Personalization of Image Enhancement

Sing Bing Kang[1] Ashish Kapoor[1] Dani Lischinski[2]
[1]Microsoft Research, Redmond, WA, USA [2]The Hebrew University of Jerusalem, Israel

Abstract

We address the problem of incorporating user preference in automatic image enhancement. Unlike generic tools for automatically enhancing images, we seek to develop methods that can first observe user preferences on a training set, and then learn a model of these preferences to personalize enhancement of unseen images. The challenge of designing such system lies at intersection of computer vision, learning, and usability; we use techniques such as active sensor selection and distance metric learning in order to solve the problem. The experimental evaluation based on user studies indicates that different users do have different preferences in image enhancement, which suggests that personalization can further help improve the subjective quality of generic image enhancements.

1. Introduction

Digital cameras are becoming increasingly ubiquitous as a result of decreasing cost and expanded functionalities, but taking good quality photos remains a challenge for the typical consumer. While almost every photograph could benefit from some tone and color adjustment, manually touching up every single image is impractical. There are generic automatic enhancement tools (such as Picasa or Windows Live Photo Gallery) that attempt to address this problem, but they operate without considering any user preference. While it is reasonable to assume that different people have different tastes in photo enhancement, there are hardly any attempts to explore different user preferences and show how these can be used to personalize image enhancements.

In this paper, we seek to explore and understand the role of such personalization and build an image enhancement system that learns about user preferences. The key points that we wish to address are: (1) Are user preferences in image enhancement different enough to warrant personalization? (2) How can we capture user preferences? To this end, we describe a system capable of user-dependent image enhancement. The system learns a user's preference model by asking her to enhance a small set of training photographs with help of a novel visualization interface. The learned preference model can then be used to automatically enhance any *test* (unseen) image. We conduct user studies whose results indicate that indeed the different users have different preferences in image enhancement and that personalization can improve the quality of automatic image enhancement.

The end-to-end system we describe addresses a unique mix of challenges from computer vision, learning, and UI design. Our work has several technical contributions: We develop a technique that helps provide a good overview of possible enhancement candidates to the user. In particular, one of the key contribution of our work is the interpretation and application of an active sensor selection to determine representative images for training. We also use a distance metric learning technique in order to determine a similarity measure that would map all images with similar enhancement requirements in the similar regions in image space. Finally, we present an end-to-end pipeline that covers training, user interface issues and testing; this system shows promising results on personalized image enhancement.

2. Related work

Most techniques for automatic image correction or enhancement typically focus on very specific features. For example, there is substantial work on denoising (e.g., [18]), geometric distortion removal (e.g., [6]), and optical correction (e.g., [12]) from a single image. There are also techniques for automatically linearizing the color space of a single image, e.g., through inverse gamma correction [5] and linearizing the color edge profile in RGB space [14]. Such approaches generally produce results that are objectively better than the inputs and thus user-independent.

There is also a fair amount of work done on automatic or semi-automatic color correction, e.g., [9, 10, 11]. Because of the ill-posed nature of the problem, such techniques may fail when assumptions made (such as average pixels being gray) are not applicable. There are also learning-based techniques for automatically enhancing images (e.g., dictionary learning [4, 16] and example-based [8]), but the database used for learning tend be generic.

The work closest to ours is that of Dale et al. [3]. They developed a system for restoring images using a large database of images. Given an image to be restored, the sys-

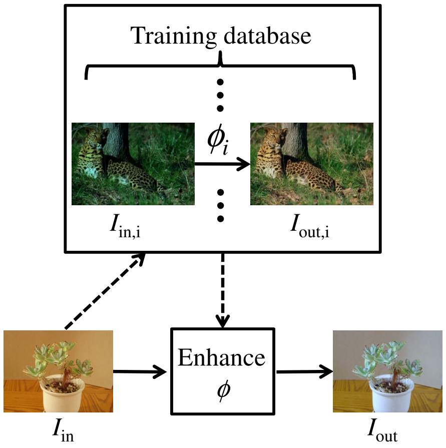


Figure 1. Basic idea of our approach. The control parameters for image enhancement are represented by vector ϕ.

tem finds the closest images in the database and inherits their color and intensity distributions to produce an intermediate corrected image. It then finds the restoration operations that is closest to the intermediate corrected image. They assume that the database consists of images of high visual quality and that taking linear combinations of color distributions is appropriate. The validation is done quantitatively with those assumptions, without a user study. In addition, it is assumed that the restoration is general.

