CrowdRetouch: An At-once Image Retouch System Applying Retouching Parameter Visualization

Yuri Saito
Department of Information Sciences
Ochanomizu University
Tokyo, Japan
yuri@itolab.is.ocha.ac.jp

Takayuki Itoh
Department of Information Sciences
Ochanomizu University
Tokyo, Japan
itot@is.ocha.ac.jp

***Abstract*— This paper proposes a new image retouch system "CrowdRetouch" which reflects users' tendency of image retouch for a set of similar photos. CrowdRetouch firstly asks initial users to manually retouch sample training images and then divides the initial users based on the image retouch parameters. It then applies regression analysis to each of user clusters to solve the relationship between the retouch parameters and image features and automatically retouches rest of similar photos based on the regression analysis results. After forming the user clusters, CrowdRetouch specifies the clusters of new users with a smaller number of training images by visualizing the learning processes, and therefore we do not need to require heavy preprocesses to the new users. CrowdRetouch realizes personalized automatic image retouching to a large number of photos while reflecting preferences of novice users. This paper introduces our user experiments which demonstrate the parameter visualization is effective for appropriate learning of users' preferences.**

Keywords*— *Image retouch parameter, Image feature, Regression Analysis, User clustering

I. Introduction

We often show a set of photos on social networking services (SNS) or digital photo frames after retouching at once. Recent photo applications feature automatic retouch services. These services are convenient to retouch variety of photos so that many users feel appropriate. Also, filter functions provided by the services make us easy to apply various effects. However, most of these commercial services do not support reflecting users' preference to the retouching results. Instead, many of the services allow users to select a variety of filters. Even, it may be often difficult for users to select appropriate filters.

Meanwhile, tone-curve-based photo retouching supported by image processing software packages has an advantage that users' preference can be finely reflected by adjusting various parameters such as brightness and hue. However, it is a too complicated and time-consuming task for novice users to adjust many parameters appropriately. Also, it may take a long time to manually retouch many photos one by one even a user retouches many similar photos like dish photos taken at the same dining every day.

This paper presents "CrowdRetouch," which retouches a set of similarly taken photos based on users' tendency of photo retouch. Given sample photos grouped according to categories of scenes, we ask initial users to retouch the sample photos based on their preferences manually. CrowdRetouch applies a clustering to the users based on the retouching parameters, and a regression scheme to solve the relationship between image features and retouching parameters.

When a new user starts using CrowdRetouch, we ask him/her to retouch a small number of sample photos so that we can specify which cluster is appropriate for him/her. Specifically, CrowdRetouch presents images retouched applying parameters estimated by regression analysis with each of user cluster (referred to as a filter). And then, new user retouch sample training image by blending each filter. When a new user retouches sample training images, CrowdRetouch displays the blend ratio for a sample training image as a single polygonal line in the parallel coordinate plots. As a result, the new user can observe the tendency of their favorite filter preference from the visualization of the filter blending ratio. Users can realize at-once photo retouch based on their preferences, even if they are novice users who are not familiar with adjustment of image retouch parameters.

We suppose CrowdRetouch is especially effective for sets of photos taking similar scenes. Users may take photos of cuisines every day if they like cooking. Or, they may take a large number of photos of favorite scenes such as their campus, cherry blossoms, falling leaves, and mountains. CrowdRetouch is especially effective for at-once retouch with such types of sets of similar photos.

II. Related Work

CrowdRetouch is an image retouch system which reflects users' tendency of image retouch for a set of similar photos. This mechanism is relevant to interactive parameter optimization techniques. This section introduces such techniques applying crowdsourcing and human-computation.

A. *Crowd-Powered Analysis*

Crowdsourcing is technique whereby we can stably employ a large number of temporary participants through the Internet. The collected data by crowdsourcing can be used directly or reused as training data of machine learning systems in recent research. Jaroensri et al. [1] presented a method to predict goodness of retouching results by using crowdsourcing data. Zhu et al. [2] applied a crowdsourcing technique to evaluate portrait photos.

