Learning Photographic Global Tonal Adjustment with a Database of Input / Output Image Pairs

Vladimir Bychkovsky
MIT CSAIL

Sylvain Paris
Adobe Systems Inc.

Eric Chan
Adobe Systems Inc.

Frédo Durand
MIT CSAIL

http://graphics.csail.mit.edu/fivek_dataset

Abstract

Adjusting photographs to obtain compelling renditions requires skill and time. Even contrast and brightness adjustments are challenging because they require taking into account the image content. Photographers are also known for having different retouching preferences. As the result of this complexity, rule-based, one-size-fits-all automatic techniques often fail. This problem can greatly benefit from supervised machine learning but the lack of training data has impeded work in this area. Our first contribution is the creation of a high-quality reference dataset. We collected 5,000 photos, manually annotated them, and hired 5 trained photographers to retouch each picture. The result is a collection of 5 sets of 5,000 example input-output pairs that enable supervised learning. We first use this dataset to predict a user's adjustment from a large training set. We then show that our dataset and features enable the accurate adjustment personalization using a carefully chosen set of training photos. Finally, we introduce difference learning: this method models and predicts difference between users. It frees the user from using predetermined photos for training. We show that difference learning enables accurate prediction using only a handful of examples.

1. Introduction

Adjusting tonal attributes of photographs is a critical aspect of photography. Professional retouchers can turn a flat-looking photograph into a postcard by careful manipulation of tones. This is, however, a tedious process that requires skill to balance between multiple objectives: contrast in one part of the photograph may be traded off for better contrast in another. The craft of photo retouching is elusive and, while a plethora of books describe issues and processes, the decision factors are usually subjective and cannot be directly embedded into algorithmic procedures. Casual users would greatly benefit from automatic adjustment tools that can acquire individual retouching preferences. Even professional photographers often wish they could rely more on automatic adjustment when dealing with large collections in a limited amount of time (*e.g.* a wedding photoshoot). Photo editing packages offer automatic adjustment such as image histogram stretching and equalization. Unfortunately, such simple heuristics do not distinguish between low- and high-key scenes or scenes with back-lighting and other difficult lighting situations.

We propose to address the problem of automatic global adjustment using supervised machine learning. As with any learning approach, the quality of the training data is critical. No such data are currently available and previous work has resorted to rule-based, computer-generated training examples [10]. Another alternative is to use on-line photo collections such as Flickr, e.g. [4]. However, since only the adjusted versions are available, these methods require unsupervised learning. This is a hard problem and requires huge training sets, up to a million and more. Furthermore, it is unclear how to relate the adjusted output images to the unedited input [4]. This makes it impossible to train such methods for one's style, as a user would have to manually adjust thousands of images. To address these shortcomings and enable high-quality supervised learning, we have assembled a dataset of 5,000 photographs, with both the original RAW images straight from the camera and adjusted versions by 5 trained photographers (see Figure 1 for an example).

The availability of both the input and output image in our collection allows us to use supervised learning to learn global tonal adjustments. That is, we learn image transformations that can be modeled with a single luminance remapping curve applied independently to each pixel. We hypothesize that such adjustments depend on both low level features, such as histograms, and high-level features such as presence of faces. We propose a number of features and apply a regression techniques such as linear least squares, LASSO, and Gaussian Process Regression (GPR). We show a good agreement between our predicted adjustment and ground truth.

While a brute-force supervised learning approach is convenient for learning a single "neutral" rendition corresponding to one of the photographers hired to retouch our dataset, it necessitates a large investment in retouching thousands of photographs. In order to accommodate a greater variety of styles without requiring thousands of examples for each style, we build on Kang et al. [10]: We seek to select a small number of photographs so that adjustments on new photos can be best predicted from this reduced training set. A user then only needs to retouch this small set of training photographs to personalize future adjustments. We show that our dataset together with our new features provide significant performance improvement over previous work.

The above-mentioned approach still requires users to retouch a predefined set of images that come from the database, as opposed to their own photos. We want to alleviate this and learn the adjustments of a user directly from arbitrary photographs. We hypothesize that there is a correlation between users. We use a two-step approach, where the prediction from our neutral style trained on thousands of images is combined with a method that learns on-the-fly the difference between neutral and the new style of adjustment. The learning is further helped by the use of a covariance matrix learned on the large database. We show that this can enable good predictions using only a handful of user-provided adjustments.

