIP*, 14(5), 2005. 1, 2
- [15] O. Pele and M. Werman. Fast and robust earth mover’s distances. In *ICCV*, 2009. 3
- [16] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances In Large Margin Classifiers*, pages 61–74. 3
- [17] Y. Rubner, C. Tomasi, and L. Guibas. The earth mover’s distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121, 2000. 3
- [18] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *IJCV*, 77:157–173, 2008. 6
- [19] M. Schmidt. <http://www.di.ens.fr/~mschmidt/software/ugm.html>. 5, 6
- [20] M. Serra, O. Penacchio, R. Benavente, and M. Vanrell. Names and shades of color for intrinsic image estimation. In *CVPR*, pages 278–285, 2012. 2, 5, 7
- [21] S. Shafer. Using color to separate reflection components. *Color Res. Appl.*, 10(4):210–218, 1985. 3
- [22] F. Shahbaz Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez. Color attributes for object detection. In *CVPR*, pages 3306–3313, 2012. 1
- [23] R. T. Tan and K. Ikeuchi. Separating reflection components of textured surfaces using a single image. *PAMI*, 27(2):178–193, 2005. 3, 4
- [24] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE TIP*, 18(7), 2009. 2, 7
- [25] E. Vazquez, R. Baldrich, J. Weijer, and M. Vanrell. Describing reflectances for color segmentation robust to shadows, highlights, and textures. *PAMI*, 33(5):917–930, 2011. 2
- [26] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *ICCV*, pages 1800–1807, 2005. 6
- [27] Q. Yang, S. Wang, and N. Ahuja. Real-time specular highlight removal using bilateral filtering. In *ECCV*, pages 87–100, 2010. 3
- [28] S. X. Yu and J. Shi. Segmentation given partial grouping constraints. *PAMI*, 26(2):173–183, 2004. 5
- [29] J. Zhu, K. G. Samuel, S. Z. Masood, and M. F. Tappen. Learning to recognize shadows in monochromatic natural images. In *CVPR*, 2010. 3