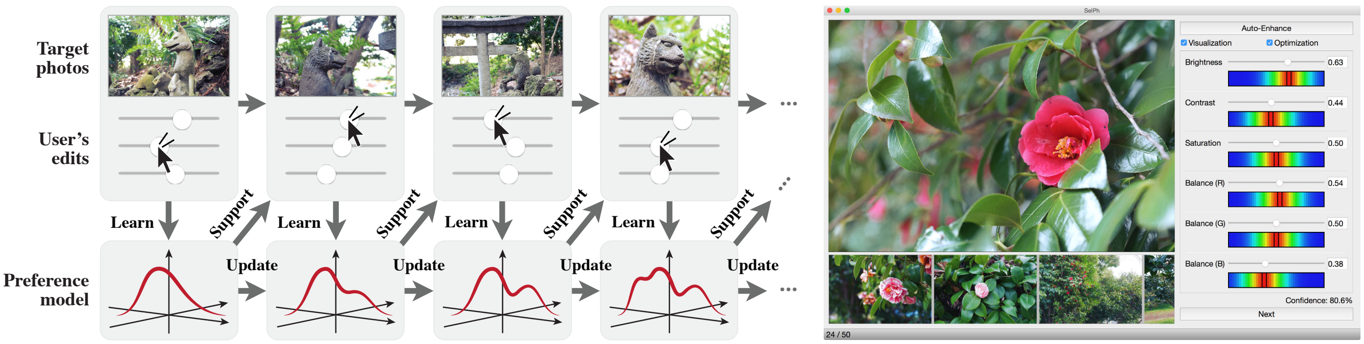
SelPh: Progressive Learning and Support of Manual Photo Color Enhancement

Yuki Koyama **Daisuke Sakamoto** **Takeo Igarashi**

The University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan

koyama@is.s.u-tokyo.ac.jp, {d.sakamoto, takeo}@acm.org



(a) (b)

Figure 1. (a) Concept of *self-reinforcing color enhancement*. As more photos are enhanced by the user, the system *implicitly* and *progressively* learns the user's preferences and, as a result, the system is able to support the user in an increasingly effective manner. (b) A working prototype system, named *SelPh*. It has several user support functions enabled by the self-reinforcement, including enhanced sliders and confidence-based adaptation.

ABSTRACT

Color enhancement is a very important aspect of photo editing. Even when photographers have tens of or hundreds of photographs, they must enhance each photo one by one by manually tweaking sliders in software such as brightness and contrast, because automatic color enhancement is not always satisfactory for them. To support this repetitive manual task, we present *self-reinforcing color enhancement*, where the system *implicitly* and *progressively* learns the user's preferences by training on their photo editing history. The more photos the user enhances, the more effectively the system supports the user. We present a working prototype system called *SelPh*, and then describe the algorithms used to perform the self-reinforcement. We conduct a user study to investigate how photographers would use a self-reinforcing system to enhance a collection of photos. The results indicate that the participants were satisfied with the proposed system and strongly agreed that the self-reinforcing approach is preferable to the traditional workflow.

ACM Classification Keywords

H.5.m Information Interfaces and Presentation (e.g. HCI): Miscellaneous

DOI: http://dx.doi.org/10.1145/2858036.2858111.

Author Keywords

Design support; photo enhancement; self-reinforcement

INTRODUCTION

When photographers enhance the color of a photo, they need to manually adjust many sliders such as brightness, contrast, and saturation. In Adobe Photoshop Lightroom CC, there are eleven "Basic" sliders, and Adobe Photoshop CC provides several dialog widgets each of which has multiple sliders. For skilled photographers, this approach is satisfactory when they have only a few photographs. However, if they have tens of or hundreds of photos to edit, manually adjusting every photograph independently would be an onerous task. This scenario often arises when, for example, a photographer returns from a long journey with a camera and wants to upload a photo album to a website.

One possible solution to avoid this tedious task is to use fully-automatic color enhancement (*auto-enhancement*) to batch the whole process. Commercial software often provides an option to perform this. However, there are several limitations associated with this solution. First, the auto-enhancement functions in recent commercial software do not satisfy all users, because every person has different preferences [21]. Second, even if personalized auto-enhancement (*e.g.*, [20, 21]) is available, there are still some non-negligible reasons why photographers find this fully-automatic solution unsatisfactory. For example, if the goal is to edit the photos so that they are consistent with a specific design concept, such personalization would not be useful, since the desired enhancement is more scenario-dependent than personal-preference-dependent. Likewise, the context in which the photograph was taken or is to be presented may influence the editing process. Even state-of-the-art auto-enhancement algorithms cannot satisfactorily edit some types of photos, such as ones where highly semantic aesthetics are involved. Moreover, photographers might prefer exploring other possible enhancements by tweaking parameters by themselves, rather than blindly trusting the auto-enhancement, even if the auto-enhanced photograph appears satisfactory.

In this paper, we investigate a way of supporting manually repetitive enhancement, rather than fully automating the task. The goal is for users to subjectively assess every enhanced photograph as being optimized; to determine whether this goal is met, the user has to view all of the photos and make independent decisions on each one. To facilitate this, we present *self-reinforcing color enhancement*, where the system *implicitly* and *progressively* learns the user's preferences and current intent. As the system learns the user's preferences, it supports color enhancement more effectively (Figure 1 (a)). In contrast to most machine learning-based approaches [20, 5, 6, 21], in our workflow the user does not need to consider the training of the system as a separate preparation process. Instead, the user enhances photographs mostly as usual, but with help from the self-reinforcing system.

We present a prototype system, named *SelPh* (Figure 1 (b)). The ability of the system to perform self-reinforcement enables it to provide several useful support functions to the user. By using our system, we conducted a user study to investigate how photographers enhance a collection of photos (*e.g.*, a photo album) with a self-reinforcing system, how effectively the self-reinforcement approach works, and the overall level of satisfaction with the system. This paper reports insights from the user study; for example, all the participants agreed that the workflow with a self-reinforcing system is preferable to the traditional one, and all the participants found the support functions of SelPh to be satisfactory. Participants particularly liked the visualization of confidence of preference estimation, saying that it makes the system more trustworthy and enjoyable. This paper also discusses the design implications revealed by the study.

In addition to investigating self-reinforcing color enhancement to aid repetitive manual enhancement, the following three more specific contributions are detailed in this paper:

System Design. We designed a prototype system, SelPh, which offers five user support functions, including enhanced sliders and adaptation based on confidence of preference estimation.

Algorithms. To enable these functions, preference learning and interaction techniques are combined into a system; a new joint-space formulation is introduced, as well as a non-trivial combination of machine-learning techniques.

User Study. We conducted a qualitative user study by using SelPh and obtained various implications for designing learning-based systems in general. To the best of our knowledge, this is the first study that investigates how photographers enhance photos with a self-reinforcing system.

RELATED WORK

Parameter Adjustment Interface

Photo color enhancement can be considered as a visual design task that involves parameter adjustment. As a general method that is applicable to various parameter adjustment tasks, Marks *et al.* [32] presented a gallery-based interface called Design Galleries. While this interface potentially works for color enhancement tasks as well, our focus is on more direct manipulations of the parameters, as many popular packages offer. Side Views [43] is a mechanism for open-ended tasks that guides users by showing design previews along GUI widgets. Their concept is orthogonal to ours; therefore, it can be integrated into our system.

Koyama *et al.* [23] proposed a new slider interface for visual parameter adjustment, called VisOpt Slider. They demonstrated that this interface could be applied to color enhancement. However, their entire method is impractical for our target case, because it relies on crowd-powered analysis of individual photos and hence lacks generalizability. Although the VisOpt Slider interface is used for one of the support functions in SelPh, the underlying algorithms are completely different in terms of generalizability (*i.e.*, our system is able to deal with new photos based on previous edits) and training data collection (*i.e.*, crowdsourcing *vs.* self-reinforcing).

