Using Color Compatibility for Assessing Image Realism

Jean-François Lalonde and Alexei A. Efros
School of Computer Science, Carnegie Mellon University
http://graphics.cs.cmu.edu/projects/realismcolor

Abstract

Why does placing an object from one photograph into another often make the colors of that object suddenly look wrong? One possibility is that humans prefer distributions of colors that are often found in nature; that is, we find pleasing these color combinations that we see often. Another possibility is that humans simply prefer colors to be consistent within an image, regardless of what they are. In this paper, we explore some of these issues by studying the color statistics of a large dataset of natural images, and by looking at differences in color distribution in realistic and unrealistic images. We apply our findings to two problems: 1) classifying composite images into realistic vs. non-realistic, and 2) recoloring image regions for realistic compositing.

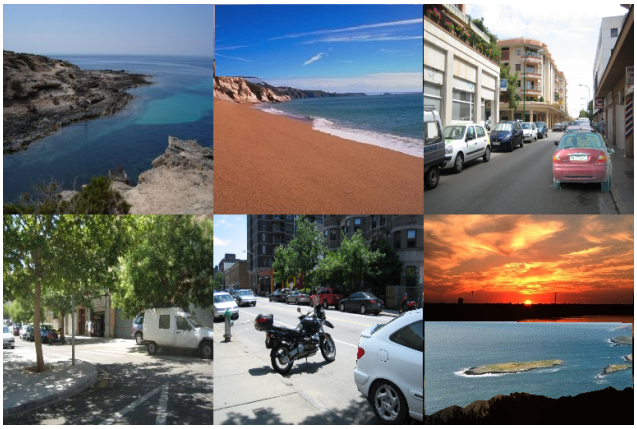

Figure 1. There are only two real images in this montage. Can you identify them?

1. Introduction

Consider the images shown on Figure 1. Only two of them are real. The rest are composite images – created by taking an object from one image and pasting it into a different one. The four synthetic images have been picked from a set of automatically generated composites and, as you can see, some look reasonably real while others appear quite fake. What is it, then, that makes a composite image appear real? Clearly, scene semantics and geometry play a key role [2] – a car floating in midair or twice as big as other cars would instantly appear out of place. In this paper, we will assume that these high-level scene structural issues have been dealt with (for an example of a user-guided approach, see [10]). Here, we are interested in investigating the more subtle artifacts that appear even if the semantic composition of the scene is correct (e.g. right column of Figure 1).

Difference in scene lighting between the source and destination images is one important consideration. The same object photographed under two different types of illumination (in a thick forest vs. a sunny beach) will usually have a strikingly different appearance. But does this mean that differently lit objects when placed in the same image will always appear inconsistent to a human observer? Not necessarily. Cavanagh [3] uses examples from art to demonstrate that humans are curiously unaware of the great liberties that artists often take with physics, including impossible lighting, inconsistent shadows, and incorrect reflections. Actually, this is not too surprising, considering that it is extremely difficult for both human and computers to estimate the true lighting environment map from a single image. But one important component of lighting that is readily available in an image is color. Indeed, the experience of Photoshop artists confirms that "getting color right" is one of the most important tasks for good image composites [1]. Therefore, in this paper, we have chosen to concentrate on the role of color in image compositing.

1.1. The Role of Color

The first question one must ask is whether color itself is the important cue for visual compatibility or just a manifestation of some higher-order semantic relationships? Do we prefer certain shades of green with certain shades of blue, or do we just like to see grass and sky together? This is a very difficult question. While the experience of graphic designers as well as recent work on color harmony [5] suggests that humans do prefer palettes of certain colors over others,

it is likely that object identity also plays a role. In this paper, we plan to investigate how far we can get without the use of explicit semantic information.

Our second question relates to the nature of color compatibility. Do different pairs of colors simply appear to be more or less compatible with each other? Or perhaps compatibility is evaluated over entire color distributions rather than individual colors? In this paper, we will take a first step in trying to answer some of these questions.