1.1. Related Work

Photo editing software such as Adobe Photoshop enables arbitrary pixel modifications with a plethora of tools. In contrast, we want to focus on the fundamentals of photo rendition, and in particular the adjustment of brightness, contrast, and a tonal response curve in the spirit of the zone system [1, 2]. These are the type of edits that motivated lighter-weight packages such as Adobe Lightroom and Apple Aperture that provide simpler, parametric control over photo renditions and enable a much faster workflow. These packages offer automatic photo adjustment tools but unfortunately, little is known about the actual techniques used. As far as we can tell, many of them apply simple rules such as fixing the black and white points of the image to the darkest and brightest pixels. Although this may work on simple cases, these approaches fail on more complex examples for which a photographer would apply more sophisticated modifications.

There are numerous books about image adjustment, e.g. [1, 2, 6, 13]. However, their suggestions cannot be directly converted into an algorithm. The guidelines rarely provide actual values and often rely on the subjective judgment of the viewer. The recommendations can also be contradictory when several elements are present in a photo. To adjust images, photographers make decisions and compromises. Our dataset provides numerous examples of these complex adjustments and thereby enables their systematic modeling using supervised machine learning.

Tone mapping algorithms [15] compress the tonal range of HDR images. By default, these techniques produce a generic rendition. Although the achieved look can be controlled by parameters, these are set by the user. Bae et al. [3] and Hertzmann et al. [9] adjust photos using a model provided by the user. In comparison, we focus on fully automatic adjustment.

Several methods, e.g. [5, 12], have been proposed to assess the visual quality of photos. However, using these techniques in the context of adjustment would be nontrivial because these methods strongly rely on the image content to discriminate the good images from the bad ones. In comparison, an adjustment modifies the rendition while the content is fixed. From this perspective, our dataset offers an opportunity to revisit the *style-vs-content* problem studied by Tenenbaum and Freeman [17].

Gehler et al. [7] have shown that supervised learning can be a successful approach to inferring the color of the light that illuminates a scene. Our work shares the machine-learning approach with this paper but focuses on tonal adjustments.

Dale et al. [4] restore damaged photos using a corpus of images downloaded from Internet. The dataset is huge but only the final rendition is available. In comparison, our images are not damaged and we seek to improve their rendition, not repair problems such as over- and under-exposure. More importantly, our dataset provides both input and output images.

Kang et al. [10] personalize the output of an automatic adjustment method by using a small but predetermined set of examples from their collection. Given a new image, their approach copies the adjustment of nearest user-retouched example. To determine the similarity metric between photos, Kang et al. use metric learning and sensor placement [11]. However, metric learning requires a large training set to be effective. On that issue, Kang et al. note that "it is infeasible for any user to find these parameters manually because no large collection of photos including untouched input and retouched versions is available," which motivates their generating synthetic training data using gray-highlight white balance and histogram stretching. In contrast, we collected adjustments from trained photographers. Also, enable users to train the system without a predetermined set of examples by learning the difference between photographers. Thus, we leverage our dataset while freeing the user from working on training images.

1.2. Contributions

A reference dataset We have collected 5,000 photos in RAW format and hired 5 trained photographers to retouched each of them by hand. We tagged the photos according their content and ran user study to rank the photographers according to viewers' preference.

Figure 1. On this photo, the retouchers have produced diverse of outputs, from a sunset mood (b) to a day light look (f). There is no single good answer and the retoucher's interpretation plays a significant role in the final result. We argue that supervised machine learning is well suited to deal with the difficult task of automatic photo adjustment, and we provide a dataset of reference images that enables this approach. This figure may be better viewed in the electronic version.

Global learning We use this dataset for supervised learning. We describe a set of features and labels that enable the prediction of a user's adjustment.

Sensor placement Our dataset enables sensor placement to select a small set of representative photos. Using adjustments made to these photos by new users we accurately learn preferences of new users.

Difference learning We show that predicting the difference between two photographers can generate better results than predicting the absolute adjustment directly, and that it can be used for learning users' preferences on-the-fly.

2. A Dataset of Input-Output Photographs

We have collected 5,000 photographs taken with SLR cameras by a set of different photographers. They are all in RAW format, i.e., all the information recorded by the camera sensor is available. We have made sure that the photographs cover a broad diversity of scenes, subjects, and lighting conditions. We then hired five photography students in an art school to adjust the tone of the photos. Each of them retouched all the 5,000 photos using a software dedicated to photo adjustment (Adobe Lightroom) on which they were extensively trained. We asked the retouchers to achieve visually pleasing renditions, akin to a postcard. The retouchers were compensated for their work. A visual inspection reveals that the retouchers made large modifications to the input images. Moreover, their adjustments are nontrivial and often differ significantly among the retouchers. Figure 1 shows an example of this diversity. We numerically evaluate these points with statistics computed in the CIE-Lab color space. The difference between the input photo and the retouched versions is 5.5 on average and can be as much as 23.7. And the average difference between the retouched version is 3.3 and the maximum is 23.5. For reference, the difference between white and black in CIE-Lab is 100. We also augmented the dataset with tags collected with Amazon Mechanical Turk to annotate the content of the photos. We also ran a user study in a controlled setting to rank photographers according to users' preference on a subset of our dataset.