Manual Photo Color Enhancement

Photographs can be enhanced either by interactive methods or automatic methods. Shapira *et al.* [40] presented an interactive method for recoloring, which considers spatial conditions (2D distributions) of colored pixels in addition to the colors themselves, which enables complex color manipulations. Histomage [9] also provides interactive tools for color enhancement, which enable the user to easily and efficiently select spatially varying pixels. In contrast to these approaches, our current method does not take such complex spatial conditions into account; rather, our interest is the scenario where a large number of photographs require enhancement.

The function provided in Adobe Photoshop Elements called Auto Smart Tone [1] is conceptually related to our approach. Although it is named "auto," the user is expected to manually fix the suggested auto-enhancement for each photo. The system learns from the user's enhancement, and then uses this learning for suggesting improved auto-enhancement for a new image. This can thus be considered to be self-reinforcing photo color enhancement. Our work makes several contributions on top of this. Our novel formulation for learning users' preferences (*e.g.*, the joint-space formulation) can be used to develop support functions beyond auto-enhancement, including enhanced sliders and confidence values of preference estimations. As well as developing the system, the other key contribution is the first user study of self-reinforcing systems.

Automatic Photo Color Enhancement

It is often considered (*e.g.*, [6]) that auto-enhancement algorithms used in most packages are based on simple heuristics such as histogram stretching, which cannot work for complex cases. To improve the quality of auto-enhancement, machine-learning techniques are often used [6, 21], of which the most relevant is the personalized auto-enhancement proposed by Kapoor *et al.* [21]. In this formulation, the user trains the system by manually enhancing a set of carefully-selected training photos, and then the system provides an auto-enhancement function that reflects the user's personal preference. Here, the training phase and the execution phases are completely separated, and it does not provide any support for manual editing. Although our work makes partial use of a similar underlying technique (metric learning), the interaction workflow is completely different as there is a seamless transition from training to task execution. Our main contribution is in the investigation of this paradigm and the functions to support the seamless transition. In addition, we newly introduce algorithms to enable our support functions, including an algorithm to compute confidence values that are used to adjust the interface behavior, and a joint-space formulation that is necessary for enhanced sliders.

HaCohen *et al.* [17] presented a method to automatically achieve consistency within a collection of photos. This method also allows users to manually correct a small number of photos and then automatically propagates the correction to the other photos in the collection. Berthouzoz and her colleagues [15, 5] presented a method to create content-adaptive photo manipulation macros for batching the process. Similar to ours, this method learns the relationship between photo features and user-specified parameters. Their goal is to enable automatic batch enhancement of a large set of photos, and they do not aim at supporting one-by-one manual editing. Jaroensri *et al.* [18] presented a method to automatically predict the acceptability of a given photo enhancement. Their model is computed using dataset obtained via crowdsourcing in advance, while ours is computed progressively using the personal editing history. Additionally, no previous work evaluates how self-reinforcement can be used to improve the user experience with manual enhancement of individual photos.

Preference Learning

Methods for learning aesthetic preferences have been presented in various design domains [33, 38, 36, 34, 35], such as web pages [36] and color themes [33]. Especially, assessment of photo preference has been investigated in great detail by many researchers [12, 22, 30, 31]. Their goal was to assess the quality of a photograph rather than to facilitate its color enhancement. In other words, while they focus on photo features, we are more interested in enhancement parameters.

Several methods of learning preferences using design parameter spaces, rather than feature spaces, have been proposed [41, 23]. For example, the method presented by Talton *et al.* [41] uses many participants to obtain data for analyzing the parameter space. These methods are similar to ours in that they use the learned preference models to facilitate design exploration. In contrast to theirs, ours considers the feature space of photographs in addition to the parameter space, which enables its use for self-reinforcing photo enhancement.

Demonstration-Based Techniques

Our method learns the user's preference from the user's demonstration. This can be considered as a derivation of

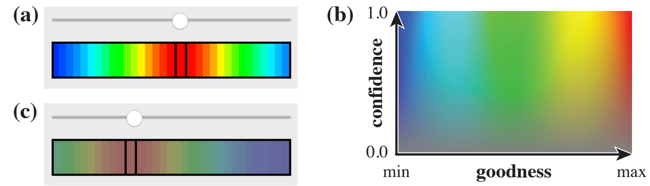


Figure 2. (a) Heat-map visualization on sliders (conf. $= 1.000$). (b) The color scheme extended by using the confidence value. (c) Visualization with low confidence (conf. $= 0.216$).

programming by demonstration [11, 26, 28]. The concept of programming by demonstration has been examined in previous research on the automation of photograph editing [15, 5]. It is also incorporated into commercial software such as the macro creation in Photoshop (called *Actions*) [2]. While they aim at automating tasks, we aim at supporting manual tasks. Also, they usually require explicit training or authoring phases, while our approach seamlessly integrates training and task execution phases. Another domain of demonstration-based techniques is *adaptive user interface* [14, 13], which adapts to the user based on the user's behavior and context to improve usability and performance. While our system also adapts to the user, we aim at facilitating open-ended creative tasks, rather than improving usability of the interface.

SELF-REINFORCING PHOTO ENHANCEMENT SYSTEM

This section describes our prototype system, called SelPh. SelPh was designed based on the concept of self-reinforcing color enhancement. The underlying algorithms will be described in the next section. Figure 1 (b) shows the user interface of SelPh. The left top part is a preview widget that shows the currently-enhanced photo. The left bottom part is a reference widget that shows already-enhanced photos for reference. The right part is a control widget that provides functions for color enhancement. SelPh's basic functionality comprises six sliders that adjust enhancement parameters: brightness, contrast, saturation, and color balance with respect to red, green, and blue. SelPh also provides five user support functions enabled by the self-reinforcement.

User interaction proceeds as follows. First, the user imports all the target photos into SelPh. Then, SelPh displays the first photo, and the user starts enhancement. Once the user is satisfied with the enhancement result, the user pushes the "next" button, and the next photo appears. This procedure is repeated until all the photos are enhanced.

User Support Functions

Our self-reinforcement enables the following five functions to be made available to the user:

Visualization of Goodness on Sliders. The system includes the VisOpt Slider interface presented by Koyama *et al.* [23], where the system provides a colorful "bar" along each slider. A heat map is displayed to show the distribution of "good" parameters for each bar (Figure 2 (a)), where the red (blue) color indicates that it is a good (bad) parameter choice. This colorful visualization is expected to help the user to explore the parameter space more efficiently.

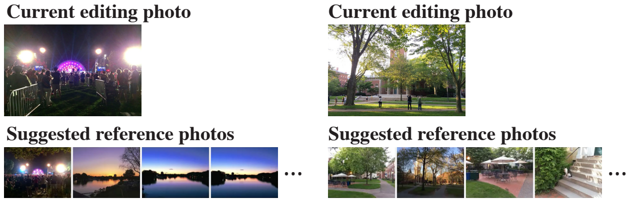


Figure 3. Two examples of suggested reference photos.

Interactive Optimization of Slider Values. In addition to the visualization, the VisOpt Slider provides an interactive optimization function [23]. When this function is enabled, as the user adjusts a slider value, all the other slider values are simultaneously optimized so that collectively the sliders values give better overall result. This function is useful for getting away from meaningless design spaces, thus reducing the user's effort during his or her exploration.

Variable Confidence Value. SelPh computes a value called *confidence value*. This indicates the confidence, or the certainty, of the estimation of the user's preference with respect to color enhancement. This confidence value is used to adjust the visualization and optimization functions. For visualization, the system modifies the color scheme as shown in Figure 2 (b). For example, when the confidence value is small, the color becomes monotonous (Figure 2 (c)). This prevents the system from displaying low quality estimation to the user. For optimization, the system adjusts the strength of the automatic guidance so that more optimization is performed when the confidence value is large, and less is performed when the confidence value is small.