1.2. Prior Work

In computer graphics, people have long been interested in methods for adjusting the colors of one image to make it match the "color mood" of another image. In a much-cited work, Reinhard *et al*. [16] propose a very simple technique based on matching the means and variances of marginal color distributions between images using the $L\alpha\beta$ color space. The central assumption of the method is that the marginal color distributions of the object and its background should match. This technique was applied to compositing of synthetic objects into real videos [15], although no quantitative evaluation of the method's effectiveness was presented. While this makes sense for some cases, often you find an object becoming very greenish due to being pasted into a forest scene. To address this problem, Chang *et al*. [4] proposed to first assign each pixel to one of 11 "basic color categories" obtained from psycho-physical experiments and relating to universal color naming in languages. The color adjustment is then performed only within each category. This method produces much more pleasing results (on the 7 images shown) but, again, no quantitative evaluation is performed. The Color Harmonization approach [5] adjusts the hue values of the image color-map according to a predefined set of templates that are thought to encode color harmony. Alternatively, Pérez *et al*. [14] propose to copy the object *gradients* and reintegrate to get the colors. While this approach results in a seamless insertion, it often generates noticeable artifacts such as color bleeding or severe discoloration when the object and background have dissimilar color distributions (again, these works are difficult to evaluate since no quantitative results are shown).

Computer vision researchers are more interested in the task of classifying images based on various characteristics. Farid and colleagues have done extensive work on using higher-order image statistics for a variety of tasks, including distinguishing between computer renderings and photographs [12], detecting digital tampering, finding art fakes [13], etc. However, their efforts are directed towards detecting differences that are not perceptible to a human observer, whereas our goals are the opposite. Cutzu *et al*. [6] present a neat technique for distinguishing paintings from photographs based on the color distribution of the image. Their insight is that paintings are likely to exhibit more color variation than photographs because it is difficult to mix paints that have the same chromaticity but different intensity. Ke *et al*. [9] propose a set of high-level image features for assessing the perceived artistic quality of a photograph. Their color features include a measure of histogram similarity to a set of high-quality photographs, as well as an estimate of hue variation in the image (apparently, professional photos have fewer hues).

The classic paper by Forsyth [8] has spawned a large body of work in color constancy. However, their goals are different: retrieve the illuminant under which a given scene was lit, from a list of known illuminants. One method related to the present work has been introduced by Finlayson *et al*. [7], in which they determine the illumination based on nearest-neighbor matching to a set of illumination palettes. More recently, Lalonde *et al*. [10] presented a data-driven technique for image compositing that uses a coarse illumination context descriptor to find scenes with similar lighting in a large database.

1.3. Overview

In this paper, our goal is to use color information to automatically predict whether a composite image such as the ones in Figure 1 will look realistic or not to a human observer. In the pursuit of this endeavor, two different and complementary approaches are evaluated.

The first approach utilizes the fact that phenomena that happen in the real world are, by definition, natural. This translates to the hypothesis that colors in an image will look realistic if they appear in real images with high probability [11]. We pose the problem as follows: given a set of colors (a palette), what are the other color palettes that are likely to co-occur in the same image? In Section 3, we propose several ways to estimate color palette co-occurences.

Our second approach does not consider global color statistics and makes the assumption that a composite image will look realistic if the object and background colors have similar distributions. This idea is directly inspired by the work of Reinhard *et al*. [15], which has never been evaluated rigorously on a large number of examples. We propose to extend the work using a better color representation and provide an extensive comparative evaluation in Section 4.

Finally, from the intuitions gathered while evaluating the global and local approaches, we suggest a way of combining them into a single classifier. Section 5 presents this combined approach and compares it to using either technique by itself. As an additional application, we show in Section 6 how to automatically shift the colors of an unrealistic object to make it look more realistic in its new scene.

