Mining Probabilistic Color Palettes for Summarizing Color Use in Artwork Collections

Supplementary Material

Ying Cao   Antoni B. Chan   Rynson W.H. Lau

City University of Kong Kong

1 Color-based Segmentation and Quantization

We assume that each artwork is composed of disjoint color regions, each of which is assigned one of K discrete colors. Therefore, we quantize each artwork in a collection in terms of both spatial and color space. For spatial quantization, we perform a color-based segmentation to subdivide an artwork into a set of regions. In particular, we first apply the quick shift algorithm [VS08] to over-segment an artwork into smaller segments based on the Lab color space. Then, we group the nearby segments to form larger regions using the DBSCAN algorithm [EKSX96]. Two adjacent segments a and b are merged into one cluster if their color difference is smaller than a grouping threshold, i.e., $\|c_a - c_b\|_2 \leq \gamma$, where c_a and c_b are the median colors of the two segments in Lab space. We empirically set $\gamma = 10$ to preserve the color richness of the artworks. In addition, to preserve the structure of objects in the image, we use the Pb boundary detector [MFM04] to compute an edge map, which is used to guide the grouping so that two nearby segments are not grouped if there exists any strong edge along their contact boundary (i.e., the average intensity of the edge map region around the contact boundary is smaller than 0.2). For color quantization, we first represent each segmented region using its median color, and quantize the region colors of all the artworks into K clusters using the k-means algorithm. Then, for each cluster, we select the color closest to the cluster center as a representative color, forming a visual dictionary of K representative colors ($K = 300$ in our implementation). Finally, each region is assigned its closest representative color.

2 Semantic Labeling

In semantic labeling step, we aim to assign each region a semantic label to represent its semantic meaning (e.g., face and hair). To this end, we make use of a deep convolution neural network (CNN) to cluster the regions into L semantic classes in a semantic representation space, and use cluster indices as the semantic labels. More specifically, we first label the regions that touch image boundaries and take up more than 30% of their respective artworks as a background class. For the remaining regions, our semantical labeling method proceeds as follows. We first map each region into a semantic feature vector using the convolution neural network (CNN) in [SM15]. The CNN is trained with more than one million illustrations to predict illustration tags and ratings, and thus their high-level features can reflect the semantic meanings of input images. For each region, we rescale the image contents covered by its bounding box to an 224×224 image patch as the input to the CNN, and take the outputs of the last hidden layer as its feature vector. Then, we apply