Auto-Enhancement. SelPh provides an "auto-enhance" button. Pressing this button automatically sets all the sliders values to the estimated optimal parameters. This optimization is computed using the preference model learned from the user's editing history. As noted, auto-enhancement cannot always work well; however, as our auto-enhancement adapts to the user's preference, it is expected to provide a reasonable starting point for further exploration.

Reference Photos. When a collection of photographs is enhanced, consistency among the enhanced photographs must be ensured [17]. To facilitate consistent photo enhancement, the system shows the user *reference photos*, which are the already-enhanced photos adaptively sorted by the estimated similarity to the photo that the user is currently editing. Figure 3 shows examples of reference photos. Note that the similarity between photos is personalized; *i.e.*, it is learned from the user's editing history.

Figure 4 shows an example sequence of color enhancement performed by SelPh. For the first and second photos, no user support is provided because the system does not have enough data for reinforcement. From the third photo onwards, the guidance functions are provided. The confidence value increases from the third to the fifth photos (0.359, 0.760, and 0.785). As these photos are similar to each other, the system is able to estimate the user's preference effectively. However, for the sixth photo, which is quite different from any of the photographs that have already been enhanced, the confidence value decreases (0.109), because it is difficult for the system to estimate the enhancement preference based on the previously enhanced photos. For the seventh and eighth photos, which are similar to the sixth photo, the system successfully estimates the enhancement preference with higher confidence (0.432 and 0.865).

ALGORITHMS

Overview of Self-Reinforcement Procedure

Every time the user finishes an enhancement of a photo and then pushes the "next" button to go to the next photo, the system computes the self-reinforcement procedure (Figure 5). This procedure consists of the following three steps:

Step 1: Update the distance metric of photographs. The system learns the distances, or dissimilarities, between photographs based on the user's past enhancement history so that the user's personal preference is reflected.

Step 2: Update the photo feature space. The system learns personalized feature vectors as descriptors of photographs. These are computed based on the learned distance metric.

Step 3: Update the enhancement preference model. The preference model for estimating the quality of the enhancement is updated based on the learned photo feature space and the user's past enhancement history.

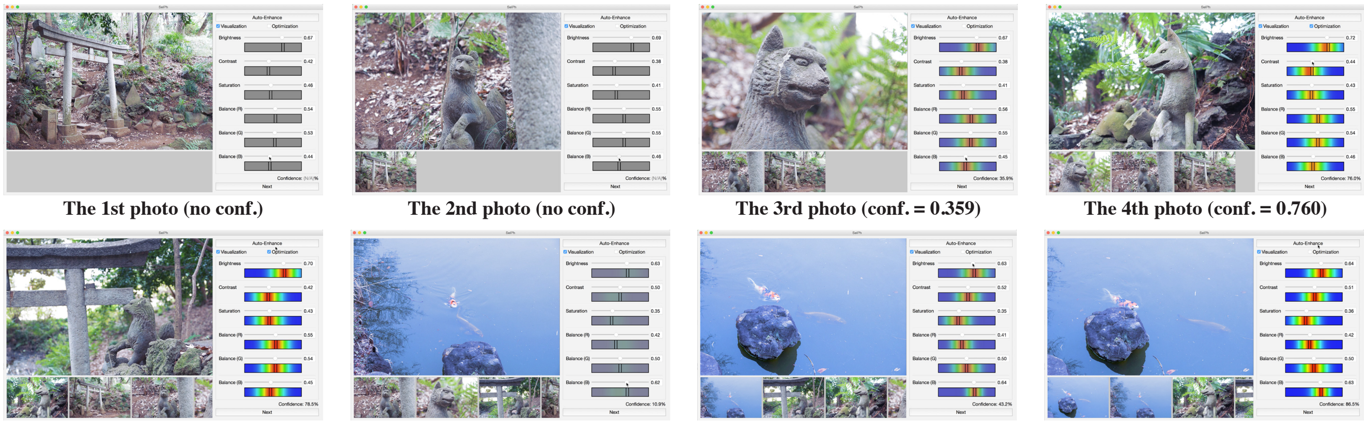
This procedure can be computed in less than 40 milliseconds with maximum 50 photos (MacBook Pro with 3 GHz Intel Core i7). After the computation of this procedure, the next photo is loaded and shown to the user, with the updated user support functions. In the following sub-sections, we describe these three steps, and then show the way these techniques are used to enable the user support functions.

Assume that the user has just pushed the "next" button. Let n be the number of already-enhanced photos, I_i is then the i-th photo image, and $\mathbf{p}_i \in [0, 1]^p$ is the enhancement parameter set that the user specified for I_i. The value p is the number of parameters (in our case $p = 6$), and the slider's minimum and maximum values are mapped to 0.0 and 1.0 respectively. Given the data $\mathcal{D} = \{(\mathbf{p}_1, I_1), \ldots, (\mathbf{p}_n, I_n)\}$, the goal is to estimate the user's preference of enhancement for the next photo I_{n+1}.

Distance Metric Learning of Photos

The goal in this step is to learn an appropriate distance metric for photos from the available data $\mathcal{D}$. This problem can be seen as a derivative of well-studied *metric learning* techniques [25]. In our case, the goal is to learn a distance function $d(I_i, I_j)$ between an arbitrary pair of photos.

For this purpose, we mostly follow the method presented by Kapoor *et al.* [21]; the distance metric is optimized so that the distances between photos are as proportional to the distances between associated parameters as possible. The distance function d is represented as a weighted sum of 38 types of non-linear distances between low-level photo features, such as the symmetric KL-divergence between intensity histograms. The reader is encouraged to consult the original paper for details. Let $\mathbf{w} \in \mathbb{R}^{38}$ be the weights that



The 1st photo (no conf.) The 2nd photo (no conf.) The 3rd photo (conf. = 0.359) The 4th photo (conf. = 0.760)

The 5th photo (conf. = 0.785) The 6th photo (conf. = 0.109) The 7th photo (conf. = 0.432) The 8th photo (conf. = 0.865)

Figure 4. An example sequence of color enhancement using SelPh.

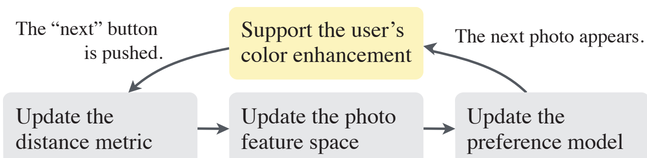


Figure 5. Overview of our self-reinforcing procedure. The yellow box is the design session, and the gray boxes are the self-reinforcing sessions.

parameterize the distance function. The weights $\mathbf{w}$ are computed by solving the minimization problem:

$$\min_{\mathbf{w}} \sum_{\substack{(i,j)\in \\ \{(a,b)|1\leq a<b\leq n\}}} \left(d(I_i, I_j; \mathbf{w}) - \alpha\|\mathbf{p}_i - \mathbf{p}_j\|\right)^2, \quad (1)$$

where $\alpha > 0$ is a parameter that determines the relative scaling between the distances of parameters and those of photos. Here, we slightly modified the original formulation by introducing α; the original formulation is the special case where $\alpha = 1.0$. We empirically found that a value of $\alpha \in [1.0, 5.0]$ works well, and we chose $\alpha = 3.0$ for all the examples in this paper. This minimization problem is solved using the limited memory BFGS [29], a local gradient-based optimization algorithm, provided in the nlopt library [19]. We observed that this minimization takes only negligible time, *e.g.*, typically 10–30 milliseconds for $n = 50$.