2. Generating the Composite Image Dataset

In order to compare the different approaches proposed, a dataset of synthetic images was generated semi-

Figure 2. Example images randomly selected from our test set. *Top row*: Real images. *Middle row*: Realistic synthetic images. *Bottom row*: Unrealistic synthetic images. The entire test database used to produce the results presented in this paper contains a total of 1000 images and was semi-automatically generated from images taken from the LabelMe database [18].

automatically[1].

Since the process of manual image compositing can be long and tedious, we seek to automatically generate composite images that will look right semantically, i.e. objects should be at the appropriate locations in the resulting images. We propose a very simple algorithm to generate semantically correct images by utilizing a large segmentation dataset. We use the popular LabelMe image database [18] which contains roughly 170,000 labeled objects. We first remove all incomplete objects by searching the label strings for words "part", "occlude", "regions" and "crop". We then manually group objects that have similar labels, and end up with the following 15 most frequently-occurring objects in the dataset: "building", "bush", "car", "field", "foliage", "house", "mountain", "person", "road", "rock", "sand", "sky", "snow", "tree", and "water".

Because segmentations are available, we can create a synthetic composite by starting with an image, and replacing one of its objects by another one of the same semantic type and shape. The algorithm only selects the objects that occupy at least 5% and at most 60% of their corresponding image area. The shape matching is done by computing the SSD over blurred and subsampled object masks, allowing for translations. Once the best matching object is found, we paste it onto the original image and apply linear feathering along the border to mask out potential seams. Even though this algorithm is very simple, it performs surprisingly well (see Figures 1 and 2), because it exploits the richness of the dataset.

[1]The dataset is publicly available online.

Some of the automatically generated composite images happen to have matching colors and appear quite realistic, while others have color distributions that make them look unrealistic. Sometimes, however, the automatic procedure fails completely, producing results that are structurally inconsistent and obviously wrong. We label those as unsuccessful and manually remove from the test data. We asked three human observers with normal color vision to label the remaining images as either realistic or unrealistic. The realistic class is augmented with randomly selected real images from the dataset. For real images, a random object in that image is selected to be the tested inserted object. Our final test set is composed of 360 unrealistic, 180 real, and 180 realistic images. Figure 2 shows examples of typical real and synthetic (realistic and unrealistic) images in our test set, which remained identical for all the experiments performed in this paper. We also employ a much larger and non-overlapping part of the LabelMe dataset containing 20,000 images to compute the natural color statistics in the following experiments.

3. Global Natural Color Statistics

In this section, our aim is to find a way to test the naturalness of colors in a given image by computing their similarity with global color statistics accumulated over a large set of real images. We propose three ways of doing so.

3.1. Universal Color Palette

The simplest way of modeling the joint natural color distribution is to assume that an image is generated ac-