To make it easier for the user to train our system, we designed its interface so that the user need only click through a series of images that the user deem to be visually more pleasing. Our interface is similar to those of [17, 20]. Adobe Photoshop has a pre-defined set of "actions" (macros) that simplifies batch image enhancement, and allows the user to save new actions. However, it is not clear how these actions can be automatically customized for a given user.

3. System overview

Our personalized image enhancement framework is depicted in Figure 1. In the training phase, a database is constructed by presenting the user with a set of training images, and asking her to enhance each image using a simple and general user interface. The database stores the feature vector describing each training set image $I_{\text{in},i}$, along with a vector of enhancement parameters ϕ_i. Given a new input image, this database is then searched for the best matching image, and the corresponding enhancement parameters are used to perform the enhancement.

We used the following enhancement operators to compose ϕ: white balancing via changes in temperature and tint, and contrast manipulation via changes in power and S-curves. We first describe our enhancement operators.

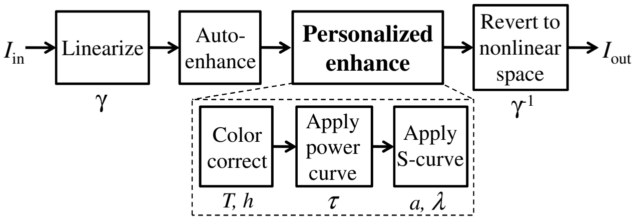


Figure 2. Image enhancement pipeline.

4. Image enhancement pipeline

Figure 2 shows our system's image enhancement pipeline. We approximate the processes of linearization and reversion to original nonlinear space with gamma curves with parameter $\gamma = 2.2$ and $\gamma^{-1} = 0.455$, respectively. RGB is linearized using $c_{\text{linear}} = c^{\gamma}$, where $c = R, G, B$ (normalized). The linearized RGB values are then color corrected and contrast enhanced, and finally "unlinearized" by applying the inverse operation.

The enhancement itself is performed in two steps: auto-enhancement, and personalized enhancement. The auto-enhancement step (Section 4.3) is necessary to handle bad quality photos that the system is not trained to handle. This step generates some kind of a baseline image that is then further adjusted using personalized enhancement.

In our work, we used three parameters associated with contrast (τ, λ, a) and two associated with color correction (T, h). We limit the number of parameters to five primarily to limit complexity, since the search space grows exponentially with the number of parameters.

4.1. Contrast curve specification

To manipulate the contrast, we use the power and S-curves.

Power curve: τ**.** This is equivalent to the gamma curve (but note that is kept separate from the gamma curve, as seen in Figure 2): $y = x^{\tau}$, with x and y being the normalized input and output intensities.

S-curve: λ, a**.** The S-curve is also commonly used to globally modify contrast. The formula we used to specify the S-curve is

$$y = \begin{cases} a - a\left(1 - \frac{x}{a}\right)^{\lambda} & \text{if } x \leq a \\ a + (1-a)\left(\frac{x-a}{1-a}\right)^{\lambda} & \text{otherwise} \end{cases}, \qquad (1)$$

with x and y being the normalized input and output intensities (see Figure 3(a)).

4.2. Color correction: temperature T and tint h

We color correct based on color temperature T and tint h, rather than applying a 3×3 diagonal matrix. The notion of

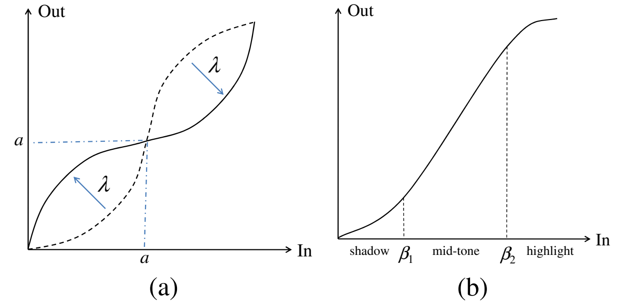


Figure 3. Two examples of contrast curves. (a) S curve to specify global contrast change. (b) Parameterized curve to specify shadow, mid-tone, and highlight regions. In our enhancement pipeline, we use (a).

temperature and tint is a more natural parameterization from the photographer's perspective. Also, we deal with only two instead of three parameters. (One can "normalize" the matrix so that the resulting luminance is unchanged, yielding two independent parameters, but these numbers are perceptually less meaningful.)

