

Multi-view Aggregation for Color Naming with Shadow Detection and Removal

Mohamed Dahy Elkhoully^{1,2}, Stuart James¹, and Alessio Del Bue^{1,3}

¹Visual Geometry and Modelling (VGM) Lab, Istituto Italiano di Tecnologia (IIT), Italy

²Università degli studi di Genova, Italy

³Center for Cultural Heritage Technology (CCHT), Istituto Italiano di Tecnologia (IIT), Italy

Abstract—This paper presents a set of methods for classifying the color attribute of objects when multiple images of the same objects are available. This problem is more complex than the single image estimation since varying environmental effects, such as, shadows or specularities from light sources, can result in poor accuracy. These depend primarily on the camera positions and the material type of the objects. Single image techniques focus on improving the discrimination of between colors, whereas in multi-view systems additional information is available but should be utilized wisely. To this end, we propose three methods to aggregate image pixel information in multi-view that boost the performance of color name classification. Moreover, we study the effect of shadows by employing automatic shadow detection and correction techniques on the color naming problem. We tested our proposals on a new multi-view color names dataset (M3DCN) which contain indoor and outdoor objects. The experimental evaluation shows that one out of the three presented aggregation methods is very efficient and it achieves the highest accuracy in term of classification results. Also, we experimentally show that addressing visual outliers like shadow in multi-view images improves the performance of the color attribute decision process.

Index Terms—Multi-view geometry, Color classification, Aggregation, Shadow detection, Shadow removal.

I. INTRODUCTION

The classification of color is used for applications such as pedestrian re-identification, automatic visual captioning, visual question and answering, and many other computer vision tasks [1], [2] where the interpretation of the attribute “color” is fundamental. As an example, for content-based retrieval, knowing the color of the object in an image narrows the possible search space. Differently from the single view case, naming the color of an object results in a more complex task. 3D reconstruction [3] can help to inform the decision, but integration is nontrivial. Different views entail changes in illumination given both by the material of the objects and position of the light sources in the scene. Moreover, when seeing an object (fig. 1) with a wide camera baseline, occlusions might provide a partial view of the object or (self-) shadows may conceal the color.

Thus estimating the correct color while having a set of possibly noisy observations is a new and challenging open problem. Past research tried to solve the problem for single view by proposing more discriminative color algorithms [1], [5]–[8], but results are still unsatisfactory on in-the-wild datasets. This problem is due to two main reasons: Firstly,

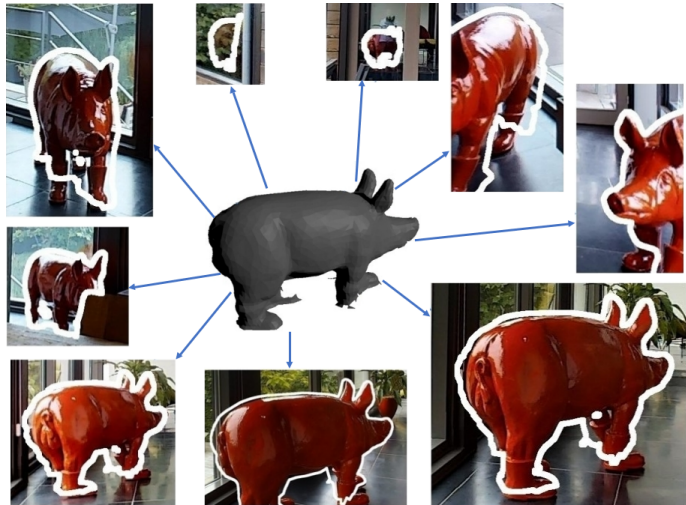


Fig. 1. The 3D mesh of an object from the Matterport 3D dataset [4] in center, with a subset of views surrounding. The surrounding views demonstrate different scales, specular reflections, and self shadowing.

colors ambiguity – boundaries between colors in color space are not well specified e.g. humans can not trivially decide what is the name of the color in the space between blue and green. As the problem is ambiguous for humans, this difficulty is translated to any computational approach. Secondly, illumination – colors are always affected by the environment lighting, and an object can be perceived as having different colors based on the surrounding light and its material reflection properties. For example, in low light or shadows, dark colors can be seen by humans as black, while white instead would be considered as gray and other colors will tend to be darker. However, if we have prior knowledge that a particular object is under shadow, this problem is made simpler. For this reason, color naming might be largely facilitated by finding the regions where the object is under shadow. In addition, if the object has specular highlights given its reflectivity, the related regions will also cause difficulty when deciding the color name.