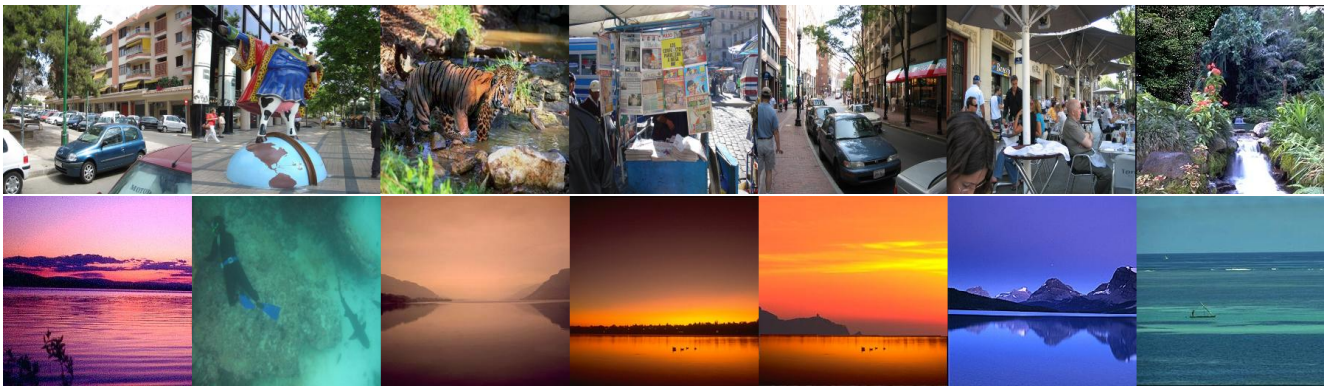

Figure 3. Most and least realistic images ranked by the universal palette algorithm. *Top row*: 7 most realistic images in the test database. *Bottom row*: 7 least realistic images. Note that this first-order analysis is completely unable to tell apart realistic images from unrealistic ones.

cording to a single, global distribution, which is the same for all images. This assumes the existence of a universal palette, from which all natural images are generated. While this first-order assumption is restrictive, we can easily estimate the joint distribution of all training images by computing a global 3-dimensional joint histogram in CIE L*a*b* color space. Given a new image, we compute its color histogram, and compare it to the global model. Unless otherwise noted, all experiments in this paper are performed in the CIE L*a*b* color space using 3-D joint histograms with 100^3 bins. Figure 3 presents images that are close (top row) and far (bottom row) from this global distribution, using the χ^2-distance metric for histogram comparison. It illustrates that the universal palette has the tendency to (falsely) predict that images with a single, saturated color are less likely to be realistic. Whereas this approach is clearly not useful in our image realism classification setting, it might still have interesting applications, such as finding striking and unusual color photographs in a large dataset (see Figure 3, bottom row).

3.2. Expected Color Palette

While the universal palette is easy to compute, it appears not to be powerful enough to model the complexity of natural images because it only models first-order color statistics. A better approach would be to consider pairs of colors that appear together in the same image. Given a 3-dimensional color space, this is a 6-dimensional function that represents the probability of observing a color distribution given a single color. Stated differently, it is modeling the palette that is likely to co-occur together with a particular color in a real image.

We represent this distribution by a 6-dimensional histogram of 16^6 bins. For every color in every object in the entire training dataset, we compute the histogram of all the colors occurring in that objet's background region. For testing: given a composite image composed of an object and its background, we sample the 6-D histogram by marginalizing over all the colors in the object and compare it with the histogram of the background.

Unfortunately, this method performs only marginally better than the first-order approximation, and still does not yield satisfying results on our test dataset. To quantitatively compare the different techniques, we use the χ^2-distance metric to assign a realism score to every image in our test dataset and construct ROC curves; the same procedure is used in the remainder of the paper. In our experiments, the area under the ROC curve for the universal palette is 0.59, and 0.61 for the second-order. Several reasons explain this poor performance: the 6-dimensional histogram is likely too coarse and smoothes over important color shades. More importantly, this model only represents a single expected palette given a single color, which is not enough to capture the complexity of our visual world. For instance, blue sky co-occurs with grass, roads, seas, cities, etc. each of which might exhibit very different color palettes. Since this method is computing the average over all observed instances, it is not powerful enough to model each of them jointly.

3.3. Data-Driven Color Palette

To address the limitations of the previous method, we would ideally need to know the co-occurences of all possible color palettes. Since this number is huge, we would require a prohibitively large number of images and computing power to compute them.

Although it might be possible to employ the recent method of Yang *et al*. [19] to extract a more powerful co-occurence feature, we note instead that because we have large amounts of real data with labelled objects, we can use a nearest-neighbor approach to approximate this distribution directly. Given the object color palette, we find a set of k most similar-looking objects based on color (k-NN), and approximate its expected co-occurring palette by the best-

α	0	0.25	0.5	0.75	1
ROC AUC	0.59	0.69	0.74	**0.79**	0.74

Table 1. Influence of color and texture on the nearest-neighbor retrieval. Using texton only ($\alpha = 0$) fails to return good nearest neighbors, and maximum performance is obtained when $\alpha = 0.75$ (in bold). Scores are obtained by computing the ROC area under the curve (AUC) on our test set.

technique	marginals	joint	joint with texture
ROC AUC	0.66	0.76	**0.78**

Table 2. Summary of local techniques. The best performance (bold) is obtained by combining a joint histogram representation with texture matching using texton histograms. Scores are obtained by computing the ROC area under the curve (AUC) on our test set.

matching background in this k set. This method yields an area under the ROC curve of 0.74, a significant improvement over the previous techniques.