The color temperature is determined by comparing its chromaticity with that of an ideal black-body radiator. In practice, however, color temperature is relative to a standard, usually D65 standard, which is equivalent to 6500 Kelvin. The temperature curve is along the blue-yellow line. Unintuitive as it may seem, higher color temperatures (5000 K or more) are "cool" colors (green to blue), while lower are "warm" colors (yellow to red). Tint, however, is orthogonal to color temperature, and controls change along the green-magenta axis. The details on the process of color correction using T and h can be found in `http://www.brucelindbloom.com/`.

4.3. Auto-correction as preprocess step

Unfortunately, in practice, images have a very wide variance in quality. It is very difficult to sample the full range of quality. Our solution is to preprocess the input image using a set of auto-correction techniques. This is to bring it closer to a more acceptable quality, and have the personalized enhancement push it to a more *subjectively* pleasing result. Our version of auto-correction consists of auto-white balance followed by auto-contrast stretch.

Auto-white balance. We assume gray-world assumption for the brightest 5% of the pixels; this is a variant of [1]. More sophisticated techniques (such as [11]) may be used, but even then, limitations exist. This portion generates 3 parameters, each a color band multiplier.

Auto-contrast stretch. We find the intensity I_0 such that a maximum of 0.4% of the pixels are darker or as dark as I_0, and intensity I_1 such that a maximum of 1% of the pixels are brighter or as bright as I_1. We then linearly stretch the brightness so that I_0 is mapped to 0 and I_1 is mapped to 255 (with appropriate clamping at 0 and 255). This portion generates 2 parameters, a shift and a scale, similarly applied to all color bands.

5. Learning components of system

At the heart of the system are the *learning* components. They enable the system to first collect training data about user preferences and then, given a new (unseen) image, attempt to provide the best set of enhancement parameters that match the user's taste. More specifically, during the training phase, the user explores the set of possible enhancements for each training image via a novel interface and selects the one she prefers the best. Once the training is completed, to enhance any unseen test image, the system first finds the training image most similar to it. The enhancement parameters associated with the closest training image are then applied to that test image.

However, there are three issues we need to resolve in order to make the above components work: (1) How to measure similarity between images in order to match the test image to the ones in the training set? (2) How to determine a good training set of images? (3) How to enable the non-expert user to easily explore the space of possible enhancements of the training images in the training phase? We tackle all of these three issues below.

5.1. Learning a distance metric between images

We propose to solve the first problem with distance metric learning. Note that there are many different metrics to compare images. For example, one can compare images according to their color distribution, intensity, contrast, texture and even aspect ratio. However, our task is to determine similarity such that it correlates well with the enhancement parameters. In particular, we would like images that require similar enhancement parameters to be similar to each other; consequently, our goal is to learn a distance metric that enforces such regularization.

We construct the distance metric between two images as a linear combination of 38 different individual distances. These distances include differences of histograms in each of the channels of RGB, HSL, and intensity space using different ways to measure histogram distances (L_2 norm, symmetric KL-divergence, smoothed L_2 norm). We also consider the difference in intensity histogram of gradient images in both x and y directions. Finally, in addition to the distribution-based distances, we also consider distances that use entropy in each of the channels as well as the size and the aspect ratio of images [7]. Formally, the parametric form of the distance metric between images I_i and I_j is

$$D_{\alpha}^{\text{images}}(i,j) = \sum_{k=1}^{38} \alpha_k D^k(i,j). \tag{2}$$

Here, α are the parameters in the linear combination and $D^k(\cdot)$ denotes the individual distances computed.