In this paper we are trying to solve the multi-view color naming problem by proposing aggregation methods to utilize as much as possible of the information provided by the multiple images. In particular, we define three strategies for

merging the color information while attempting to reduce noise and outlying effects (e.g. shadows). To this end, before initiating the color labeling process, we try to obtain prior information about shadows regions, in order to either remove the shadow pixels or to correct them before our multi-view decision process.

We can list the main contributions in this paper as:

We propose three simple and robust aggregation techniques for multi-view color naming that utilizes wisely the available information across views;

We study the effect of shadows in multi-view color labeling and propose to integrate shadow information in the aggregation algorithms to enhance the performance; Present the first multi-view 3D dataset for color naming, Matterport 3D Color Naming dataset (M3DCN) ¹.

II. RELATED WORK

Early work in color naming by Berlin and Kay [9] identified eleven basic color labels shared among most human languages. One of the first work in automatic color labeling was by Lammens et al. [8] who proposed a fuzzy computational model where the membership of a set of Gaussians fitted to Berlin and Kay's labels predicted the color label. Seaborn et al. [6] fitted data from a psychophysical human experiment using fuzzy k-means to create the grouping of colors. Similarly, Mojsilovic [10] performed subjective experiments to label the colors from the Inter-Society Color Council-National Bureau of Standards (ISCC-NBS), creating a color vocabulary based on the agreement between their labels and ISCC-NBS color names. Vandenbroek et al. [11] also used subjective human experiments, creating a similar dictionary that are then used as markers within HSI space. Vandenbroek et al. then projected these markers into 2D using Fast Exact Euclidean Distance projection technique to create a complete segmentation of the color space. Benavente [5] defined the color naming task as a decision problem using fuzzy-set theory based on the definition of triple sigmoid with an elliptical center as a membership function for the different colors. Serra et al. [12] subsequently used the work of Benavente [5] as a color descriptor over regions which could be combined with edge descriptors for image decomposition through a Markov Random Fields (MRF) to create color aware homogeneous regions. The aforementioned works in automatic color labeling are generally referred to as chip based methods as it used color chips as training data.

Weijer et al. [7] proposed to learn from real-world images allowing more general color distributions to be learned. Their method learned a Probabilistic Latent Semantic Analysis (PLSA) topic distribution from the noisy data. Images pixels were represented as words in a LAB histogram feature and the ordered latent topics were used to provide a probability of colors. This method is still considered as the state-of-the-art in single image color classification and commonly used to generate visual question and answering ground truth. More

recently, Liu et al. [13] used the color naming as a part of fashion parsing model, in contrast to other referenced methods, they infer 13 color names and other attributes for each pixel via an MRF inference model. Similarly using MRF, Liu et al. [14] used the color-naming model proposed in [5] to build a MRF to propagate the color labels from regions under normal illumination to abnormal regions to help estimate color names.

Most recently, Cheng et al. [1] proposed an end-to-end, pixel-to-pixel Convolutional Neural Network (CNN) learned from a pedestrian color naming dataset to assign a consistent color name to regions of single object's surfaces. The color naming problem has also been considered from a multi-label perspective for ImageNet for assigning attributes to images by Russakovsky and Fei-Fei [15]. To the best of our knowledge there are no works about multi-view color labeling.

The shadow detection and removal problems were originally formulated as a physical model of illumination and color [16], [17]. These illumination invariant techniques performed well on high quality images and with calibrated sensors, but typically performed poorly on web-quality or consumer photographs. Alternative to the physical models, data driven approaches extracted features from pixel values or regions to treat the problem as a binary classification task. Different classifiers have been used such as SVM [18]–[20], kernel LS-SVM [21], [22] and decision trees [23], [24] to label image regions into shadow and non-shadow regions. After, an MRF graph-cut with energy minimization optimization provided spatial consistency [18], [20] to the mask. Also, Conditional variants using CRF [24] have been used to further improve spatial consistency.