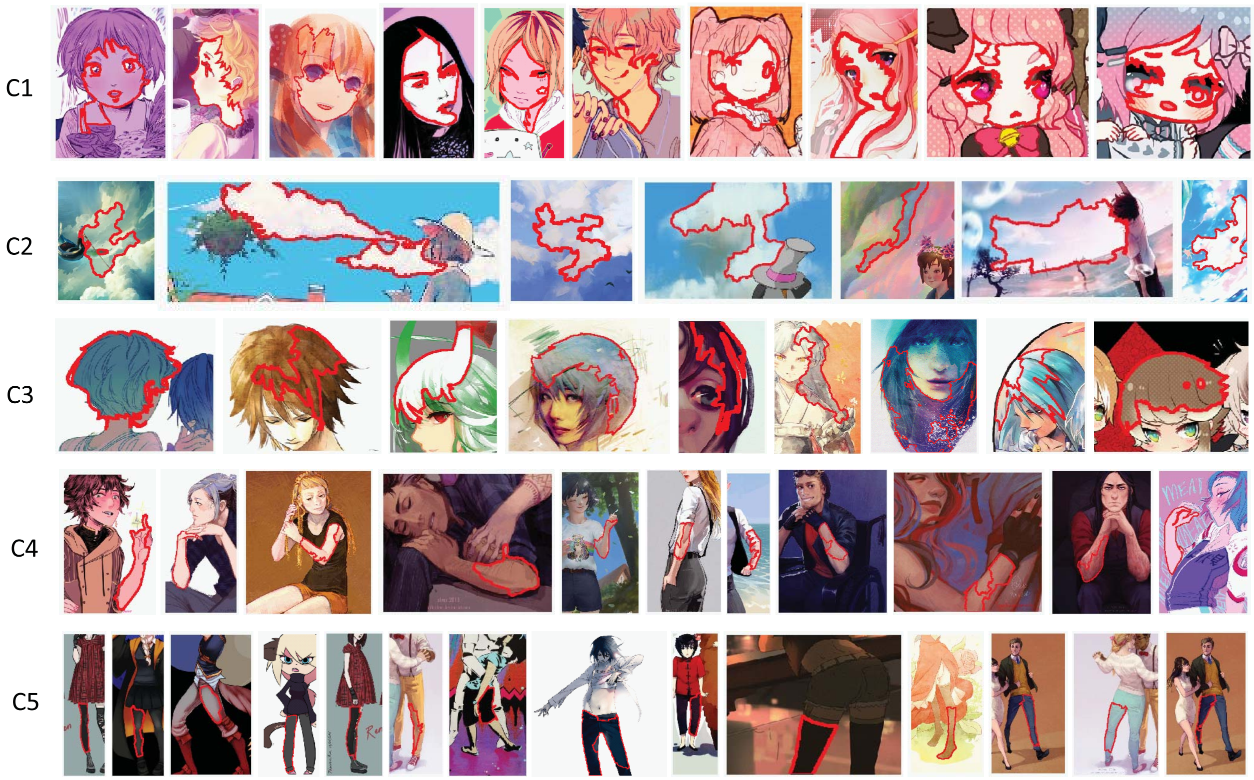

Figure 1: Clusters of image regions in latent semantic space. Each row shows several image regions (outlined in red) from one cluster. The artwork images are from DeviantArt.

the k-means algorithm to cluster all the feature vectors into $L-1$ foreground semantic classes (where L is set to be 60 empirically). Note that we only use the method above to label the regions whose normalized areas w.r.t. their respective artworks are larger than 0.01, since the very small regions contain little information for obtaining reliable semantic features. After the labeling is completed, we propagate the labels from the labeled regions to the small unlabeled regions based on spatial proximity. Finally, each color region is assigned to one of the L latent semantic classes ($L-1$ foreground classes and 1 background class). In Figure 1, we show several semantic clusters of image regions obtained by our semantic labeling step. Note that each cluster represents a semantic concept, such as face (C1), cloud (C2), hair (C3), arm (C4) and leg (C5).

3 Learning Probabilistic Colorization Model

To train our model, we adopt a sampling-based method as in [GS04]. The basic idea is to draw samples from posterior distribution $p(\mathbf{t}|\mathbf{c}, \mathbf{l}, \mathbf{f}, \mathbf{r})$, and use the samples to estimate the parameters. In particular, we adopt Gibbs sampling, which is an instance of Markov Chain Monte Carlo (MCMC) method. Gibbs sampling updates each variable $t_{m,n}$ in turn by sampling the conditional distribution of $t_{m,n}$, assuming that the values of all other variables are fixed. We are interested in applying Gibbs sampling to compute a posterior probability of a palette assignment $t_{m,n}$ for region (m, n)

given all other palette assignments up to a constant factor:

$$p(t_{m,n}=t|\mathbf{t}_{\backslash m,n},c_{m,n}=c,\mathbf{c}_{\backslash m,n},l_{m,n}=l,\mathbf{l}_{\backslash m,n},\mathbf{f}_{m,n},r_n=r)\propto \left(\frac{N_{t,r}^{\backslash m,n}+\alpha}{\sum_{t'=1}^{T}N_{t',r}^{\backslash m,n}+T\alpha}\right)\left(\frac{N_{t,c}^{\backslash m,n}+\beta}{\sum_{c'=1}^{K}N_{t,c'}^{\backslash m,n}+K\beta}\right)\left(\frac{N_{t,l}^{\backslash m,n}+\gamma}{\sum_{l'=1}^{L}N_{t,l'}^{\backslash m,n}+L\gamma}\right)\mathcal{N}(\widehat{\boldsymbol{\mu}}_t^{\backslash m,n},\widehat{\mathbf{\Sigma}}_t^{\backslash m,n}), \tag{1}$$