In order to learn the right distant metric, we start by first randomly selecting 5,000 images from the LabelMe dataset [19] (`http://labelme.csail.mit.edu/`). Assume that we knew enhancement parameters for all these images. Then, we would seek to learn a distance metric D_{α}^{images}, parameterized with α, such that it minimizes the following objective:

$$\alpha^* = \arg\min_{\alpha} \sum_{i,j} ||D_{\alpha}^{\text{images}}(i,j) - D^{\text{params}}(i,j)||^2, \tag{3}$$

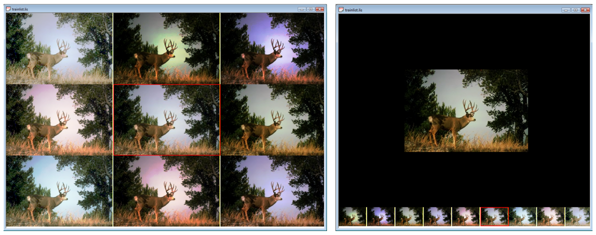
where $D^{\text{params}}(i,j)$ is the L_2 norm distance between the enhancement parameters for I_i and those for I_j.

The objective (3) examines all pairs of images and measures how much the distance in image space differs from the distance in the parameter space. Thus, minimizing this objective leads to finding an appropriate distance function that reflects how far two images should be in terms of their enhancement parameters. Note that this objective is convex; the unique optimum can thus be easily found by running a gradient descent procedure. In our implementation, we used an implementation of limited memory BFGS [15], which is a quasi-Newton optimization algorithm to optimize the function. Note that the optimization procedure needs enhancement parameters for the 5,000 images and it is not feasible for any user to find these parameters manually. Instead, we used the automatically-determined parameters from our auto-enhance component (Figure 2, Section 4.3) to estimate $D^{params}(\cdot)$. Although not personalized, these parameters do capture how these images are likely to be enhanced, and thus it is assumed that the learnt distance metric using these proxy parameters leads to reasonable estimates of the relevant distance metric in the image space.

5.2. Selection of training set

Ideally, we need a rich enough training data set that samples well the set of possible input images and appropriate enhancement transformations. Unfortunately, including a large number of images in the training set is not feasible, because each user would have to go through a tedious training phase. Thus, we seek to answer the following question: if the typical user is willing to spend the time enhancing, say, only 25 images, what should those 25 images be?

We answer this question of selecting the training images as a sensor placement problem [13]. Each instance can be thought of as a possible sensor location, where a probe is placed in order to "sense" the space of images. Given a sensor budget (the number of training images), we choose a set that can provide maximum information about rest of the images. Intuitively, our approach builds upon the observation that instances that are close to each other can be expected to share similar properties, including the appropriate enhancement parameters. Thus, our aim is to select a subset of instances that share the highest mutual information with the rest of the high-dimensional space and is therefore most representative of the full set.



(a) (b)

Figure 4. Two versions of UI. (a) Our interface, where the central image is currently being considered while the images at the periphery are different modified versions. (b) A version where the same images are arranged in linear order. The subject can flip between (i.e., blink compare) any pair of enhanced image candidates by mousing over them.

Selecting such *maximally informative* set is NP-complete, and a greedy and myopic selection procedure maximizing mutual information gain is the best possible approximation [13]. We therefore employ this approach. Formally, we consider a Gaussian Process (used in [13]) perspective with covariance function (alternatively, kernel function):

$$k_{ij} = e^{-\frac{D_{\alpha}^{\text{images}}(i,j)}{mean(D_{\alpha}^{\text{images}}(:))}}.$$

Here $D_{\alpha}^{\text{images}}(i,j)$ is the learnt distance as described in Section 5.1. Note that because of the negative exponent, k_{ij} measures similarity between images I_i and I_j and ranges from 0 (infinitely far apart) to 1 (at the same location). The matrix $K = [k_{ij}]$ therefore encodes similarity and how well information flows in the space. Including image I_i in the selected set provides a good representation of instances where k_{ij} is high, but is unrepresentative of the portion of the space where k_{ij} approaches zero. At each step in the greedy selection, given the existing set of selected instances S and unselected instances U, we select the instance that maximizes the gain in mutual information on the remainder of the unselected instances:

$$I^* = \arg\max_{I_i \in U} f(i),$$

$$\text{where: } f(i) = \text{MI}(U-i; S \cup i) - \text{MI}(U-i; S) = \frac{1 - \mathbf{k}_{S,i}^{\text{T}} K_{S,S}^{-1} \mathbf{k}_{S,i}}{1 - \mathbf{k}_{U-i,i}^{\text{T}} K_{U-i,U-i}^{-1} \mathbf{k}_{U-i,i}}.$$