Goals of these studies were to find the parameter set which satisfies many people; they did not aim to reflect preferences of particular users.

B. *Human Computation Analysis*

Human computation makes use of human abilities for computation to solve problems [3]. This section introduces studies on human computation applied to image retouching.

Marks et al. proposed Design Galleries [4]. This system first shows randomly generated designs to the user, supposing users repeatedly perform the task of selecting their suitable while adjusting design parameters. Shapira et al. proposed a method which presents a candidate image to a user and adjusts hue or saturation using a mixed Gaussian distribution [5]. Images presented by these systems are not always intended by the user. Users can still adjust the parameters according to the users' preferences on these systems; however, it is a time-consuming task to adjust parameters for a large number of contents independently.

Recent machine learning techniques have contributed to learning preferences of users' photo retouching. The learning results can be applied to retouch a large number of photos at once. Kapoor et al. [6] presented a technique which consumes users' evaluation of automatically retouched sample images so that it can reflect their preferences on automatic image retouching. Selph presented by Koyama et al. [7] estimates users' preference during their manual retouch parameter adjustments. Bychkovsky et al. [8] presented another approach which learns users' preferences from sample retouching results by professional photographers. Zhicheng [9] presented a technique which apply neural networks to predict retouching parameters. CrowdRetouch is somewhat similar to this approach because it also learns users' preferences from retouching results of initial users; however, we suppose a crowd of initial users, not limited to professional but also ordinary users, to form clusters of users and apply regression analysis to them. After forming the user clusters, CrowdRetouch specifies the clusters of new users with a smaller number of training images while visualizing the learning processes. Therefore, we do not need to require heavy preprocesses to the new users.

III. TECHNICAL DETAIL OF CROWDRETOUCH

Following is the processing flow of CrowdRetouch, consisting of training and use phases. Following sections describe technical details of each component.

1. Training phase
   1-1. Photo categorization and clustering
   1-2. Manual retouch of sample training images
   1-3. User clustering
   1-4. Regression analysis
   1-5. Discriminant analysis
2. Use phase
   2-1. Cluster specification for new photos and users
   2-2. Retouch of new input photos

A. *Photo categorization and clustering*

CrowdRetouch firstly divides prepared sample training images based on their categories of scenes. We suppose brief keywords (e.g. "landscape", "food", and "nature") are assigned to the sample training images. As a preprocessing, we calculate image features of sample training images and apply a clustering algorithm for each category of photos. This process calculates image features and applies a K-means algorithm. Our current implementation simply calculates the following features and treats them as 18-dimensional vectors:

- Average, mean, mode, and standard deviation of R, G, B, and intensity
- Maximum and minimum number of pixels of sub-regions divided based on pixel values.

B. *Manual retouch of sample training images*

We ask initial users to manually retouch the sample training images, and save the retouch parameters. Figure 1 shows CrowdRetouch GUI for initial users.

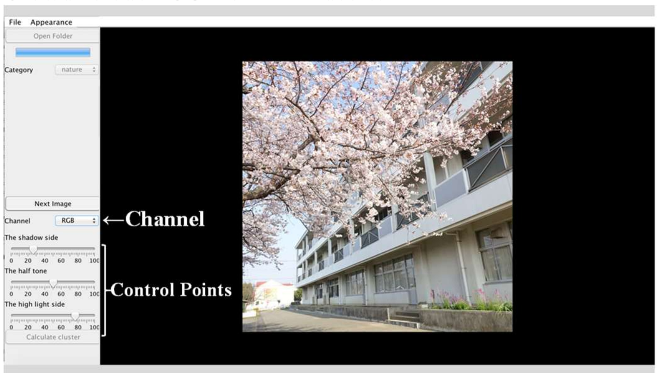

Figure 1 CrowdRetouch GUI for initial users.