We studied the dimensionality of the tone remapping curves that transform the input image luminance into the adjusted one. We found that the first three principal components explain 99% of the variance of the dataset and that the first component alone is responsible for 90% of it. This is why we focus our learning on this component.

3. Learning problem setup

3.1. Labels

We express adjustments as a remapping curve from input luminance into output luminance, using the CIE-Lab color space because it is reasonably perceptually uniform. The curve is represented by a spline with 51 uniformly sampled control points. We fit the spline to the pairs of input-output luminance values in a least-squares sense.

We want to avoid bias due to the type of camera used for a photo and the skill of the particular photographer. In particular, different camera metering systems or a user's manual settings might result in different exposures for a given scene. This is why we normalize the exposure to the same baseline by linearly remapping the luminance values of each image so that the minimum is 0 and the maximum 100.

We focus on learning the first PCA coefficient of the remapping curves, which is a good approximation to the full curve (§ 2). At run time, we predict the new adjustment by reconstructing the full curves and interpolating linearly between samples.

3.2. Features

The features that we use for learning are motivated by photographic practice and range from low level descriptions of luminance distribution to high-level aspects such as face detection. Before computing features, we resize the images so that their long edge is 500 pixels.

▷ *Intensity distributions:* Photographers commonly rely on the distribution of intensities as depicted by a log-scale histogram to adjust the tonal balance. We consider the distribution of the log-intensity $\log(R+G+B)$ and compute its mean and its percentiles sampled every 2%. We also evaluate the same percentiles on two Gaussian-convolved versions of the photo ($\sigma = 10$ and $\sigma = 30$) to account for the tonal distributions at larger scales.

▷ *Scene brightness:* We hypothesize that scenes that are dark vs. bright in the real world might be adjusted differently. We evaluate the scene brightness as: $(\hat{Y} \times N^2)/(\Delta t \times ISO)$ where $\hat{Y}$ is the median intensity, N is the lens aperture number that is inversely proportional to the aperture radius, Δt is the exposure duration, and ISO is the sensor gain. This quantity is proportional to the light reaching the camera sensor and assumes that there is no filter attached.

▷ *Equalization curves:* Photographers tend to use the entire available intensity range. Histogram equalization is a coarse approximation of this strategy. We compute the corresponding curve, i.e., the cumulative distribution function (CDF) of the image intensities, and project it on the first 5 PCA components of the curve

▷ *Detail-weighted equalization curves:* Detailed regions often receive more attention. We represent this by weighting each pixel by the gradient magnitude, and then project the weighted CDF onto the first 5 PCA components of the curve. We estimate the gradients with Gaussian derivatives for $\sigma = 1$, $\sigma = 100$, and $\sigma = 200$ to account for details at different scales.

▷ *Highlight clipping:* Managing the amount of highlight that gets "clipped" is a key aspect of photo retouching. We compute the label values that clip the following fraction of the image: 1%, 2%, 3%, 5%, 10%, and 15%.

▷ *Spatial distributions:* The fraction of highlights, midtones, and shadows are key aspects discussed in the photography literature. However, their percentage alone does not tell the whole story, and it is important to also consider how a given tone range is spatially distributed. We split the intensity range in 10 intervals. For each of them, we fit a 2D spatial Gaussian to the corresponding pixels. The feature value is the area of the fitted Gaussian divided by the number of pixels in the given tone range. We also use the xy coordinates of the center of the Gaussian as a feature representing the coarse spatial distribution of tones.

▷ *Faces:* People are often the main subject of a photo and their adjustment has priority. We detect faces and compute the following features: intensity percentiles within facial regions (if none, we use the percentiles of the whole image), total area, mean xy location, and number of faces.

We also experimented with other features such as local histograms, color distributions, and scene descriptors but they did not improve the results in our experiments.

3.3. Error Metric

We use the L_2 metric in the CIE-Lab color space to evaluate the learning results because this space is perceptually uniform. The difference between white and black is 100, and distance of 2.3 corresponds to a just-noticeable-difference (JND) [16]. Since we focus on tonal balance, we measure the difference in luminance between the predicted output and the user-adjusted reference. We evaluate our learning methods by splitting our dataset into training on 80% dataset and testing on the remaining 20%.