Photo Feature Space Computation

Having found an appropriate metric in the previous subsection, the distance between any pair of photographs can now be measured. Based on this distance metric, we define appropriate coordinates for any photos, *i.e.*, positions of photos in a Euclidean space. This operation of assigning coordinates in a particular space to elements is called *embedding*. To this end, we use *metric multidimensional scaling* (metric MDS) [10], which computes positions of target elements in a k-dimensional Euclidean space, given distances between them. In this work, we specify $k = 5$. We embed the already-enhanced photos $I_1, \ldots, I_n$ and the next photo I_{n+1} into the k-dimensional space. We refer to the resulting Euclidean space as *learned feature space*, and the position of I_i

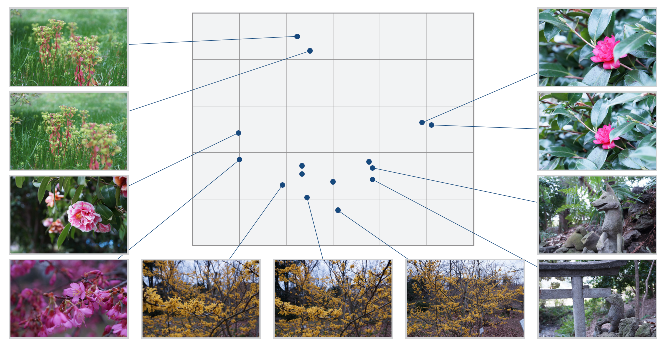


Figure 6. Visualization of a learned photo feature space computed by the metric learning and embedding. This space is updated based on the user-provided parameters each time the "next" button is pushed.

as the *learned feature vector* $\mathbf{f}_i \in \mathbb{R}^k$. Figure 6 shows an actual result of embedding 15 photos, where $k = 2$ is specified just for visualization purpose. It denotes that a closer pair of photos in this learned feature space is likely to be provided with more similar enhancement parameters, and *vice versa*.

Other algorithms such as Isomap [42] can also be used here. However, since this embedding is performed every time, it needs to be efficient. Thus, we chose the simple and fast metric MDS in our implementation rather than Isomap.

Enhancement Preference Model

Joint-Space Formulation

As the computational model of photo enhancement preferences, we formulate the *goodness function* of color enhancement as

$$g(\mathbf{p}, \mathbf{f}) \in \mathbb{R}, \quad (2)$$

which returns a scalar-valued "score" (*i.e.*, *goodness*) of the enhancement parameter $\mathbf{p}$ for the input photo whose learned feature vector is $\mathbf{f}$. For example, given a target photo whose learned feature vector is $\mathbf{f}$, the goodness function $g(\mathbf{p}, \mathbf{f})$ would return a large value if the enhancement parameter set $\mathbf{p}$ provides a good enhancement, and return a small value if it provides bad one.

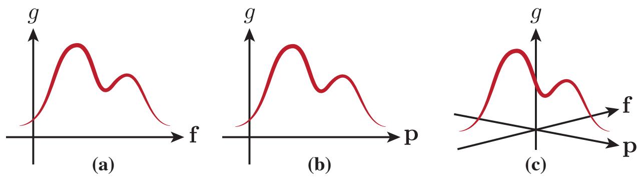


Figure 7. Comparison of goodness function definitions. (a) In typical preference learning approaches, the goodness is learned in the feature space. (b) Some methods for facilitating parameter tweaking learn the goodness in the parameter space. (c) In this work, the goodness is learned in the joint space of the features and the parameters.

To integrate these two different spaces, the parameter vector space $\mathbf{p}$ and the learned feature space $\mathbf{f}$, we consider a higher-dimensional joint space of these two spaces:

$$\mathbf{x} = (\mathbf{p}^T \ \ \mathbf{f}^T)^T \in \mathbb{R}^{p+k}, \tag{3}$$

and then formulate the goodness function in the joint space.

Figure 7 shows concept-level comparisons of the goodness function definitions between previous ones and ours. Many computational aesthetics methods (*e.g.*, [38, 22, 31]) consider the relationship between the feature space and the goodness value (Figure 7 (a)). Most of these papers discuss the selection of features or the learning algorithms required for appropriate assessment. Some methods to facilitate design parameter adjustment (*e.g.*, [41, 23]) analyze the parameter space and derive a distribution of "good" parameters (Figure 7 (b)). In contrast, we consider *both* the feature space and the parameter space *jointly* (Figure 7 (c)). This is the key to facilitating parameter adjustment for new photos that have not been analyzed directly. To the best of our knowledge, this specific formulation has not been previously investigated.

Goodness Function Representation

Given the jointed data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the specific goal in this step is to derive a scalar-valued function $g(\mathbf{x}) \in \mathbb{R}$ that is smooth and has higher values at the points in $\mathcal{X}$. To compute such a function, we use the *kernel density estimation* techniques, especially those from the work of Talton *et al.* [41]. The reader is referred to the original paper for details. Essentially, the goodness function is represented as a weighted sum of the Gaussian kernels $K_i(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{x}_i, \mathbf{\Sigma}_i)$, where $\mathbf{x}_i$ is the i-th data point, and $\mathbf{\Sigma}_i$ is the bandwidth matrix. $\mathbf{\Sigma}_i$ is computed *adaptively* by using the techniques described in [4, 37]. In contrast to Talton *et al.*'s work, we add a simple post-processing step on $\mathbf{\Sigma}_i$. In this step, the diagonal elements are forced to be more than $\epsilon > 0$. This prevents unnecessary discontinuity in the resulting function and enables smooth exploration. In SelPh, the value $\epsilon = 0.02$ was used.

Implementation of User Support Functions

Confidence Value. The following assumption was made: if there is an already-enhanced photo that is similar to the next photo, the goodness function can provide an accurate estimation of preference for the next photo. Based on this idea, we empirically define the confidence value as

$$(1 + d_{\text{closest}})^{-\beta}, \tag{4}$$

where d_{closest} is the distance between the next photo and its closest already-enhanced photo in the learned feature space, and β is a parameter to control the mapping curve. The fixed value $\beta = 3.0$ is used throughout this work. By this definition, the confidence value becomes nearly 1.0 if there is an already-enhanced photo similar to the next photo, and becomes nearly 0.0 in the opposite case.

Slider Interface. For the visualization and optimization functions, the VisOpt Slider interface [23] is used, where the input to the interface consists of a design parameter vector $\mathbf{p}$ and a scalar-valued function $f(\mathbf{p}) \in [0, 1]$. To fit our formulation to that required to use the VisOpt Slider interface, we define f as $f(\mathbf{p}) = (g(\mathbf{p}, \mathbf{f}) - g_{\min})/(g_{\max} - g_{\min})$, where $\mathbf{f}$ is the learned feature vector of the target photo, which is fixed during users editing. $g_{\min}, g_{\max}$ are the minimum and maximum values, respectively, when $\mathbf{f}$ is fixed. These values are computed by the limited memory BFGS [29]. In addition, we extend this VisOpt Slider to incorporate the confidence value. This is done by modifying the color scheme (Figure 2 (b)) and changing the strength of the interactive optimization.

Auto-Enhancement. When the auto-enhancement function is called, the system computes the optimal parameter set for maximizing $g(\mathbf{p}, \mathbf{f})$ where $\mathbf{f}$ is considered to be fixed. The result is then applied to all of the sliders. The maximization problem is solved by the limited memory BFGS [29], typically taking < 15 milliseconds.

USER STUDY

We conducted a user study to investigate how photographers enhance photos repetitively with a self-reinforcing system, whether and how they are satisfied with the self-reinforcement approach and the user support functions of SelPh. The task was color enhancement of a collection of photos (*e.g.*, a photo album). In this study, the efficiency of the enhancement was not evaluated. This is because photo color enhancement is essentially a creative, open-ended exploration task. Hence, it is difficult to quantify by the task-completion time. We consider that the time taken to complete the task is not critical to the overall user experience. The quality of the enhancement result was not evaluated either. In our target use case, the resulting photos are considered to be always satisfactory for the photographer.