Our current implementation features manipulation of tone curves of R, G, B, and RGB channels while retouching. Users can manually adjust positions of three control points, Cs (pixel value 64), Cm (128), and Ch (192) as shown in Figure 2. Consequently, 12 parameters are specified for each photo.

C. *User clustering*

We divide the initial users based on the retouching results by applying a clustering algorithm. Let the number of sample training images as n_S, and the number of retouch parameters as n_P (12 in our implementation). We treat the retouching results of a user as a ($n_S n_P$)-dimensional vector. CrowdRetouch applies principal component analysis to reduce the dimensions and then applies dendrogram (ward) method to divide the users. These user clusters are generated for each photo cluster. Figure 3 shows the tree structure of photo and user clusters. Let the i-th user cluster of the j-th photo cluster as C_{ij} in the below sections.

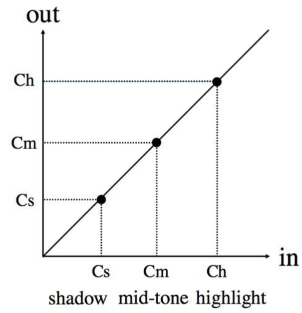

Figure 2 Tone Curve.

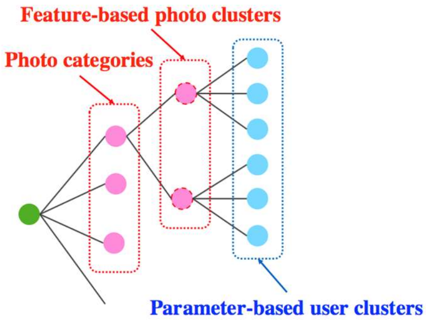

Figure 3 Tree structure of image and user cluster.

D. Regression analysis

CrowdRetouch applies linear multiple regression analysis for each user cluster. The linear multiple regression analysis is a statistical method for predicting one dependent variable r with two or more independent variables x_i ($i \geqq 2$). It is processed using the following equation (1), where a is a partial regression coefficient and b is a regression coefficient.

$$r = b_1x_1 + b_2x_2 + \ldots b_nx_n + a \quad (1)$$

This result is used to specify the retouch parameters for automatic retouch of new photos. Let the image features as X= ($x_1, x_2, \ldots, x_{nF}$). We solve the function $r_k = f_k$(X), where r_k denotes the k-th retouch parameter.

This process is applied to all user clusters C_{ij}. Let f_{ijk}(X) as the function to calculate the k-th retouch parameter for the uses belonging to the i-th user cluster and photos belonging to the j-th photo cluster.

We often fail the regression analysis when the number of dimensions is too high. To solve this problem, we calculate Pearson correlation coefficients for each pair of explanatory variables (image features in this case) and objective functions (retouch parameters in this case). We eliminate them from the regression analysis if they have no strong correlations. Or, it may also be problematic if multiple explanatory variables have too strong correlations. In this case, our implementation eliminates one of such variables from the regression analysis. Figure 4 shows the processing process.

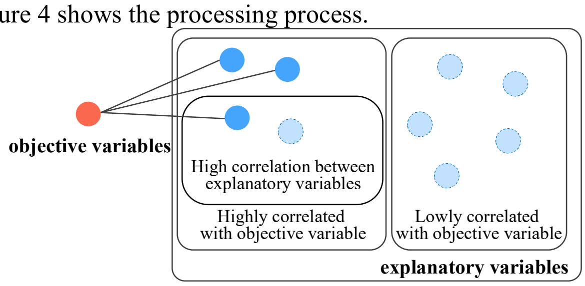

Figure 4 Relation between explanatory variable and objective variables.

E. Discriminant analysis

When new photos are given, CrowdRetouch determines which photo cluster each of new photos belongs. First, it specifies the category for each of new photos based on brief keywords (e.g. “landscape,” “food,” and “nature”). And furthermore, it applies a support vector machine for these categories. Therefore, as part of "training phase," sample photos are input to the support vector machine. More specifically, multidimensional variables (image feature vector) and labels (belonging image cluster number) relating to sample photos classified by the keyword are input as a training dataset.