$\mathrm{MI}(X, Y)$ is the mutual information between X and Y [2]. The second line in the equation follows from the Gaussian Process models, where $K_{S,S}$ is the similarity matrix among images in S, $K_{U-i,U-i}$ is the similarity matrix among U excluding I_i, and $\mathbf{k}_{S,i}$, $\mathbf{k}_{U-i,i}$ are each similarity vectors between I_i and images in S and $U - i$ (U excluding I_i), respectively. The numerator characterizes the similarity of I_i to the previously selected instances and the denominator characterizes the similarity of I_i to other currently unselected instances. Choosing i that maximizes the ratio selects an instance that is farthest from previously selected instances (most dissimilar to those previously selected) but also central to the unselected instances (most representative of those currently unselected).

The above procedure results in a ranking of all of our 5,000 images. The top 25 images (shown in Figure 5) are selected as the training set. We found 25 to be a reasonable number, resulting in a training phase of 25–45 minutes, in our user studies.

5.3. Enabling seamless browsing for training

We developed a user interface that allows a user to seamlessly explore the space of possible enhancements for each of the training images and indicate his/her preferences. This user interface, shown in Figure 4, was designed to allow non-expert users to quickly steer a given training image towards the most subjectively pleasing version, using only a few mouse clicks. The design of our interface is similar to those of [17, 20]; the user has the choice of toggling between the 3 × 3 tiled view in (a) and the linear view in (b). While the tiled view provides an overview that reflects underlying structure of enhanced image space, the linear view enables a higher resolution view with an ability to do pairwise comparisons across candidates.

The basic idea in both views is to give a UI to the subject where she can inspect and browse the space of images resulting from all possible combination of enhancement parameters. Since the number of images resulting from all possible combinations is prohibitively large, we use tools from machine learning to reduce the number of images shown to the user at any given time, and lay them out in a manner that reflects some structure in the space of enhanced images.

More specifically, given an input image, we first apply all the operators described in Sections 4.1 and 4.2 to sample a neighborhood of the images in the enhancements space. This neighborhood is sampled by considering 3 parameter settings for each of the 5 enhancement parameters: a negative step, a zero step, and a positive step. All possible combinations of these steps yield $3^5 = 243$ candidate images. From these images we select 8 representatives and display them to the user, as described below. The current, unmodified image is displayed as well, resulting in a total of 9 choices. The user then selects the version that he/she likes the best by clicking on it, and the entire procedure is repeated around the selected image. The process continues until the user selects the unmodified image. The user is also able to control the step size used to generate the variations.

Determining what images to show. Given the 243 images, the system must decide what images to show the user. The goal is to select 8 images which best represent the different variations present in the set of 243 images, and provide a holistic overview of the corresponding neighborhood in the space of enhancements. We use the same sensor selection procedure described in Section 5.2 to accomplish this task.

Tiling images in user interface. Once the 8 representative images are generated, the next step is to determine how to lay them out in the UI. Ideally, the layout should effectively convey the underlying structure of the space. It should be noted that due to a handful of operators and parameters we use, it is expected that the images lie on a subspace or a manifold, and we use dimensionality reduction to project them onto a 2D space. To this end, we can use any standard dimensionally reduction technique including PCA. However, since many of the parameters in our image enhancement pipeline are correlated, a linear projection might not be useful. Hence, our choice is non-linear dimensionality reduction with ISOMAP [21] to project the images onto a 2D space. The resulting 2D projections from ISOMAP should expose some structure to the user and visualize the distribution of the images on a plane where the two axis correspond to the highest degree of variations on the manifold. The selected 8 images can be then tiled around the current (unmodified) image according to the quadrant they lie in the 2D projection. Figure 4(a) shows example of such tiling.