where $\mathbf{t}_{\backslash m,n}$ refers to all palette assignments except $t_{m,n}$. $N_{t,r}^{\backslash m,n}$ is the number of times that palette t is used by artist r excluding the contribution of region (m,n). $N_{t,c}^{\backslash m,n}$ is the number of times that color c is assigned to palette t excluding the contribution of region (m,n). $N_{t,l}^{\backslash m,n}$ is the number of times that semantic label l is assigned to palette t excluding the contribution of region (m,n). $\widehat{\boldsymbol{\mu}}_t^{\backslash m,n}$ and $\widehat{\mathbf{\Sigma}}_t^{\backslash m,n}$ are the mean and covariance, respectively, of the spatial features of all the regions associated with palette r excluding the contribution of region (m,n).

To sample from the posterior distribution $p(\mathbf{t}|\mathbf{c},\mathbf{l},\mathbf{f},\mathbf{r})$, we initialize $\mathbf{t}$ randomly, and iteratively update each variable in $\mathbf{t}$ by sampling Eq. 1. After a sufficient number of iterations (over all of the variables in $\mathbf{t}$), we take the most recent samples $\mathbf{t}^*$ and use them to estimate the parameters as follows [GS04]:

$$\begin{aligned}
\Phi_{r,t} &= \frac{N_{r,t}+\alpha}{\sum_{t^*}^{T}N_{r,t^*}+T\alpha},\\
\lambda_{t,c} &= \frac{N_{t,c}+\beta}{\sum_{c^*}^{K}N_{t,c^*}+K\beta},\\
\theta_{t,l} &= \frac{N_{t,l}+\beta}{\sum_{l^*}^{L}N_{t,l^*}+L\gamma},\\
\boldsymbol{\mu}_t &= \frac{\sum_i^{N^t}\boldsymbol{f}_{t,i}}{N^t},\\
\mathbf{\Sigma}_t &= \frac{\sum_i^{N^t}(\boldsymbol{f}_{t,i}-\boldsymbol{\mu}_t)^T(\boldsymbol{f}_{t,i}-\boldsymbol{\mu}_t)}{N^t},
\end{aligned} \tag{2}$$

where $N_{r,t}$ is the number of times that palette t is used by artist r in $\mathbf{t}^*$. $N_{t,c}$ is the number of times that palette t is assigned to color c in $\mathbf{t}^*$. $\boldsymbol{f}_{t,i}$ is the i-th spatial feature vector belonging to palette t. N^t is the number of spatial feature vectors belonging to palette t.

4 Learned Probabilistic Color Palettes Organized by Artist

Some examples of the learned probabilistic color palettes organized by artist are shown in Figure 2.

5 Comparison with other Representations

Figure 3 shows two examples that compare our probabilistic color palettes (Ours) with other representations, i.e., global color histogram (Global) and spatial-binning color histogram with $K = 2$ (Spatial 2×2) and $K = 3$ (Spatial 3×3).

References

[EKSX96] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96*, 1996.

[GS04] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004.

[MFM04] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE TPAMI*, 26(1), 2004.

[SM15] M. Saito and Y. Matsui. Illustration2vec: A semantic vector representation of illustrations. In *Proc. SIGGRAPH Asia Technical Briefs*, 2015.

[VS08] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *ECCV'08*, 2008.