In the current implementation, LIBSVM [10] is used for classification by support vector machines. Appling the linear methods such as support vector machines, the kernel function is needed to map samples into a higher dimensional space. The RBF kernel is adopted as the kernel function defined as equation (2), where γ is a kernel parameter. The defaults value of LIBSVM is used for the γ value and the C value. Although this process is one type of multi-class classification, it implements "one-against-one" approach.

$$K(x_i, x_j) = exp(-\gamma \, ||x_i - x_j||^2), \; \gamma > 0. \quad (2)$$

F. Cluster specification for new photos and users

CrowdRetouch can automatically retouch new photos when we finish the training phase. CrowdRetouch determines which photo cluster each of the input photos belongs, and which user cluster the current user belongs. It then applies the retouch parameters brought by the regression analysis results so that we can automatically retouch the photos based on users’ preference. In this process, CrowdRetouch estimates which photo category new photo belongs. First, it specifies the category of a new photo based on brief keywords (e.g. “landscape,” “food,” and “nature”). As a preprocessing, it calculates image features of the new photos and predicts their image clusters by the image features by applying Support Vector Machine (SVM). Then, CrowdRetouch determines which user cluster the new user belongs. We ask the new user to retouch the sample images prepared for the training phase. Figure 5 shows the GUI screen for this task.

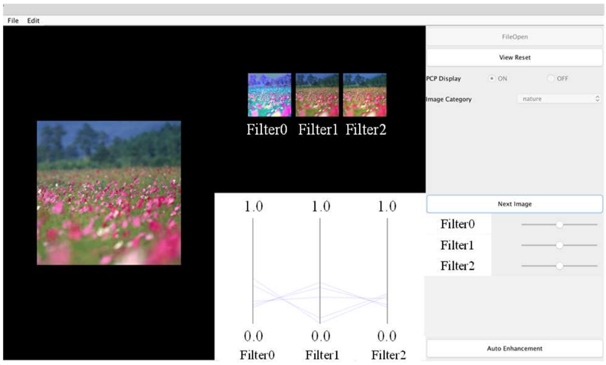

Figure 5 CrowdRetouch GUI for new users.

[Left side]
Manual retouching result of a sample training image.
[Upper-center side]
Automatically retouched photos applying parameters estimated by the regression analysis for three user clusters.
[Lower-center side]
Parallel coordinate plots which visualizes the ratio of blending the parameters of the three user clusters.
[Right side]
Control panel for retouching photos.

This GUI asks a new user to retouch sample photos by blending parameters of user clusters. The upper-center side of Figure 5 shows three photos retouched with parameters of three user clusters. The new user is required to manipulate the ratio of the three photos to blend them as displayed in the left side of Figure 5. This task identifies which user cluster the new user should belong. A smaller number of sample photos comparing with initial users are required to identify the cluster for a new user. Here, appropriateness of the cluster identification strongly depends on the selection of sample photos. We select appropriate sets of sample photos by applying Davies-Boulding Index (DB Index). Let a set of retouch parameters applied to the i-th user as $\boldsymbol{a}_i$. The similarity of retouch parameters between the i-th and j-th users in the same cluster is described as $ave|\boldsymbol{a}_i - \boldsymbol{a}_j|$. We can determine that similar parameter values are applied to the photos in a cluster if $ave|\boldsymbol{a}_i - \boldsymbol{a}_j|$ is smaller. Meanwhile, we can calculate the average retouch parameter of the k-th cluster, which the i-th user belongs, as $\boldsymbol{c}_k$. A distance between the k-th and l-th cluster is therefore described as $ave|\boldsymbol{c}_k - \boldsymbol{c}_l|$. From the definition of DB Index, we can determine that the clustering result for an arbitrary photo is preferable if the value $ave|\boldsymbol{a}_i - \boldsymbol{a}_j|/ave|\boldsymbol{c}_k - \boldsymbol{c}_l|$ is smaller. So, we calculated this value to all of the prepared sample photos and generated the set of a small number of sample photos those values are lower than the pre-defined threshold.