6. Experiments

We ran two experiments; the first to train the system, the second (pairwise comparison user studies) to test the significance of personalizing image enhancement. For both experiments, we used the same 14 subjects (colleagues and acquaintances in our organization), 9 males and 5 females. None of the subjects are experts in photography. Let us denote the set of subjects as $B = \{b_i, i = 1, ...14\}$, with b_i being the ith subject. Given the training data, a test image can be enhanced by first finding the most similar training image (using the metric described in Section 5.1) and then using the enhancement parameters associated with that training image.

6.1. Pairwise comparison user studies

In the pairwise comparison user studies, we ask the same set of subjects B to perform pairwise comparisons amongst

Figure 5. Selected 25 training images.

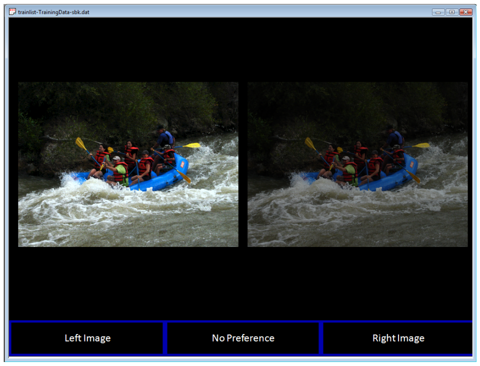


Figure 6. Interface for pairwise comparison.

the following versions:

1. Original
2. Auto-corrected using Picasa (`http://picasa.google.com/`)
3. Auto-corrected using Windows Live Photo Gallery (`http://download.live.com/photogallery`)
4. Proposed technique (auto-corrected followed by personalized enhancement using the subject's preferences)
5. Proposed technique (auto-corrected followed by personalized enhancement using a "median" subject's preferences)

The "median" subject is selected by first computing sum-of-squared distance over the enhancement parameters associated with all the training images for subject b_i and those for the other subjects $B - \{b_i\}$. The "median" subject is the subject from $B - \{b_i\}$ with the median distance.

The following pairs were compared: 4-1, 4-2, 4-3, 4-5, and 2-3. Note that the order of pairs and placement of first and second images within each pair shown to the subject are randomized. The interface for the pairwise comparison is shown in Figure 6. The subject selects either "left," "right," or "no preference." The user study was conducted in the same room under the same lighting condition and same display as the first (learning) phase.

We used 20 images (see Figure 7) for our pairwise comparison user study. These images were selected from a large group of images taken by our colleagues and from the web. They were selected based on two criteria: (1) there is a reasonable variation in scenes and lighting condition, and (2) they look like they require some form of color correction and/or contrast enhancement. The images are all different from those used for training. The pairwise comparison portion took each subject between 10-20 minutes to complete.

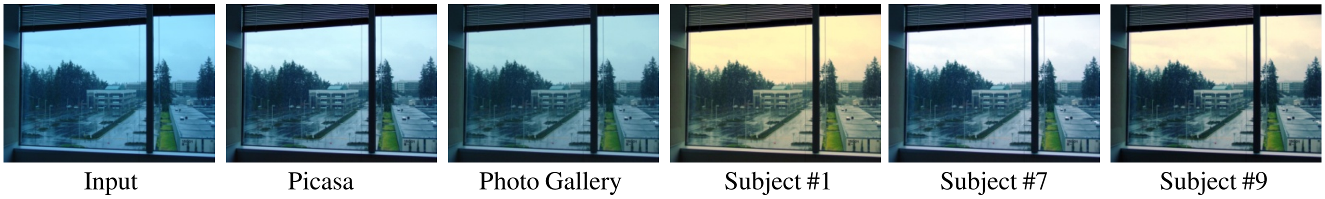
6.2. Results of pairwise comparison user study

The results of the user study are summarized in the two graphs in Figure 9. We first looked at the result of pairwise comparisons across different subjects. In particular, we consider the 20 images in the test data, we look at the percentage of times a participant choose a system in each of the comparisons (Subject vs. Input, Subject vs. Median, Subject vs. Picasa, Subject vs. Photo Gallery, and Picasa vs. Photo Gallery). In summary, for every subject we have the percentage of times (out of 20 images) that participant choose a system over another for each pairwise task and can analyze the data to see significant effects.