Can we improve its performance further? Let's consider an example. When determining if a tree matches a particular forest scene, it might be more important to look for similar forest images, which typically have very consistent color palettes, than for green buildings which might exhibit different shades of green and still look realistic. Clearly, incorporating object recognition could greatly help in matching similar scenes. Here, we experiment with a weak recognition cue by using texture matching between images.

First, a texton dictionary of 1000 instances is learned by clustering 32-dimensional oriented filter responses on our 20,000 training images. A texton histogram can then be computed for each object and associated background in the training data and be used in the k-NN process.

To evaluate the distance between two objects, we take a linear combination of their color and texton histograms χ^2-distances. A parameter α is used to control the relative importance of color and texture, where $\alpha = 0$ indicates only texture, and $\alpha = 1$ means only color. In Table 1, we provide a comparative evaluation using $\alpha = \{0, 0.25, 0.5, 0.75, 1\}$, progressively increasing the influence of color. We observe that texture information improves upon results obtained with color only, but is ineffective when used alone, which seems to confirm our intuition.

The limitations of such an approach is that it wholly depends on the training data. Given the huge dimensionality of the space of all scenes, we cannot expect to find a matching scene for any given image, even with a dataset of 20,000 images. We will now investigate a different, more local class of techniques, which can hopefully compensate when the training data cannot help.

4. Local Color Statistics

Reinhard *et al*. [15] have demonstrated a very simple way of making an object match its background by shifting its colors to make them closer to the background colors. The intuition behind this idea is that this shift appears to be increasing the correlation between the object and background illuminants. For example, the reddish hue of a sunset sky should appear on all objects in the scene. An object taken from a bright day scene can be made to look better in the new sunset background by shifting its colors towards red. While their application is in image recoloring, it can also be used in our context by computing the distance between the object and its new background colors.

The color description used in [15] is a simple marginal histogram in $L\alpha\beta$ color space. A straightforward improvement is to use the full 3-D joint histograms instead of marginals because color components are still quite correlated even in $L\alpha\beta$. Interestingly, this yields substantial improvement, going from an area under the ROC curve of 0.66 for marginals to 0.76 for joint on our test data, at the cost of a higher-dimensional representation. The same intuition of using texture as mentioned in the previous section also applies here, and improves performance as well, as shown in Table 2.

5. Combining Global and Local Statistics

Let us consider for a moment the two techniques introduced in the previous sections. When the global method yields a high realism score, we can be confident that it is correct because it relies on matches to actual real scenes. However, when it is uncertain, it means that no good match was found, and the results are not reliable. We can then only rely on the local approach. This suggests that we can combine both global and local ideas into one coherent classifier.

We propose a two-stage cascade. First, the algorithm computes the distance to the nearest-neighbor according to the best global measure from Section 3. If the match is good enough (as determined by a threshold τ), it classifies this image as realistic. Otherwise, it uses the local method from Section 4 to assign a realism score. We use 10-fold cross-validation on our labeled dataset to determine the best parameter τ (0.35 in our case) which will maximize the area under the ROC curve.

Figure 4 shows the ROC curves for the best techniques presented in this paper. We compare our techniques against the baseline proposed by Reinhard *et al*. [15]. The results clearly show that combining the global and local techniques results in performance superior to any single one. In another paper [16], they make a point of using $L\alpha\beta$ color space. We performed each experiment by using the $L\alpha\beta$, CIE L*a*b*, HSV and RGB color spaces and found that the CIE L*a*b* color space performs the best, closely followed by $L\alpha\beta$.