Participants

We recruited eight skilled photographers (male: 4, female: 4) via several community mailing lists and Facebook. To ensure they had the desired skills, participants were required to satisfy all of the following conditions: (1) be majoring or had majored in Art or Design, (2) own and be familiar with his or her own camera, not including casual cameras such as smartphones, and (3) be familiar with software for photo color enhancement such as Photoshop. As a result, three of the participants were students majoring in Art or Design ($P_1, \ldots, P_3$) and the other five were professional designers ($P_4, \ldots, P_8$).

Procedure

Preparation: Photo Taking

First of all, we asked participants to take a series of photographs for the user study. We asked participants to take

100 photographs in a single day by the same camera, with either manual or automatic exposure settings. We allowed participants to take photos in arbitrary times and places. We instructed participants to assume that their assignment was to create a commercial photo book. The theme of the photo book was left to the discretion of the individual photographers but example themes such as "travel," "animal," "wedding party of your friends," and "walking around" were provided. Finally, we told the participants not to enhance any of the photographs prior to the commencement of the enhancement task.

Main Task: Photo Enhancement

We arranged a day to conduct the main task for each participant. The study was performed in our laboratory or classrooms. First we asked them to fill a preliminary questionnaire. This contained questions on biographic information and their opinions of the auto-enhancement in commercial software. Then, we introduced the systems for the study. We prepared two systems: SelPh and a baseline system (*Baseline*), which simulates a simple, typical software interface. Baseline was prepared by limiting the user support functions in SelPh. First, we omitted both the visualization and optimization functions from sliders. Second, as for the auto-enhancement, we removed the adaptation to individual photos from that of Baseline; it simply takes the average of all the user-specified parameters from the previous edits to perform auto-enhancement. Finally, instead of showing reference photos in the order of similarity, Baseline shows reference photos in the original order; that is, the most recently-edited photos are shown first as references.

Each set of photographs was divided into two sets of 50. The order of the photos was retained so that the participant edited the photographs in the order in which they were taken. Participants were asked to enhance each set of photographs for each system. We instructed the participant how to use the system and its functions before he or she started the editing task. We then asked them to practice the system by editing 20 photos that we prepared. We alternated the order that the participants used SelPh and Baseline systems to counterbalance order effects. The participants were told to judge the speed to work at, and quality to aim for, themselves, bearing in mind that they were aiming to create a commercial photo book. The visualization and optimization functions, as well as the auto-enhancement function, were optional; the participants were shown how to toggle this functionality using check boxes and were told that they were free to do so as they wished.

After completing the tasks of enhancing 100 different photos in total, 50 using Baseline and 50 using SelPh, the participants filled in another questionnaire. This post-task questionnaire contains questions about the approach of self-reinforcement and the support functions available in SelPh. Then, we conducted informal interview. The questions in the interviews were based on individual's answers to the questionnaires, and the enhancements they made. We conducted this study in almost the same lighting conditions using the same display. The overall process took around 3 hours for each participant.

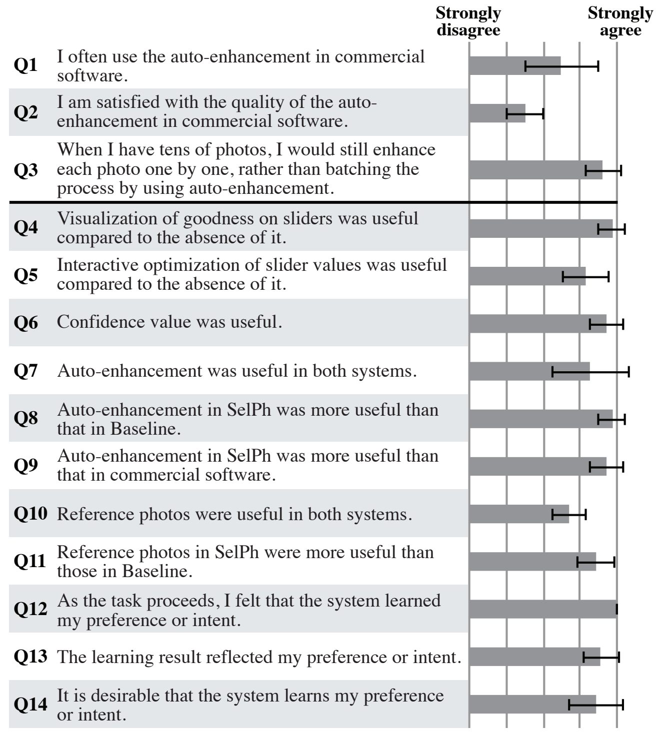


Figure 8. Results of the preliminary (Q1–Q3) and post-task (Q4–Q14) questionnaires. We used a 5-pt Likert scale. The error bars represent the standard deviation.

RESULTS

Preliminary and Post-Task Questionnaires

Our preliminary and post questionnaires were arranged on a 5-pt Likert scale, where 5 corresponds to "strongly agree." Figure 8 (Q1–Q3) shows the results of the preliminary questionnaire. These results validate the decision to work on improving repetitive manual enhancement rather than a fully-automated batch process. Figure 8 (Q4–Q14) shows the results of the post-task questionnaire with respect to the self-reinforcement approach and the user support functions. Overall, participants gave positive scores.

Feedback in Interview

Self-Reinforcement Approach

Compared to Baseline and traditional software where the enhancement of each photo is independent, P_8 said, *"(SelPh) gave me a sense of continuity"* between each enhancement. He said he often began by *"pushing the auto-enhancement button"* and then *"explored (the design space) from that point,"* from which he felt the sense of continuity in this repetitive task.

P_2 said, *"The functions based on the learning result, such as the visualization on sliders, evoke the feeling of collaborating with another me."* Similarly, P_8 said that, using Baseline, he felt *"lonely because I needed to do everything."* In contrast, he said that, in SelPh, *"there is interaction with the system"* through the support functions, thus *"executing the task (with SelPh) was fun."*

Overall P_3 was satisfied with our system, but she had one concern, namely that *"(my decision) was sometimes too affected by the visualization,"* while she was partially positive about this effect because it is helpful in ensuring consistency across edits. On the other hand, P_4 also mentioned a similar point but from a more positive stance, saying that *"This (visualization) is helpful (in making decisions) when I have little confidence in myself."*

P_5 mentioned that he often uses customizable filters in Aperture for preprocessing photos, but *"when the number of filters increases, it is tiresome to manage them"* because it is difficult to remember all the filters and find the most appropriate one. Compared to this, he said, *"the method of (automatically) showing the recommendation of parameters is effective for smoothly proceeding with a photo enhancement task."*

Regarding the preference learning, P_5 commented that *"I felt that my preferences and tendencies are gradually learned, which was good."* P_7 said, *"(SelPh) reflected my tendency of tweaking two parameters in pairs,"* and thus she realized that *"the system actually learned my preference."*

Visualization of Goodness on Sliders

As already mentioned above, P_4 said that the visualization helps with decision-making, especially in cases where he has little confidence in himself. This sentiment was echoed by P_3, P_5, P_7, and P_8. Also, P_7 mentioned that *"(the visualization was) useful for avoiding unnecessary exploration"*; P_5 agreed with this idea. While P_8 agreed that the visualization is helpful, he also said *"The information that I have to see during enhancement increased."*

Interactive Optimization of Slider Values

P_1, P_3, P_4, and P_6 agreed that the option to interactively optimize slider values was useful because it allows efficient exploration. P_6 liked it also because *"it is not fully-automatic"* but semi-automatic, where she *"can still tweak each slider"* as she desires. P_7 also liked it because *"Some common procedures that I have in my head were (automatically) done (by this function) so I did not have to do them by myself."* However, P_7 also said, *"When I was doing fine-tuning, I had to turn it off,"* because it conflicted with her edits. She suggested that the problem of having to turn the function on and off *"can be easily solved by introducing a shortcut key."*

Variable Confidence Values

All the participants agreed that the confidence value was useful. For example, P_7 said, *"By knowing the confidence, users can efficiently make decisions."* P_3 said that, according to the confidence value, she *"decided to use or not to use the optimization and the auto-enhancement,"* and that *"when the confidence value is relatively low, I did (explore the design space) by myself"* rather than using the support functions.