When a new user retouches sample training images, CrowdRetouch displays the blending ratio for a sample training image as a single polygonal line in the parallel coordinate plots as shown in Figure 5. As a result, the new user can observe the tendency of blending ratio which brings their favorite retouching results by looking at the parallel coordinate plots. The system calculates the average ratios for each of user clusters based on these training results. Finally, we identify the user cluster for the new user as the one which has the largest average ratio value.

G. Retouch of new input photos

Finally, CrowdRetouch realizes at-once photo retouch on their preferences by regression result of the corresponding cluster. Although if there is a large number of photos, we expect CrowdRetouch realizes personalized automatic retouching while reflecting preferences of novice users.

IV. EXPERIMENT

This section introduces example results and user experiments of CrowdRetouch.

A. Running environment

We implemented CrowdRetouch with Java JDK 1.7.0 and tested on MacBook Pro. We tested CrowdRetouch with photos published on ImageNet and our own photos as training photos. We assigned one of the following categories, "food," "landscape," "nature" to the photos.

B. User experiment

We had a user experiment with 26 university students majoring computer science. They were divided into the group of 16 initial users and 10 new users beforehand for the experiments. Following are the tasks of the experiments.

1) Clustering with initial users

First, we conducted the following tasks with 16 initial users.

[Step 1-1] We asked the initial users to retouch 30 training images for each photo category with GUI (Figure1). We then applied user clustering and regression analysis.

[Step 1-2] CrowdRetouch retouched ten new photos applying the parameters estimated with the regression analysis results.

We showed the following images and asked the initial users to evaluate comparatively.

A) Original input photos.
B) Photos automatically retouched by commercial software (Picasa in this experiment).
C) Photos retouched applying parameters estimated by regression analysis without photo categories, photo clustering, and user clustering.
D) Photos retouched applying parameters estimated by regression analysis, with photo categories, without photo clustering and user clustering.

E) Photos retouched applying parameters estimated by regression analysis, with photo clustering, without photo categories and user clustering.

F) Photos retouched applying parameters estimated by regression analysis, with photo categories and user clustering, without photo clustering.

G) Photos retouched applying parameters estimated by regression analysis, with photo categories, photo clustering, and user clustering.

2) Evaluation of cluster detection by the new users

Second, we conducted the following task with ten new users.

[Step 2-1] We asked new users to retouch 20 training images for each photo category with GUI for new users (Figure 5). The new users retouched sample training image by blending parameters.

[Step 2-2] CrowdRetouch calculated the average of blending ratios from the result of Step 2-1 and identified the user cluster which has the highest average blending ratio as the corresponding cluster of a new user.

[Step 2-3] CrowdRetouch retouched new images applying parameters estimated by a regression analysis of the appropriate cluster.

[Step 2-4] We showed the retouched images by CrowdRetouch and asked the users to evaluate comparatively.

We conducted experiments with or without displaying the parallel coordinate plots of blending ratios to evaluate its effectiveness. Then, CrowdRetouch specified the appropriate cluster for each of new users and retouched new photos applying parameters when the users finished retouching 6, 9, and 12 sample training images. We verified the desirable number of sample training images which brings appropriate identification of user clusters.

C. Experimental results

1) Effectiveness of image categories by keywords

We conducted the following evaluation task with the initial users. First, we randomly selected ten photos from each of A), C) and D) without considering photo category. We asked the initial users to select the photo which most closely matches their preferences. Figure 6 shows the statistics of their choices.

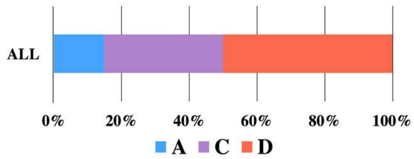

Figure 6 Effectiveness of image categories by keywords.