In Figure 5, we show a visual representation of the rank-

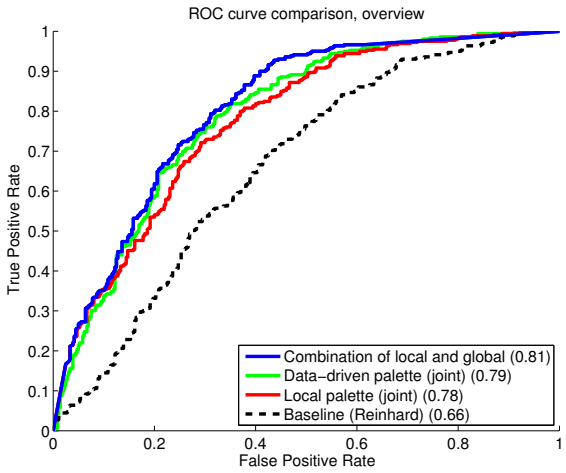

Figure 4. The ROC curves comparing the best approaches of the global and local categories, as well as the combination of both, shown against the baseline method, suggested by Reinhard *et al*. [15]. The combination of both local and global methods outperforms any single method taken independently, and is significantly better than the baseline approach.

ings produced by our combined classifier on our test set. Images are shown ranging from least (top) to most (bottom) realistic. Ground truth labels are illustrated by color borders: a red border indicates an unrealistic image, green and blue indicates realistic and real respectively. It is interesting to observe the output of the algorithm at mid-range where confusion is higher: even for humans, assessing image realism is much less obvious and requires a more careful inspection than for images at both ends of the spectrum.

6. Application to Automatic Image Recoloring

An interesting application as well as a visual evaluation of this technique is the recoloring of an image to make it appear more realistic. The idea is to first classify the image using our proposed method and retrieve either a nearest scene (if the global method is used), or the determination that no matching global scene is available. In the first case, we need to recolor the object to match the colors of similar objects in that nearest scene. In the second case, we can only try to make the object more similar to its surroundings, as in [15].

The goal of recoloring is to modify a source color distribution D_s in order to match a target color distribution D_t. In our setting, D_s represents the object colors, and D_t is the nearest neighbor object if the global method is used, or the background otherwise.

Our color matching procedure is an automatic extension of the interactive recoloring approach from [16], where they propose to represent both D_s and D_t by k color clusters, with k being manually chosen to be the number of major colors in the scene. Each cluster in D_s is matched to a cluster in D_t by comparing their means and variances.

Instead, we propose an entirely automatic algorithm. Each color distribution is represented by a mixture of k spherical gaussians ($k = 100$ and remains constant for all images), and the distributions are matched in a soft way using the solution to the well-known transportation problem. The algorithm is divided in three steps. First, we use the Earth Mover's Distance algorithm [17] to compute the best assignment between the clusters in D_s and D_t. Second, color shift vectors for each cluster in D_s are computed as a weighted average of its distance in color space to each of its assigned clusters in D_t. Finally, every pixel in D_s can be recolored by computing a weighted average of clusters shifts, with weights inversely proportional to the pixel-cluster distance. These three steps are performed in the CIE L*a*b* color space, and the results are converted back to RGB for visualization. Examples of recoloring using the global and local models are illustrated in Figure 6, and show that we can automatically improve the realism of composite images by using the same general approach.

7. Conclusion

In this paper, we study the problem of understanding color compatibility using image composites as a useful application domain. We propose two measures for assessing the naturalness of an image: 1) a global, data-driven approach that exploits a large database of natural images, and 2) a local model that depends only on colors within the image. We show that while both techniques provide substantial improvement over previous work, the best approach needs to use both techniques for different types of images.