Khan et al. [25] were the first to apply deep models for shadow detection, they used two CNN to learn deep features for shadow detection. Where one of the CNNs learned the boundary features and the other interior of the shadow. The predicted posteriors were then fed into a CRF for spatially consistent masks. End-to-end approaches were explored by Qu et al. [26] who used a three branch network, global, appearance and semantic, which are combined to generate a mask. Alternatively, Hu et al. [27] proposed Direction-aware Spatial Context RNN module which models the gradients that occur at shadow boundaries to infer the shadow region within the image. Nguyen et al. [28] introduced GANs for shadow detection, proposing to use Stacked Conditional Generative Adversarial Network (scGAN) where the loss of the trained shadow detector is parameterized through adding an additional sensitivity weight provided to the generator to weight examples and avoid the issue of unbalanced training data. Wang et al. [29] used GANs in a multi-task setting considering detection and correction based on scGAN. Where the first network generates the mask and second the shadow removed image which would be evaluate by an adversarial network. Le et al. [30] proposed GAN framework composed of two networks, a shadow attenuation network (A-Net) and a detection network (D-Net), which are jointly trained where the output of A-Net used to train D-Net and subsequently generate the shadow masks.

¹M3DCN available at <https://github.com/mohamed-elkhoully/M3DCN>

III. MULTI-VIEW COLOR NAMING AGGREGATION

We define the color labeling problem in multi-view as a data aggregation challenge where auxiliary information from views can mitigate shadow and specularities to achieve improved accuracy in classification. We therefore base our work on the state-of-the-art in single image color classification using PLSA from Weijer et al. [7] and extend it to apply different approaches to aggregate the information among views. At first, we obtain our multi-view data in two forms sparse and dense points (described in subsec. III-B) from the M3DCN (subsec. IV-A). The classification is then performed over 11 colors, as is common in prior work, we outline the method of Weijer et al. [7] (subsec. III-A). We then define three methods for aggregating this information across views (subsec. III-B), finally we address a common environmental issue of shadows that can cause significant changes in the classification decision. The full system pipeline shown in (Fig. 2).

A. Color Naming

Color classification is often treated as a topic learning problem, as in [7] which learns from weakly labeled images using a variant of the Probabilistic Latent Semantic Analysis (PLSA) model. Given a set of images $I = \{i_0; \dots; i_N\}$ in LAB color space, the set of pixels for I_j produce a frequency histogram by quantizing the color space by $[10; 20; 20]$ of LAB channels respectively. Where each histogram, as in traditional text analysis notation, represents a document $D = \{d_0; \dots; d_N\}$. Therefore, the set of words $W = \{w_0; \dots; w_m\}$ refer to the bins of the color space quantization. PLSA then optimizes a set of latent topics $Z = \{z_0; \dots; z_J\}$ through expectation maximization of the conditional probabilities as in

$$p(w|d) = \sum_z p(w|z)p(z|d); \quad (1)$$

where $p(w|z)$ and $p(z|d)$ are multi-nomial distributions (notation as per [7]). Weijer et al. proposed two ways to exploit the learned topics, through an indexed look-up of the word color probabilities (eq. 2) and exploiting a region prior. Given our multi-view scenario, we exploit the object segmentation, and therefore the prior is not required. The indexed approach is referred to by PLSA-ind:

$$p(z|w) \propto p(z)p(w|z); \quad (2)$$

Where the prior over the color names $p(z)$ is taken to be uniform.

B. Multi-View Color Naming aggregation

Given a sequence of images I , we propose three methods for aggregating the multiple views of a given object. The object views $V = \{v_1; \dots; v_r\}$ where $V \subseteq I$ provide the color image, object mask and the object 3D point cloud. We propose to use sparse points – which are projected from the 3D point cloud into the color image; and dense points – acquired by filtering the color image with the object masks. Then the corresponding LAB values of the points and their

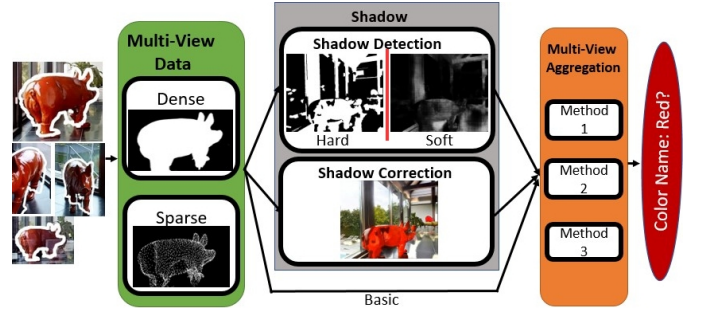


Fig. 2. Pipeline for multi-view aggregation for color naming using the object multi-view data. Experiments are performed by considering both sparse and dense points representation of the object. Then three different aggregation methods are defined. The first is passing the data to test on the aggregation techniques directly (basic). The second, applies shadow detection by using either hard or soft masks we exclude the data pixels labeled as shadow. The third, is to apply shadow correction then use the corrected data to test on the aggregation techniques. Finally the output is the color name of the object.