Figure 9(a) graphically shows the means of these percentages averaged over all the 14 subjects (error bars denote the standard error). We further did significance analysis using Wilcoxon [22] signed ranked test for each of the pairwise conditions, and found significant differences in scores between Subject vs. Input ($p < 0.01$) and Subject vs. Median ($p < 0.01$). These results indicate that the participants overwhelmingly selected their own model (mean = 59.30%) over the input image (mean = 30.70%), suggesting that the procedure did help enhance the input image. More interestingly, the participants also preferred their own model (mean = 57.10%) instead of the Median model (mean = 28.20%), suggesting that the preferences among participants vary quite a bit and provides further evidence that the personalized image enhancement is required instead of just a single "auto-enhance" functionality. Figure 8 shows the different versions for one of the test images.

Finally, the participants showed some bias in preference towards the personalized enhancement when compared to existing commercial systems (mean = 50.35% vs 39.30% against Photo Gallery and mean = 44.65% vs 41.75% against Picasa). While the difference was significant for Photo Gallery ($p < 0.05$), it was not for Picasa. Note that the proposed system only uses 5 simple enhancement operations; we hypothesize that the personalized enhancement

Figure 7. The 20 images used for our pairwise comparison user study.



Input Picasa Photo Gallery Subject #1 Subject #7 Subject #9

Figure 8. Sample image versions. From left to right: input, Picasa auto-enhanced, Photo Gallery auto-enhanced, enhanced using preferences of Subject 1, 7, and 9, respectively. In this case, the subject-enhanced images were favored by the respective subjects. Notice the significant differences in color across the different versions.

has the potential to further improve upon the performance of the existing commercial systems by using them as a "pre-process" step and then overlaying the personalization.

Next, we also compared the number of subjects that preferred one system over another in the five pairwise comparisons. Specifically, we consider that a participant prefers one system over another when he/she chose more images corresponding to the former system than the latter. Figure 9(b) graphically shows the results. To judge the significance of the numbers we did an exact binomial sign test and the results indicate that the subjects personalized model was significantly preferred over the input image ($p < 0.01$) and the Median model ($p < 0.01$).

7. Discussion

In this work, we focus on user-specific image enhancement instead of correcting arbitrarily bad images (which span a large space). We see the role of personalization as *refining* the output of a "generic" automatic enhancement module. Our improvement over Picasa is marginal but statistically insignificant; our "generic" (i.e., non-user-specific) portion of our system is likely to be less effective than that of Picasa. However, note that our back-end "generic" version can be easily replaced with Picasa or Windows Live Photo Gallery, thus potentially providing further opportunities for enhancements.

In our system, only 5 parameters are used for personalizing image enhancement. While results do show in favor of personalized versions, it is likely that more parameters are needed to optimize the personalization effect. There are other important features, such as auto-cropping, filters (e.g., sharpening), other types of contrast enhancement (e.g., shadow-midtone-highlight curve specification as shown in Figure 3(b)), and optical correction (e.g., vignetting and barrel distortion correction).

Clearly we need to balance the initial training set size with user effort in personalizing the system. We are looking into incremental training: the user manually corrects if the system-generated output is inadequate, and the correction is used to update the user's preference database.

Another interesting direction is the possibility of clusters in the training data. The existence of clusters would suggest a simpler approach to personalization: we generate "preset" preferences (each set corresponding to a cluster) and select the best "preset" preference based on just a few images the user corrects. Here, studies on a much larger scale would be required.

8. Concluding remarks

We present an end-to-end pipeline that covers training, and user interface issues and testing. In designing our system, we apply two learning techniques. We use a distance metric learning technique to allow us to match images that have similar enhancement requirements. We use an active sensor selection technique for two important functions: (1) to select the training set from a larger database of images, and (2) during the training phase, to determine a subset of images that best represent variation in the spectrum of possible candidates in the space of enhanced images. Results suggest that while general techniques for enhancing images are helpful, image enhancement has a strong personalization component, which should help improve the (subjective) quality of images even further.