We evaluate our approach on a large test dataset of synthetic images, generated by a novel semi-automatic technique. We are the first work in this field to provide a quantitative evaluation and we demonstrate performance superior to the state of the art method [15]. We also qualitatively validate our approach using a novel image recoloring method that makes composite images look more realistic.

A number of issues, such as position of objects in the image, object semantics and material properties still need to be addressed. While this paper is only a first step, lessons learned from the presented experiments should be extremely useful to steer this area towards a better understanding of natural color statistics and color perception.

Acknowledgements

We gratefully acknowledge Bryan Russell, Antonio Torralba and the rest of the LabelMe team for setting up and maintaining the database. This work has been partially supported by NSF grants CCF-0541230 and CAREER IIS-0546547.

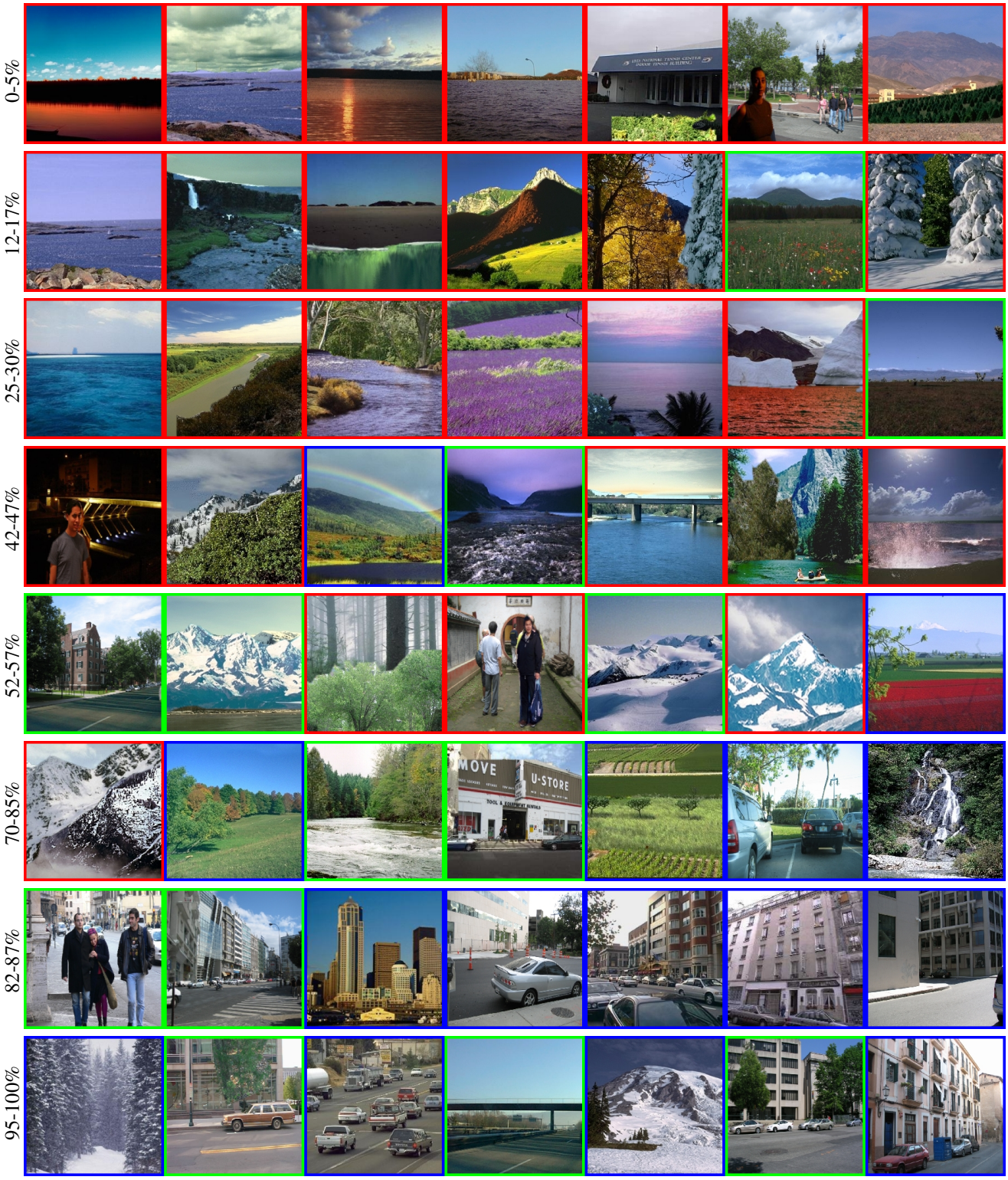

Figure 5. Images ranked according to their realism score, determined by combining the best global and local methods. The border color indicates the labeled class for each image (red: unrealistic synthetic; green: realistic synthetic; blue: real). Each row corresponds to the percentile interval shown on the left, and images are randomly selected within each interval. *Row 1*: 0-5% interval (most unrealistic), *2*: 12-17%, *3*: 25-30% , *4*: 42-47%, *5*: 52-57%, *6*: 70-75%, *7*: 82-87%, *8*: 95-100% (most realistic).

(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6. Automatic image recoloring. The input images (a,b,c,d) are recolored by using the local (e,f) and the global statistics (g,h). Recoloring these unrealistic input images increases their realism.

References

[1] E. H. Adelson. personal communication, 2006.

[2] I. Biederman. On the semantics of a glance at a scene. In M. Kubovy and J. R. Pomerantz, editors, *Perceptual Organization*, chapter 8. Lawrence Erlbaum, 1981.

[3] P. Cavanagh. The artist as neuroscientist. *Nature*, 434:301–307, March 2005.

[4] Y. Chang, S. Saito, K. Uchikawa, and M. Nakajima. Example-based color stylization of images. *ACM Transactions on Applied Perception*, 2(3):322–345, 2005.