Some participants mentioned that the confidence value influenced other aspects of the user experience. P_5 commented, *"I could trust the system more"* by knowing the confidence of the estimation. P_7 felt *"a sense of closeness because it seems human especially when the confidence value decreased."* P_1 said, *"it was good to know the confidence because I could guess the thoughts of the system,"* and she *"could empathize with the system."* P_8 also felt *"humanity"* from the confidence value, and he said, *"It was an enjoyable experience to do the task while feeling that 'This guy (SelPh) failed to guess what I want to do!' "* when the confidence value decreased.

Figure 9. Examples of photos that took a relatively long time to be enhanced during the study. From left to right, the photos were taken by P_1, P_3, and P_4, and the times were 56, 37, and 33 seconds, respectively.

Auto-Enhancement

P_5 commented that the auto-enhancement in Baseline could make the design *"approach a little better, but it is still far (from the best design)."* He agreed that the auto-enhancement in SelPh was better than that in Baseline, and said that he *"used it (as a starting point) for further exploration."* P_4 commented that, compared to the auto-enhancement in SelPh, that in Photoshop *"cannot reflect my intent"* and *"only provides a safe enhancement."* P_3 said that, typically when using Photoshop, *"I rarely use the auto-enhancement,"* because *"usually it does not fit my preference."* On the other hand, she used the auto-enhancement in SelPh for about half of photographs during the study. She agreed that she *"would use the auto-enhancement if it learns from my past edits."*

Reference Photos

P_1 and other participants agreed that the reference photos helped them to edit the photographs in a consistent manner. She also commented that, when making an adjustment, references were helpful in avoiding being *"overly influenced by machine (the system recommendation)"*. On the other hand, some participants (P_4, P_6, P_7, and P_8) did not agree that they made use of the reference photos during the study.

Time-Consuming Photos

Figure 9 shows some of photos that took relatively longer to edit during the task execution. We asked participants why these photos took longer to edit than the others. P_1 said that she wondered *"whether the sky should be completely 'white-out', or should retain its texture,"* so *"I had to explore a lot"* to enhance this photograph appropriately. P_3 said that he *"intended to take a special direction"* regarding the sunlight filtering through the trees. P_4 mentioned that he wanted to *"balance the blue in the sky with the red in the ground."* To do so, he adjusted the green slider as well, to ensure balance between the blue and red sliders. These statements show that the exploration time is significantly dependent on the contents of photographs. Thus, it is not plausible to evaluate the effectiveness of SelPh by simply using the task-completion time.

Other Comments

P_3 and P_6 wanted the option to control which dataset would be used for learning and supports based on particular use cases. For example, P_6 said that she changes the atmosphere of photos according to the clients, and thus *"it is very helpful if I can switch (the datasets) (based on clients)."* P_4

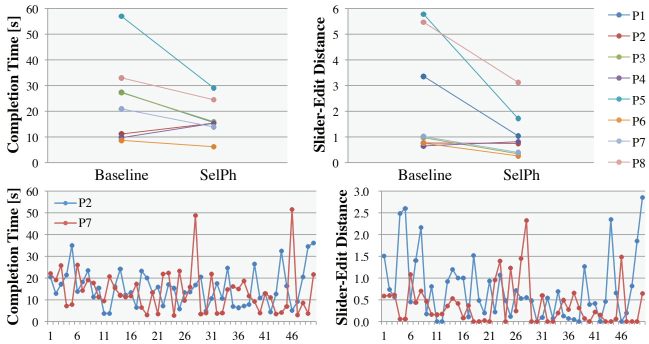


Figure 10. (Top) The mean completion time and slider-edit distance per photo. (Bottom) Series of the completion time and slider-edit distance.

commented that, when finishing the enhancement of 50 photos with SelPh, he *"wanted to re-visit some of the already-enhanced photos, with fully-trained user support functions."* This idea was also mentioned by P_3, P_5, and P_6.

Quantitative Results

Although quantitative evaluation is not our main focus, we recorded some quantitative variables. Figure 10 (Top) show the mean task-completion time and the mean *slider-edit distance*. We refer 'the slider-edit distance' to the total amount of slider movements that were made by mouse drags, where the distance 1.0 corresponds to the width of a slider. The overall average of the mean task-completion time was 24.4 [sec] in Baseline and 17.0 [sec] in SelPh. The overall average of the mean slider-edit distance was 2.35 in Baseline and 1.05 in SelPh. Figure 10 (Bottom) shows the task-completion time and slider-edit distance sequences through 50 photos with SelPh, by P_2 and P_7. As for preference learning performance, we recorded the prediction errors, *i.e.*, the L^2-norms between the predicted optimal parameter sets and the user-provided ones, and the confidence values. The mean prediction error in each editing sequence was $\{\mathrm{min} = 0.05\ (P_3), \mathrm{max} = 0.26\ (P_8), \mathrm{mean} = 0.11\}$. The mean confidence value was $\{\mathrm{min} = 0.18\ (P_8), \mathrm{max} = 0.65\ (P_3), \mathrm{mean} = 0.46\}$.

When participants were using SelPh, they chose to use the auto-enhancement feature for 83.0% of the photographs. We did not record how often the interactive slider optimization was actually used during exploration, but the check box was in the "checked" state for 62.8% photographs when the "next" button was pushed. Similarly, the check box for the visualization function was in the "checked" state for 100% photographs when the "next" button was pushed.

DISCUSSION

Summary of User Study

- **All the participants were satisfied with self-reinforcing color enhancement.** This was validated by responses to the post questionnaire items (Q12–Q14) and by the feedback comments given in the interview. We obtained positive comments compared to both fully-automatic and fully-manual approaches.
- **All the participants were satisfied with the support functions.** This was validated by questions (Q4–Q10) of the post-task questionnaire. While some participants were indifferent to the reference photo function, other functions were overall rated highly positively.
- **SelPh is useful for avoiding unnecessary exploration.** Participants explicitly mentioned this aspect. Also, compared to the baseline condition, the mean slider-edit distances decreased as a general trend when SelPh was used.
- **SelPh is useful for making decisions efficiently.** According to the participants' comments, this is true especially when users are not confident in themselves.
- **SelPh affects decisions.** A few of the participants were aware of this effect. It is possible that the decisions of other participants were also affected, but unconsciously.
- **The inclusion of the confidence value renders SelPh more helpful.** Many participants were excited about the confidence value, which they found very useful. For example, depending on the confidence, a user can decide whether he or she first applies the auto-enhancement, or adjusts sliders from the beginning.
- **The inclusion of the confidence value makes SelPh more trustworthy and enjoyable to use.** This was the effect that we found most interesting. The confidence value gave the system a sense of humanity, and also gave participants a sense of interaction and collaboration with the system. This facilitated trust between the users and the system, and helped users to take more enjoyment.

Implications

Several lessons for further developing a self-reinforcing color enhancement system can be learned from the study. We believe that these lessons are also applicable to the design of systems that use machine learning in general.

- It is preferable to show users what the system is thinking of, rather than to make the system a "black box." In the study, the confidence value played this role; it helped users with the editing task, and also made the system more trustworthy and enjoyable to use. Showing "why the system thinks so" is a possible future direction in this aspect.
- The workflow should be semi-automatic rather than fully-manual nor fully-automatic. For example, participants liked the interactive optimization of slider values because, in addition to the automatic guidance, this function gives the user the freedom to adjust each slider individually.
- Rather than guiding the user to a possible best design, enhancement systems should support the user's own exploration of the design space. From our observations and the feedback by P_4 and P_8, we found that photographers often manipulate sliders back and forth. This helps them to be sure that they have found a good design.
- In practice, photographers enhance photographs in different ways according to their intended usage. Thus, unlike the setting in our study, the system has to support a function to switch between several modes for self-reinforcement.