corresponding word then form a decisions C_p we use to refer to the distribution $p(w|z)$, and C_v to refer to the color decision for frame. The three methods are therefore as follows:

Method 1: Follows PLSA-ind [7] to predict the color of the object in each view v_i , then aggregated by the most frequent across views:

$$C_v = \text{mode}\{\text{argmax}_z(C_{pk}) : k = 1; \dots; n\}; \quad (3)$$

$$C_{\text{Object}} = \text{mode}\{C_{v_l} : l = 1; \dots; r\}; \quad (4)$$

where n is the number of object pixels in the given frame, C_{Object} is the color decision from all frames.

Method 2: Uses all probabilities and therefore not applying max which restricts the propagation of information. Defined as:

$$C_v = \text{argmax}_z(\sum_{k=1}^n \{C_{pk}\}); \quad (5)$$

$$C_{\text{Object}} = \text{mode}\{C_{v_l} : l = 1; \dots; r\}; \quad (6)$$

The difference between Method 1 and 2 can be seen in Fig. 3, which demonstrates how intra-class confusion can be lost at this initial stage.

Method 3: As with Method 2 the probability distribution C_p is used, in addition to considering an importance for each frame in the decision process. This importance will be calculated by the visible area of the object at each frame, using it as a weighting function. It is defined as:

$$C_{\text{Object}} = \text{argmax}_z(\sum_{l=1}^r !_l \sum_{k=1}^n C_{pk}) \quad (7)$$

where $!$ is a l_1 normalized vector containing the weight for each frame according to the visible area of the object.

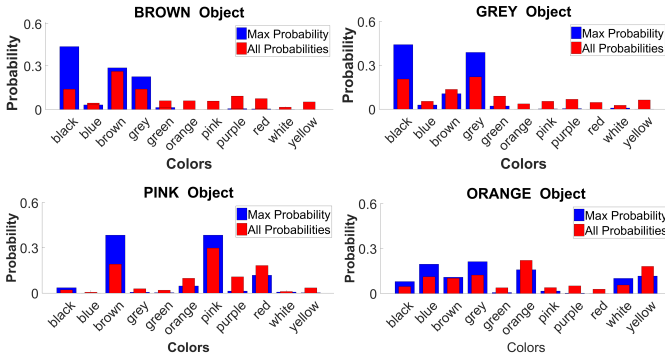


Fig. 3. The difference between using probabilities of colors of four objects from all of their frames against using the maximum probability at each color index for deciding the color name of the object. Using all probabilities (highlighted in red) is giving more information about all colors, and is better for the most occurring color in all frames even if it is not the maximum, which drives to the right label.

C. Using Shadow Detection and Removal in Color Naming

To avoid including outliers due to shadows on or by the object in classification we incorporate shadow detection. We look at three types of shadow analysis, soft shadow mask – a probability distribution over the image; hard shadow mask – a binary mask of areas in shadow; and shadow correction – for correcting the color within the shadowed regions.

Soft Shadow Mask: The posterior probabilities from the shadow detection algorithm corresponding to how much each pixel affected by shadow. The GAN-based framework ADNet of Le et al. [30] which generate soft shadow masks can be used as a shadow weighting for color classification C_{pk} . Their technique composed of attenuation network (A-Net) and a shadow detection network (D-net). The A-Net mainly was used to generate training examples for augmenting the training data for D-Net with hard-to-predict cases to fool it. Finally generating a soft classification shadow mask. To use the soft shadow mask as a weighting we normalize the soft mask and use it as a probability of shadow.

Hard Shadow Mask: A binary mask labels each pixel as either affected or not by shadow. Guo et al. [20] proposed a Graph Cut based method that outputs a binary segmentation, as the resultant optimization either falls into the source or sink. In Guo et al. approach classifying individual regions subsequently pairing regions to construct an MRF graph. The unary term is derived by classifying the paired regions to *same illumination* and *different illumination* according to their similarity on appearance and textures and the pairwise term is the distance within the image between regions energy minimization is applied to solve for binary labeling. In addition we can apply thresholds to the soft shadow technique of Le et al. [30] to get the hard masks, using three thresholds [0.3;0.6;0.9] to neglect shadows under the threshold (see Fig. 4). Other thresholds were empirically evaluated and had similar results, for simplicity we only mention three levels.