Comparing original photos A), retouched photos C) and D) received higher evaluations. Also, D) had higher evaluations comparing with C). This result suggests that preferable image retouching parameters are different for each photo category.

Next, we randomly selected photos from each of A), D) and E) without considering photo category.

We asked the initial users to select the photo which most closely matches their preferences. Figure 7 shows the statistics of their choices.

D) had higher evaluations comparing with original photos A) and retouched photos E). This result suggests that keyword-based photo classification is more effective and important comparing with feature-based classification.

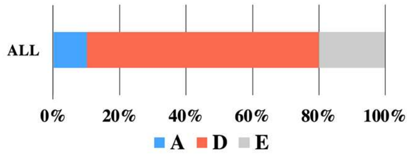

Figure 7 Comparison of effectiveness of clustering by image categories and image features.

2) Effectiveness of user clustering

We conducted the following evaluation task with the initial users. First, we randomly selected ten photos from each of A), D), F) without considering photo category. We asked the initial users to select the photo which most closely matches their preferences. Figure 8 shows the statistics of their choices.

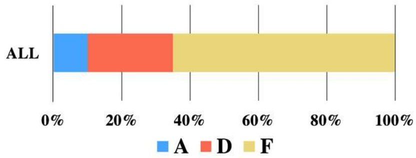

Figure 8 Effectiveness of user clustering.

Comparing with original photos A), D) and F) received higher evaluations. Also, F) had higher evaluations comparing with D). This result suggests that user clustering improves satisfactory.

Next, we selected photos D) and F) from each user cluster while considering photo category. We asked the initial users to select the photo which most closely matches their preferences from D) and F) for each user cluster. Then, we calculated the probability of selecting the photo which applied parameters estimated by a regression analysis of user cluster which the user belongs. As a result of examination using chi-square test, $x^2 = 5.556$, $p = 0.018$, and statistical significance was observed in the photo selection result by $p < 0.05$.

3) Effectiveness of image cluster by image feature

We conducted the following evaluation task with the initial users. First, we selected ten photos from each of A), B), F), G) with considering photo category. We asked the initial users to select the photo which most closely matches their preferences. Figure 9 shows the result in the photo category "nature".

Comparing A) and B), F) and G) had higher evaluations. Also, G) had higher evaluations comparing with F). Results with other image categories were similar to the above result.

We discussed which types of photos in F) and G) received higher evaluations. We found that F) received higher evaluations if a photo contains various colors. On the other hand, G) was preferable if similar colors occupy a significant portion of a photo. In other words, feature-based image clustering is effective especially for the photos which similar colors occupy a significant portion. Figure 10(left) shows example photos where F) was preferable, while Figure 10(right) shows examples where G) was better.

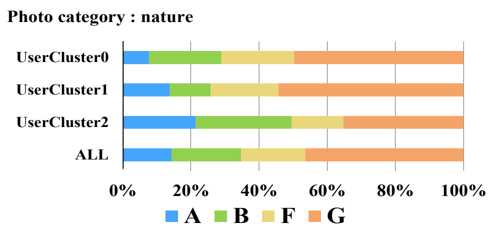

Figure 9 Effectiveness of clustering by image features.

Figure 10 Example photos

4) *Preference tendency for each user cluster*

We observed the differences in preferences in each user cluster. Figure 11 shows examples of highly evaluated retouches of each user cluster.

Here, retouching result for UserCluster 0 was relatively contrastive since the RGB tone curve was convex downward. Users in UserCluster 1 preferred relatively warmer tones as well as contrastive retouches since G tone curve was convex downward in addition to RGB tone curve. Retouched photos for UserCluster 2 were more reddish since R tone curve was more convex upward rather than tone curves of other user clusters. These results suggest that users in each of clusters have different preferences of photo retouch. They also suggest that CrowdRetouch retouches photos with different parameters brought by different regression analysis results for each of user clusters.