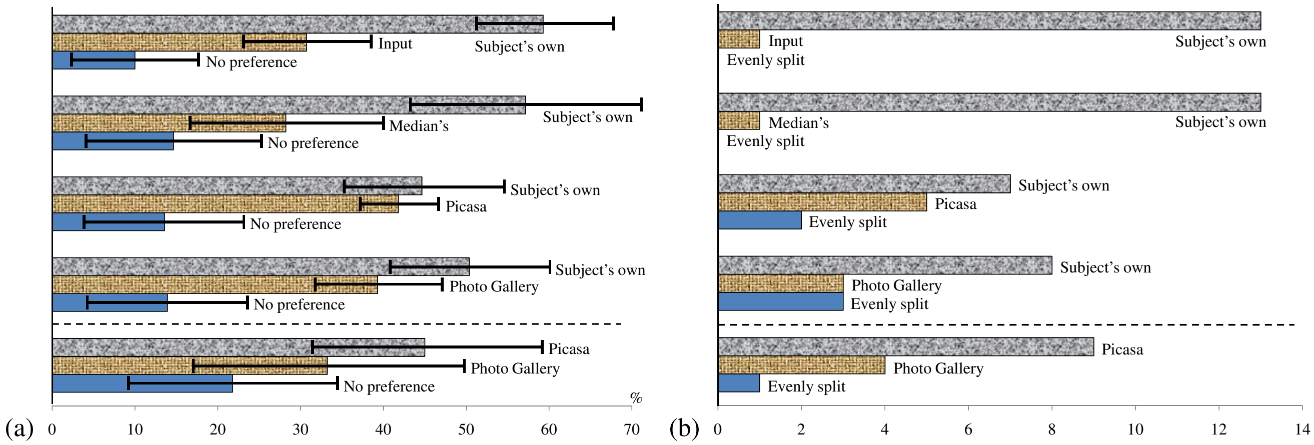


Figure 9. Results of pairwise comparison user study comparing the inputs and subjects', "median" (see text) subjects', Picasa auto-enhanced, and PhotoGallery auto-enhanced versions. The first triplet in each graph, for example, is the result of comparing the subject's enhanced version with the input. (a) Graph comparing mean frequency of image version favored (in percent over 20 images, with standard deviation bars). (b) Graph comparing number of subjects predominantly favoring the image version. Here, each triplet add to 14 subjects.

Acknowledgments

We would like to thank Yan Ke for discussions on photography and Ed Cutrell for advice on conducting user studies.

References

[1] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310:337–350, 1980.

[2] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. New York: Wiley, 2006.

[3] K. Dale, M. K. Johnson, K. Sunkavalli, W. Matusik, and H. Pfister. Image restoration using online photo collections. In *ICCV*, 2009.

[4] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Processing*, 54(12):3736–3745, 2006.

[5] H. Farid. Blind inverse gamma correction. *IEEE Transactions on Image Processing*, 10(10):1428–1433, October 2001.

[6] H. Farid and A. C. Popescu. Blind removal of lens distortions. *Journal of the Optical Society of America*, 18(9):2072–2078, 2001.

[7] J. Fogarty, D. Tan, A. Kapoor, and S. Winder. Cueflik: Interactive concept learning in image search. In *Conference on Human Factors in Computing Systems (CHI)*, 2008.

[8] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, pages 56–65, March/April 2002.

[9] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *CVPR*, June 2008.

[10] A. Gijsenij and T. Gevers. Color constancy using natural image statistics. In *CVPR*, June 2007.

[11] E. Hsu, T. Mertens, S. Paris, S. Avidan, and F. Durand. Light mixture estimation for spatially varying white balance. *ACM Transactions on Graphics and SIGGRAPH*, 27(3):article 70, August 2008.

[12] S. B. Kang. Automatic removal of chromatic aberration from a single image. In *CVPR*, 2007.

[13] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 2008.

[14] S. Lin, J. Gu, S. Yamazaki, and H.-Y. Shum. Radiometric calibration using a single image. In *CVPR*, volume 2, pages 938–945, 2004.

[15] D. C. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528, 1989.

[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009.

[17] J. Marks, B. Andalman, P. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: a general approach to setting parameters for computer graphics and animation. *ACM SIGGRAPH*, pages 389–400, 1997.

[18] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, November 2003.

[19] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, May 2008.

[20] L. Shapira, A. Shamir, and D. Cohen-Or. Image appearance exploration by model based navigation. In *Computer Graphics Forum, Eurographics*, 2009.

[21] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

[22] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.