Fig. 4. From left to right: original image, hard shadow mask from soft shadow mask using thresholds 0.3, 0.6, and 0.9 respectively. Finally the most right is the soft shadow mask, where shadow is the white label pixels

Shadow rejection condition: Dark objects are often misclassified as in shadow (see fig. 6), also noted in [27]. Therefore, we apply a condition to decide when to apply shadow masks, where if $> 70\%$ of the object is labelled as in shadow, the weighting is not applied for that image.

Shadow Correction: To correct the color for regions under the shadows mask we apply the technique of [20]. For a given region pair R_i (as described in Hard Shadow) which falls under two sources of lights, direct L_d and environmental L_e , see eq. (8). They correct the shadowed region pairs identified as *same illumination* using the higher illumination with the assumption the illumination conditions are preferred. Constrained by estimating a fractional shadow coefficient value using a matting technique and the ratio of direct to environmental light in each color channel, enabled them to recover a shadow-free region, see [20] for more details.

$$I_i^{shadowfree} = (L_d \cos \theta_i + L_e) R_i; \quad (8)$$

As the shadow correction technique depends on the same method as in hard shadow masks [20], we apply the same shadow rejection condition to improve the correction results.

IV. EXPERIMENTS AND DISCUSSION

A. The Matterport3D Color Naming Dataset (M3DCN)

We propose the Matterport3D Color Naming dataset (M3DCN) for multi-view color naming which is based on objects from the Matterport3D dataset [4]. M3DCN was collected by asking 7 participants to annotate the color names of objects from the Matterport3D dataset. Each participant was presented with the visible parts of objects from all views outlined using the object mask to make it clear which object to focus on (see Fig. 1). We then use the objects consensus with uniform color label in our dataset which was class balanced resulting in 10 objects per color, 110 total. Although some categories are dominant, *pink* and *orange* objects are rarely present in the Matterport3D scenes. M3DCN objects vary in material and number of views (see Fig. 5). We show the semantic class distribution of objects in Table I. For conciseness in the table, we group similar objects under one label *e.g.* sofa_chair or barbers_chair are grouped under chair.

B. Results & Analysis

We extensively evaluate the different permutations of the three proposed aggregation methods and the effect of incorporating hard and soft shadows for both sparse and dense points shown in Table II by classification accuracy of these different configurations.

TABLE I
FREQUENCY OF SEMANTIC CLASSES IN M3DCN DATASET.

Object	Count	Object	Count	Object	Count	Object	Count
fireplace	1	canister	1	picture	2	toilet	1
bathub	1	floor	3	lamp	3	sink	1
trashcan	3	door	2	grass	1	stool	1
bed_sheet	1	vanity	1	guitar	1	kettle	1
door_frame	1	table	3	tv	1	pan	1
decoration	4	chair	17	bed	2	pot	2
clothes	1	pool	1	curtain	1	toaster	1
cushion	3	pillow	13	stair	1	plant	3
container	1	recliner	2	towel	1	cabinet	7
light	1	chaise	1	stand	1	wall	9
concrete	1	kitchen island	1	couch	6	Total	110

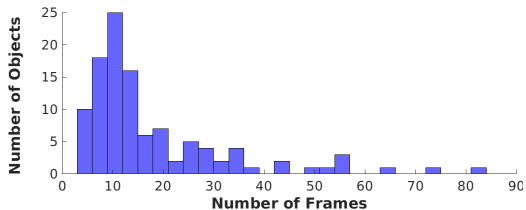


Fig. 5. Statistics for the frequency of images (V) for the 110 objects in the M3DCN dataset.

When considering the use of sparse and dense points as input to the multiview classification within the basic approach (no shadow detection or correction), it can be seen in two of the three cases there is an improvement by using dense points with significant improvement for Method 1 and Method 3 with 5% and 4% respectively. Whereas Method 2 suffers a decrease of 2% caused by the sensitivity to the size of the object and an increase in noise present by considering all posterior probabilities. This behavior is generally seen across other experiments considering shadow detection or correction.