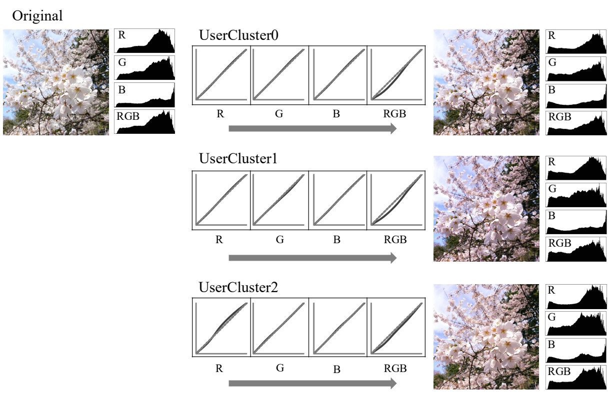

Figure 11 Example of users' preference.

5) *Cluster accuracy of the initial user*

We observed DBIndex values to verify how the appropriateness of user clustering results improved as the number of sample training images increased as explained Ⅲ chapter F. section. We specified the number of clusters so that the value of DBIndex values gets the best. Figure 12 shows the DBIndex values and the dendrogram changing the number of sample training images to 10, 20, 30 in each photo category. We found that the appropriateness of initial user clustering improves as the number of sample training image increases in all image categories.

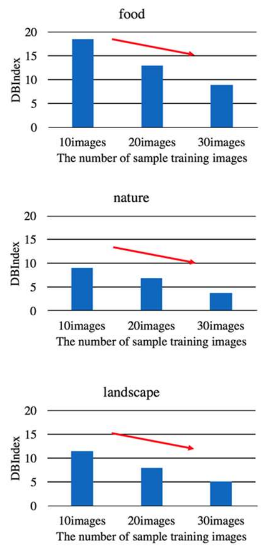

Figure 12 The accuracy of the clustering for initial user and the result of clustering by dendrogram.

6) *Effect of displaying visualization result of blend ratio*

We conducted the following task with new users. We tested with or without displaying the parallel coordinate plots of blending ratios as described in Section 4.2.2. Then, we specified user clusters for each of new users and retouched photos by applying parameters of appropriate user clusters. We asked new users to select the photo which most closely matches their preferences. The results shown in Figure 13 suggest that displaying the visualization of blending ratio was effective. As a result of examination using chi-square test, $x^2 = 3.9729$, $p = 0.0462$, and statistical significance was observed in displaying visualization by $p < 0.05$.

Also, we asked new users to blend nine sample training images with or without displaying the parallel coordinate plots of blending ratios. Then, we calculated the averages of the total operation time for retouching. The result shown in Figure 14 demonstrates the parallel coordinate plots contributed to reduce the total operation time. After the experiments, we received positive comments about displaying the parallel coordinate plots as follows.

- The visualization made easy to operate.
- I understood the tendency of my blending ratios.

On the other hand, we received just one negative comment that visualized past blending ratios dragged his/her next blending operations unconsciously.

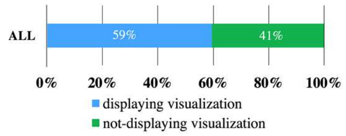

Figure 13 Effect of displaying visualization result of blend ratio

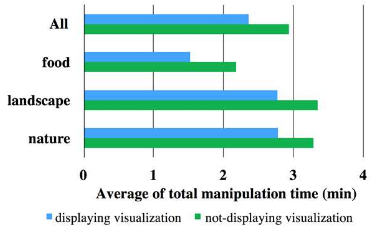

Figure 14 Average of total operation time required for retouching for sample training images.

7) Comparison of the number of sample training images of new users

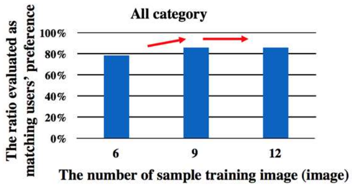

Figure 15 The ratio evaluated as matching users' preference.