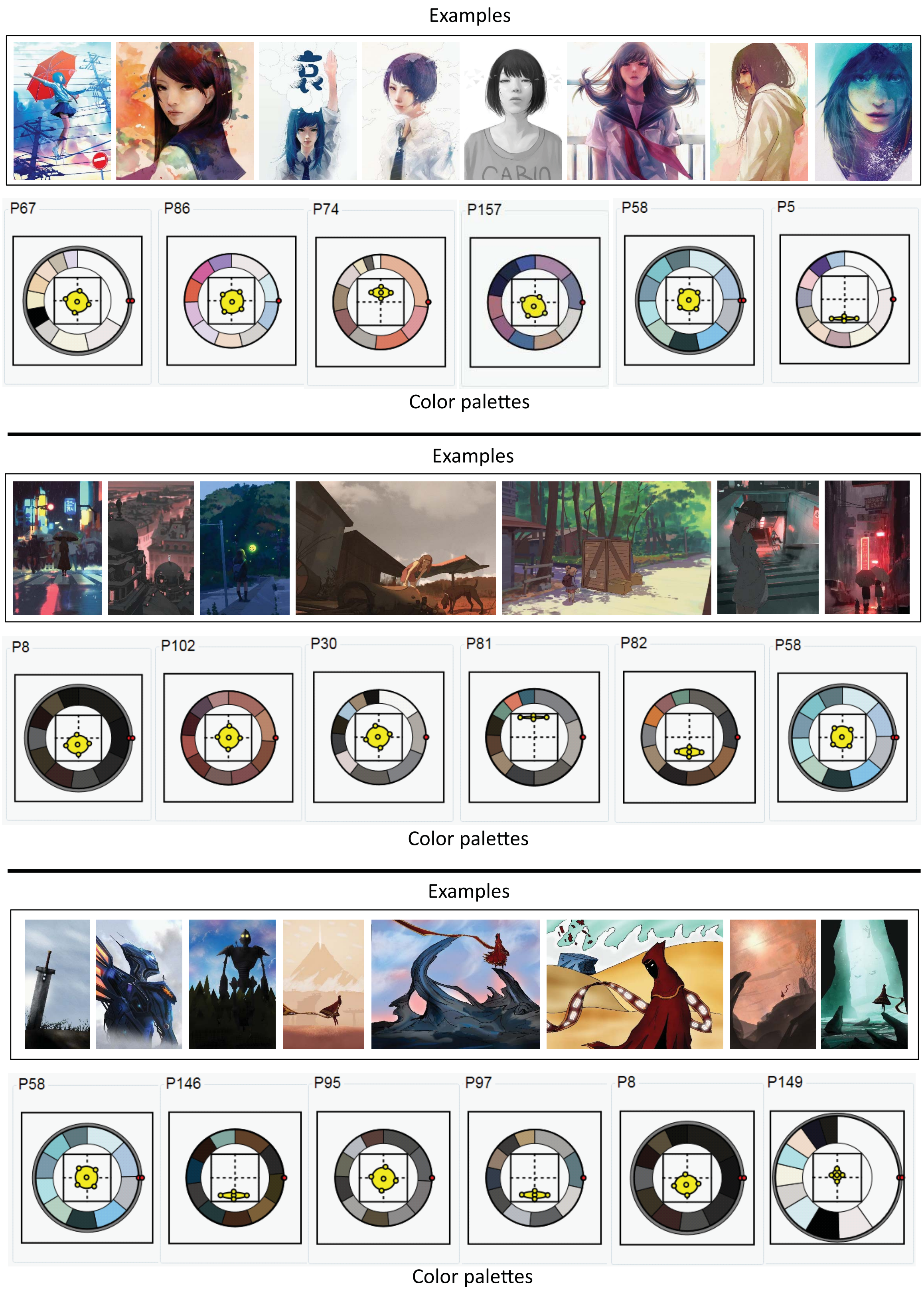

Figure 2: learned probabilistic color palettes. Each row shows some examples from the gallery of an artist and the 6 most frequently used color palettes. The artwork images at the first, second and third rows are by wataboku, snatti89 and TacoSauceNinja on DeviantArt, respectively.