Limitations

User Study Method. We conducted the user study to collect initial feedback from participants, thus the method is not comprehensive and has several limitations. For example, we assumed that the same person takes photos and enhances them, but in practical usage, it is possible that different persons work on them separately. In addition, we required participants to enhance photos only in a linear order and disallowed revisiting previous photos for balancing the conditions across participants. We prepared our baseline by simply limiting the user support functions of SelPh so that participants could easily notice the qualitative properties of SelPh. However, other baseline conditions are also possible and could be useful for further investigation. Finally, the participants might have had a bias towards SelPh because of novelty factor; this needs to be taken care of for more rigorous evaluation.

Quantitative Evaluation. As the focus of our study was on the qualitative properties, quantitative aspects were not fully investigated. It is difficult to evaluate the effectiveness of the self-reinforcement in terms of either the task-completion time or the amount of mouse interactions; for example, Figure 10 (Bottom-left) reveals highly random variation in task-completion times during the study. During the interviews, it was made clear that the difficulty of photo enhancement and hence the time it takes to complete this task, is highly dependent on the target photo's properties, including context, elements, and subjects. While we used an open-ended task, more concrete goals for enhancement may need to be specified for an accurate quantitative evaluation. It is also important to design the study in such a way as to reduce the learning effects of participants and to ensure that the participants remain sufficiently motivated throughout the study.

Simple Global Photo Color Enhancement. The current system only has six parameters for color enhancement. This choice was based on previous work [23], as a first step for investigation of self-reinforcing systems. Future system should include more types of supported parameters (*e.g.*, Highlights/Shadows, parametric tonal curves [16, 17], and filters in Instagram) to see how this affects performance. The current system is limited to global enhancement; therefore, when a parameter is varied, it is varied for the entire image. Local enhancement is important for achieving more detailed enhancement [40, 9].

Empirically-Set Parameters. In our algorithms, there are several empirically-set parameters such as k, α, β, and ϵ. Although these parameters are not essential, they might affect user experience, which was not fully investigated in the study.

Accuracy of Machine Learning. There is a lot of scope for improving the accuracy of the system's predictions of user preferences. Our current implementation of distance metric learning uses only simple low-level photo features such as color histograms, following a previous work [21]. Investigating the use of semantic features such as object recognition [6, 27], local correspondences between photos [16, 17], or generic features [31, 24] represents a promising future direction.

Future Work

Extension to Multiple Users. Our current implementation of preference learning is limited to a single user; thus a user enhances photos based on only his or her own editing history. As some of the participants mentioned, it would also be interesting to develop algorithms and interaction techniques to share the learning results among many users and utilize them in a collaborative manner [7, 21]. Note that, relying on learned data from others may cause loss of a sense of "ownership and achievement" [3] in the resulting enhancement; we consider that the high satisfaction in our study was partially achieved by the fact that the learned data was from the user herself. We need to carefully design such a collaborative system considering this idea.

Other Support Functions. By using our self-reinforcement techniques, it is possible to provide more user support functions that were not available in SelPh. One interesting direction is to sort photos; it could be helpful to manipulate the order of photos, rather than providing them in the original order. For example, by always selecting the current-*worst*-confident photo as the next target photo, the system might effectively learn a user's preference (*i.e.*, active learning [39]). Another direction would be to develop a suggestive interface [32, 23] based on the learned preference model. In some scenarios, photographers may want to simultaneously edit multiple photos [16], where the learned distance metric between photos could be used to propagate edits to similar photos.

Towards Casual Users. Although adjusting sliders is a popular interface for skilled users, casual users might prefer alternative interfaces. Investigating a combination of the self-reinforcement approach and a palette-based [8] or gallery-based interface [32, 40] would be an important future work for casual usage. Also, it is important to evaluate how casual users enhance photos with a self-reinforcing system.

CONCLUSION

We investigated the concept of self-reinforcing color enhancement, where, as the user enhances photos, the system implicitly and progressively learns the user's intentions and preferences of color enhancement. Based on this concept, we designed and developed a prototype system, SelPh. It provides several support functions to the user such as enhanced sliders, tailored auto-enhancement, and confidence-based adaptation. Our qualitative user study revealed the effectiveness of this approach; for example, it could be preferable to the traditional manual adjustment, the self-reinforcing support functions are useful overall, and photographers are satisfied with them. The study also showed that it is very important to visualize what the system is thinking of to make a learning-based system trustful and enjoyable. Finally, a long-term evaluation of SelPh is necessary to better understand its potential for future practical usage.

Acknowledgements

Yuki Koyama is funded by JSPS research fellowship. This work was supported by JSPS KAKENHI Grant Number 26·8574 and 26240027.

REFERENCES

1. Adobe Systems Inc. 2015a. Photoshop Elements Help | Auto Smart Tone. (2015). Retrieved September 23, 2015 from https://helpx.adobe.com/photoshop-elements/using/auto-smart-tone.html.

2. Adobe Systems Inc. 2015b. Photoshop Help | Creating actions. (2015). Retrieved September 23, 2015 from https://helpx.adobe.com/photoshop/using/creating-actions.html.

3. Luca Benedetti, Holger Winnemöller, Massimiliano Corsini, and Roberto Scopigno. 2014. Painting with Bob: Assisted Creativity for Novices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 419–428. DOI: http://dx.doi.org/10.1145/2642918.2647415

4. Yoshua Bengio and Pascal Vincent. 2004. *Locally Weighted Full Covariance Gaussian Density Estimation*. Cirano working papers. CIRANO.

5. Floraine Berthouzoz, Wilmot Li, Mira Dontcheva, and Maneesh Agrawala. 2011. A Framework for Content-adaptive Photo Manipulation Macros: Application to Face, Landscape, and Global Manipulations. *ACM Trans. Graph.* 30, 5, Article 120 (Oct. 2011), 14 pages. DOI: http://dx.doi.org/10.1145/2019627.2019639

6. V. Bychkovsky, S. Paris, E. Chan, and F. Durand. 2011. Learning Photographic Global Tonal Adjustment with a Database of Input/Output Image Pairs. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*. IEEE Computer Society, Washington, DC, USA, 97–104. DOI: http://dx.doi.org/10.1109/CVPR.2011.5995413

7. J.C. Caicedo, A. Kapoor, and Sing Bing Kang. 2011. Collaborative personalization of image enhancement. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. 249–256. DOI: http://dx.doi.org/10.1109/CVPR.2011.5995439

8. Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. 2015. Palette-based Photo Recoloring. *ACM Trans. Graph.* 34, 4, Article 139 (July 2015), 11 pages. DOI: http://dx.doi.org/10.1145/2766978

9. Fanny Chevalier, Pierre Dragicevic, and Christophe Hurter. 2012. Histomages: Fully Synchronized Views for Image Editing. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 281–286. DOI: http://dx.doi.org/10.1145/2380116.2380152

10. Michael A. A. Cox and Trevor F. Cox. 2008. Multidimensional Scaling. In *Handbook of Data Visualization*. Springer Berlin Heidelberg, 315–347. DOI: http://dx.doi.org/10.1007/978-3-540-33037-0_14

11. Allen Cypher. 1991. EAGER: Programming Repetitive Tasks by Example. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*. ACM, New York, NY, USA, 33–39. DOI: http://dx.doi.org/10.1145/108844.108850

12. Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. 2006. Studying Aesthetics in Photographic Images Using a Computational Approach. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part III (ECCV'06)*. Springer-Verlag, Berlin, Heidelberg, 288–301. DOI: http://dx.doi.org/10.1007/11744078_23

13. Leah Findlater and Jacob Wobbrock. 2012. Personalized Input: Improving Ten-finger Touchscreen Typing Through Automatic Adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 815–824. DOI: http://dx.doi.org/10.1145/2207676.2208520

14. Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. 2006. Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '06)*. ACM, New York, NY, USA, 201–208. DOI: http://dx.doi.org/10.1145/1133265.1133306

15. Floraine Grabler, Maneesh Agrawala, Wilmot Li, Mira Dontcheva, and Takeo Igarashi. 2009. Generating Photo Manipulation Tutorials by Demonstration. *ACM Trans. Graph.* 28, 3, Article 66 (July 2009), 9 pages. DOI: http://dx.doi.org/10.1145/1531326.1531372

16. Yoav HaCohen, Eli Shechtman, Dan B. Goldman, and Dani Lischinski. 2011. Non-rigid Dense Correspondence with Applications for Image Enhancement. *ACM Trans. Graph.* 30, 4, Article 70 (July 2011), 10 pages. DOI: http://dx.doi.org/10.1145/2010324.1964965

17. Yoav HaCohen, Eli Shechtman, Dan B. Goldman, and Dani Lischinski. 2013. Optimizing Color Consistency in Photo Collections. *ACM Trans. Graph.* 32, 4, Article 38 (July 2013), 10 pages. DOI: http://dx.doi.org/10.1145/2461912.2461997

18. R. Jaroensri, S. Paris, A. Hertzmann, V. Bychkovsky, and F. Durand. 2015. Predicting Range of Acceptable Photographic Tonal Adjustments. In *Proceedings of the 2015 IEEE International Conference on Computational Photography (ICCP '15)*. IEEE Computer Society, Washington, DC, USA, 1–9. DOI: http://dx.doi.org/10.1109/ICCPHOT.2015.7168372

19. Steven G. Johnson. 2015. The NLopt nonlinear-optimization package. (2015). http://ab-initio.mit.edu/nlopt

20. Sing Bing Kang, A. Kapoor, and D. Lischinski. 2010. Personalization of image enhancement. In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*. IEEE Computer Society, Washington, DC, USA, 1799–1806. DOI: http://dx.doi.org/10.1109/CVPR.2010.5539850

21. Ashish Kapoor, Juan C. Caicedo, Dani Lischinski, and Sing Bing Kang. 2014. Collaborative Personalization of Image Enhancement. *Int. J. Comput. Vision* 108, 1-2 (May 2014), 148–164. DOI: http://dx.doi.org/10.1007/s11263-013-0675-3

22. Yan Ke, Xiaoou Tang, and Feng Jing. 2006. The Design of High-Level Features for Photo Quality Assessment. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR '06)*. IEEE Computer Society, Washington, DC, USA, 419–426. DOI: http://dx.doi.org/10.1109/CVPR.2006.303

23. Yuki Koyama, Daisuke Sakamoto, and Takeo Igarashi. 2014. Crowd-powered Parameter Analysis for Visual Design Exploration. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 65–74. DOI: http://dx.doi.org/10.1145/2642918.2647386

24. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 1097–1105.

25. Brian Kulis. 2013. Metric Learning: A Survey. *Foundations and Trends® in Machine Learning* 5, 4 (2013), 287–364. DOI: http://dx.doi.org/10.1561/2200000019

26. Tessa Lau, Lawrence Bergman, Vittorio Castelli, and Daniel Oblinger. 2004. Sheepdog: Learning Procedures for Technical Support. In *Proceedings of the 9th International Conference on Intelligent User Interfaces (IUI '04)*. ACM, New York, NY, USA, 109–116. DOI: http://dx.doi.org/10.1145/964442.964464

27. Congcong Li, Alexander C. Loui, and Tsuhan Chen. 2010. Towards Aesthetics: A Photo Quality Assessment and Photo Selection System. In *Proceedings of the International Conference on Multimedia (MM '10)*. ACM, New York, NY, USA, 827–830. DOI: http://dx.doi.org/10.1145/1873951.1874089

28. Greg Little, Tessa A. Lau, Allen Cypher, James Lin, Eben M. Haber, and Eser Kandogan. 2007. Koala: Capture, Share, Automate, Personalize Business Processes on the Web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 943–946. DOI: http://dx.doi.org/10.1145/1240624.1240767

29. D. C. Liu and J. Nocedal. 1989. On the Limited Memory BFGS Method for Large Scale Optimization. *Math. Program.* 45, 3 (Dec. 1989), 503–528. DOI: http://dx.doi.org/10.1007/BF01589116

30. Yiwen Luo and Xiaoou Tang. 2008. Photo and Video Quality Evaluation: Focusing on the Subject. In *Proceedings of the 10th European Conference on Computer Vision: Part III (ECCV '08)*. Springer-Verlag, Berlin, Heidelberg, 386–399. DOI: http://dx.doi.org/10.1007/978-3-540-88690-7_29

31. Luca Marchesotti, Florent Perronnin, Diane Larlus, and Gabriela Csurka. 2011. Assessing the Aesthetic Quality of Photographs Using Generic Image Descriptors. In *Proceedings of the 2011 International Conference on Computer Vision (ICCV '11)*. IEEE Computer Society, Washington, DC, USA, 1784–1791. DOI: http://dx.doi.org/10.1109/ICCV.2011.6126444

32. J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. 1997. Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 389–400. DOI: http://dx.doi.org/10.1145/258734.258887

33. Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2011. Color Compatibility from Large Datasets. *ACM Trans. Graph.* 30, 4, Article 63 (July 2011), 12 pages. DOI: http://dx.doi.org/10.1145/2010324.1964958

34. Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2014. Learning Layouts for Single-Page Graphic Designs. *IEEE Transactions on Visualization and Computer Graphics* 20, 8 (Aug. 2014), 1200–1213. DOI: http://dx.doi.org/10.1109/TVCG.2014.48

35. Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2015. DesignScape: Design with Interactive Layout Suggestions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1221–1224. DOI: http://dx.doi.org/10.1145/2702123.2702149

36. Katharina Reinecke, Tom Yeh, Luke Miratrix, Rahmatri Mardiko, Yuechen Zhao, Jenny Liu, and Krzysztof Z. Gajos. 2013. Predicting Users' First Impressions of Website Aesthetics with a Quantification of Perceived Visual Complexity and Colorfulness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2049–2058. DOI: http://dx.doi.org/10.1145/2470654.2481281

37. Juliane Schäfer and Korbinian Strimmer. 2005. A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics. *Statistical Applications in Genetics and Molecular Biology* 4, 1 (November 2005). DOI: http://dx.doi.org/10.2202/1544-6115.1175

38. Adrian Secord, Jingwan Lu, Adam Finkelstein, Manish Singh, and Andrew Nealen. 2011. Perceptual Models of Viewpoint Preference. *ACM Trans. Graph.* 30, 5, Article 109 (2011), 12 pages. DOI: http://dx.doi.org/10.1145/2019627.2019628

39. Burr Settles. 2009. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison, Madison, WI, USA.

40. L. Shapira, A. Shamir, and D. Cohen-Or. 2009. Image Appearance Exploration by Model-Based Navigation. *Comput. Graph. Forum* 28, 2 (2009), 629–638. DOI: http://dx.doi.org/10.1111/j.1467-8659.2009.01403.x

41. Jerry O. Talton, Daniel Gibson, Lingfeng Yang, Pat Hanrahan, and Vladlen Koltun. 2009. Exploratory Modeling with Collaborative Design Spaces. *ACM Trans. Graph.* 28, 5, Article 167 (2009), 10 pages. DOI: http://dx.doi.org/10.1145/1618452.1618513

42. Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 290, 5500 (2000), 2319–2323. DOI: http://dx.doi.org/10.1126/science.290.5500.2319

43. Michael Terry and Elizabeth D. Mynatt. 2002. Side Views: Persistent, On-demand Previews for Open-ended Tasks. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST '02)*. ACM, New York, NY, USA, 71–80. DOI: http://dx.doi.org/10.1145/571985.571996