We conducted the following task with new users. This task aimed to verify how many sample training images are desirable to identify user clusters for new users appropriately. We identified the user clusters when a new user finished retouching 6, 9 and 12 sample training images, and retouched photos with the parameters of the identified user clusters. Then, we asked new users to select the photo which most closely matches their preferences. Figure 16 shows the percentage of photos retouched by CrowdRetouch in the user-selected photos. This result suggests nine sample training images were necessary to identify appropriate user clusters in this task.

V. Conclusion and Future Work

This paper presented CrowdRetouch which retouches sets of photos taking similar scenes reflecting users' preferences. CrowdRetouch first generates clusters of initial users and identifies preferable retouch parameters for each user cluster applying regression analysis. It then identifies appropriate clusters of new users with the small learning process and retouches sets of photos of the new users applying the parameters of the identified cluster.

Experiments introduced in Section 4 shows that preferred retouch differs depending on the photo category and the user cluster. Especially, the proper construction of user clusters is important for appropriate photo retouch by CrowdRetouch. It may be a time-consuming task for initial users to retouch sample training images manually and construct user clusters. Instead, new users need a smaller task which blends the predefined number of retouched photos to specify the appropriate user cluster for the new users. This mechanism contributes to retouch sets of photos reflecting preferences of the new users with just a small learning task. The experiment demonstrated that visualization of blending ratios by parallel coordinate plots effectively assisted in reducing the operations of the learning tasks.

The following two issues will be raised as future tasks. First, our implementation simply applies linear regression analysis. We could not apply non-linear schemes (e.g., Gauss-kernel regression analysis) because the number of initial users was too small in our experiments. We would like to extend our experiments so that we can apply non-linear regression analysis schemes. Second, we also would like to extend retouching methods in addition to the current implementation of tone-curve-based color enhancement.

After the above improvement, we would like to re-examine the usefulness of CrowdRetouch by conducting experiments with a large number of participants.

Acknowledgment

This work has been partially supported by Japan Society of the Promotion of Science under Grant-in-Aid for Scientific Research.

References

[1] R. Jaroensri, et al., Predicting range of acceptable photoic tonal adjustments, Computational Photoy (ICCP), 2015 IEEE International Conference on. IEEE, 2015.

[2] J.Y. Zhu, et al, Mirror mirror: Crowdsourcing better portraits, ACM Transactions on Graphics (TOG) 33.6 (2014): 234, 2014.

[3] L. Von Ahn, Human computation, Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on. IEEE, 2008.

[4] J. Marks, et al., Design galleries: a general approach to setting parameters for computer graphics and animation, ACM SIGGRAPH '97, pp.389-400,

[5] L. Shapira, A. Shamir, D. Cohen-Or, Image appearance exploration by model-based navigation, Computer Graphics Forum (EUROGRAPHICS 2009), pp.629-638, 2009.

[6] A. Kapoor, J. Caicedo, D. Lischinski, S. Kang, Collaborative Personalization of Image Enhancement, International Journal of Computer Vision, Vol.108, No.1-2, pp.148-164, 2014.

[7] Y. Koyama, I. Sato, D. Sakamoto, and T. Igarashi. Sequential Line Search for Efficient Visual Design Optimization by Crowds. ACM Trans. Graph. 36, 4, pp.48:1--48:11, 2017.

[8] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, Learning Photographic Global Tonal Adjustment with a Database of Input / Output Image Pairs, IEEE Conference on Computer Vision and Pattern Recognition, pp.97-104, 2011.

[9] Y. Zhicheng, et al., Automatic photo adjustment using deep neural networks, ACM Transactions on Graphics 2016.

[10] LIVSVM
http://www.csie.ntu.edu.tw/~cjlin/libsvm/ (browsed date: 2018.3.24).