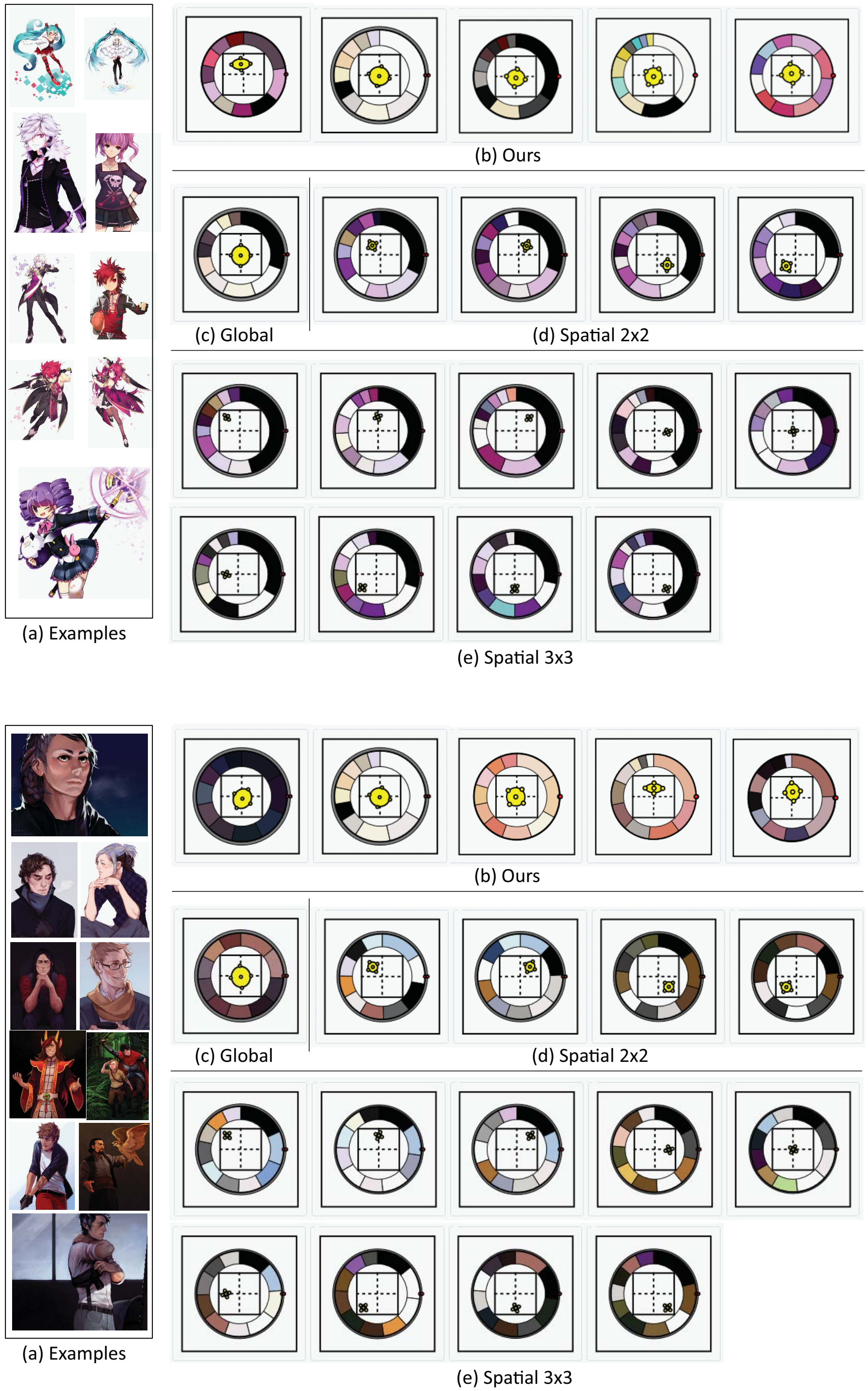

Figure 3: Comparison of our probabilistic color palettes with other representations in summarizing the color use in an artwork gallery. (a) shows some examples from the gallery. The artwork images at the first and second rows are by LittleDiety and littleulvar on DeviantArt, respectively.