When applying shadow detection we show results with and without shadow rejection condition (latter shown in parentheses). Applying shadow detection improves the performance for sparse on average for method 2 and 3 in sparse with 0.4% and 1.5% increase on average across all shadow detection methods and thresholds. In general, it can be seen that it either improves the performance or has consistent performance. In contrast for dense points, there is a consistent improvement with 0.5%, 0.9% and 1.5% across methods with almost all methods and configurations improving on the basic performance.

When using hard shadows from Guo et al. [20] or our threshold variants of Le et al. [30], in the majority of cases there is an improvement in contrast to the basic methods. Guo et al. can be seen to consistently improve the accuracy, alternatively Le et al. is sensitive to the threshold being applied. Soft shadows improve on the basic method for both types of points but can be improved by a carefully selected threshold. Sample confusion matrices for weighting and excluding points based on soft and hard shadow detection respectively can be seen in fig. 7, we compare using soft shadow masks against hard shadow masks for Method 3. In the confusion matrices,

TABLE II
CLASSIFICATION ACCURACY OF THE PROPOSED THREE METHODS OF AGGREGATION USING SPARSE AND DENSE OBJECT POINTS, SHADOW DETECTION (HARD/SOFT) AND SHADOW CORRECTION. HIGHLIGHTED IN RED AND BLUE ARE ANOMALOUS RESULTS ANALYZED IN SEC IV-B AND IN PARENTHESES ARE WITHOUT THE SHADOW REJECTION CONDITION.

Testset		M3DCN				
Aggregation Method		Method 1	Method 2	Method 3		
Sparse	Using Shadow Detection	Basic	Guo	63.64	69.09	70.91
			ADNet Threshold=0.3	(56.36)	(60.91)	(66.36)
			ADNet Threshold=0.6	66.36	70.00	70.91
			(63.64)	(70.00)	(70.00)	
			ADNet Threshold=0.9	72.73	68.18	72.73
	(64.55)	(68.18)	(71.82)			
	soft shadow	ADNet	63.64	69.09	70.91	
		(63.64)	(69.09)	(70.91)		
	Using Shadow Correction	Guo	67.27	68.18	71.82	
	(70.91)	(72.73)	(74.55)			
Dense	Using Shadow Detection	Basic	Guo	67.27	67.27	75.45
			ADNet Threshold=0.3	68.18	69.09	79.09
			(59.09)	(64.55)	(67.27)	
			ADNet Threshold=0.6	68.18	69.09	76.36
			(67.27)	(70.00)	(71.82)	
	hard shadow	ADNet Threshold=0.6	67.27	67.27	76.36	
		(67.27)	(65.45)	(67.27)		
	ADNet Threshold=0.9	68.18	67.27	76.36		
		(68.18)	(67.27)	(76.36)		
	soft shadow	ADNet	67.27	68.18	76.36	
(67.27)	(68.18)	(76.36)				
Using shadow Correction	Guo	68.18	68.18	78.18		
(70.91)	(71.82)	(77.27)				

it can be seen that colors of objects that are usually lighter (grey, orange, pink, and white) are misclassified in the case of soft shadow in favor of their darker equivalence, e.g., white to gray. This can be explained by the explicit hard masks being more aggressive as well as being spatially consistent by the MRF.

It is interesting to note that, when applying the shadow rejection condition (without parentheses), in most cases it shows significant improvement except in the cases highlighted in blue, which are using shadow correction. In such cases, the over-saturated color from the correction makes classification simpler even if it does not create visually appealing results.

Overall Method 2 is always better than Method 1 except in two cases (highlighted in red), while Method 3 is always better than both Method 1 and 2 in all cases. Showing that using per point probabilities (not max or mode) and the visible area weighting of objects helps to come to a reliable classification. Method 3 with shadows provides improvement of 10% for sparse and 11.8% for dense over the lowest performing basic methods showing a benefit using multiple views.

V. CONCLUSION

We propose three methods to solve the multi-view aggregating of decisions for the color naming problem. We have shown how the outliers caused by shadows affect the ability to classify color, and how excluding or correcting the affected pixels improves the classification accuracy. Testing on our proposed dataset M3DCN dataset, we achieved an accuracy of 79.09%. The proposed methods can be further extended to consider other outliers like specular highlights. While the methods are amenable to be applied to other probabilistic visual attribute problem in multiple views.