[5] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, and Y.-Q. Xu. Color harmonization. *ACM Transactions on Graphics (SIGGRAPH 2006)*, 25(3):624–630, 2006.

[6] F. Cutzu, R. Hammoud, and A. Leykin. Estimating the photorealism of images: Distinguishing paintings from photographs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[7] G. D. Finlayson, S. D. Hordley, and P. M. Hubel. Color by correlation: A simple, unifying framework for color constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1209–1221, 2001.

[8] D. A. Forsyth. A novel algorithm for colour constancy. *International Journal of Computer Vision*, 5(1):5–36, July 1990.

[9] Y. Ke, X. Tang, and F. Jing. The design of high-level features for photo quality assessment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[10] J.-F. Lalonde, D. Hoiem, A. A. Efros, C. Rother, J. Winn, and A. Criminisi. Photo clip art. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), August 2007.

[11] R. B. Lotto and D. Purves. The empirical basis of color perception. *Consciousness and Cognition*, 11(4):609–629, December 2002.

[12] S. Lyu and H. Farid. How realistic is photorealistic? *IEEE Transactions on Signal Processing*, 53(2):845–850, 2005.

[13] S. Lyu, D. Rockmore, and H. Farid. A digital technique for art authentication. *Proceedings of the National Academy of Sciences*, 101(49):17006–17010, 2004.

[14] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (SIGGRAPH 2003)*, 2003.

[15] E. Reinhard, A. O. Aküyz, M. Colbert, C. E. Hughes, and M. O'Connor. Real-time color blending of rendered and captured video. In *Interservice/Industry Training, Simulation and Education Conference*, December 2004.

[16] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications, special issue on Applied Perception*, 21(5):34–41, September - October 2001.

[17] Y. Rubner, C. Tomasi, and L. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision and Image Understanding*, 2000.

[18] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. Technical Report AIM-2005-025, MIT AI Lab Memo, September 2005.

[19] L. Yang, R. Jin, C. Pantofaru, and R. Sukthankar. Discriminative cluster refinement: Improving object category recognition given limited training data. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.