4. Learning Automatic Adjustment

We consider two practical cases. First, we aim for reproducing the adjustment of a single photographer given a large collection of examples. In the second case, we seek to learn adjustments from a specific user from a small set of examples, assuming that we have access to a large collection of examples by another photographer. To validate our approach, we compare it to the recent method of Kang et al. [10] because it tackles similar issues and requires only minor changes to work on our dataset.

4.1. Predicting a User's Adjustment

In this scenario, we have a large dataset of examples from a single user and we learn to adjust images similarly to this photographer. This is useful for a camera or software company to train an automatic adjustment tool. We tested several regression algorithms: linear regression as a simple baseline, LASSO as a simple and still efficient technique [8], and Gaussian Processes Regression (GPR) as a powerful but computationally more expensive method [14]. LASSO performs a linear regression on a sparse subset of the input dimensions. We trained it using 5-fold cross-validation on the training set. GPR has been shown to have great abilities to learn complex relationships but is also significantly more expensive in terms of computation. To keep the running time reasonable, we trained it only on 2,500 randomly selected examples.

Comparison to Metric Learning For comparison, we implemented a variant of the method by Kang et al. [10] so that it uses our dataset and handles a single user. We used the user's adjustments instead of computer-generated data for learning the metric. We kept sensor placement unchanged, i.e., we select the images that maximize the mutual information with the user's adjustments. The nearest-neighbor step is also unaltered except that we transfer the tonal curve extracted from our data instead of Kang's parametric curve and white balance.

Results We selected Retoucher C for our evaluation because the high ranking in our user study. Using labels from Retoucher C we compared several options: the mean curve of the training set; the metric learning method using 25 sensors as recommended by the authors of [10]; least-squares regression (LSR); LASSO set to keep about 50 features; and GPR. The prediction accuracy is reported in Table 1. Regression techniques perform significantly better than other approaches. We also computed the leave-one-out performance of metric-learning method: 9.8, which means that it is limited independently of the number of sensors that we select. This is further confirmed in the next section.

mean	metric learning	LSR	LASSO	GPR
13.2	11.5	5.2	4.9	4.7

Table 1. LAB error of several methods (lower is better) when predicting a user's adjustment. For reference, not adjusting the photos at all produces an error of 16.3.

Figure 2 shows error CDFs for automatic image adjustment methods. In this figure, all methods were evaluted on the same test set of 2,500 photos. Our method was trained on 2,500 examples. The metric computation (a variant of [10]) used all 5,000 examples, but the nearest neighbor prediction was done only using 2,500 training examples. Commercial methods were not trained on our dataset and are shown for reference only.

Data versus Covariance GPR proceeds in two steps. During training, it optimizes the hyper-parameters of a covariance function so that it best explains the training set. At run time, it uses this covariance function to drive the combination of some of the training curves. To find out whether the performance of GPR comes from the covariance or from the training data, we did the following comparison. First, we trained the GPR covariance on the whole training set of 2,500 photos but used only small number n of example curves at run-time for prediction. We also trained the covariance with only n images and used the same n images for prediction, practically reducing the size of the training set. In the two tests, the run time data are the same, but in the first case, the covariance function comes from a rich training set while in the second case, it comes from a small set. Figure 4 shows that using the well-trained covariance function

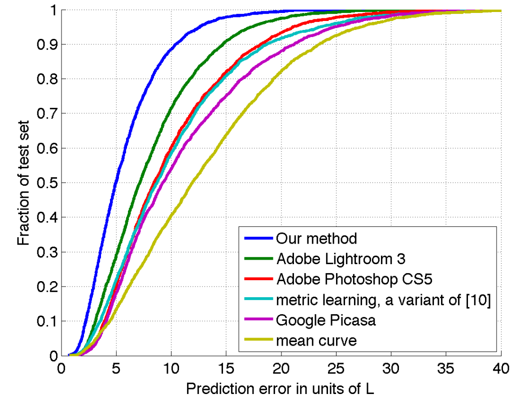
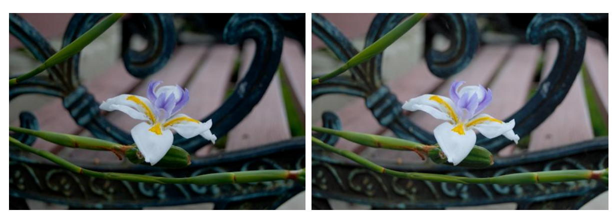


Figure 2. Error CDFs of automatic photo adjustment methods (higher is better). An error of 2.3 L units corresponds to 1 JND (just noticeable difference). For visual calibration see Figure 3. Lightroom, Photoshop, and Picasa were not trained on our dataset and are shown for refernce only.

(a) Expert rendition (b) Prediction with error 0.8

(c) Expert rendition (d) Prediction with error 5.5



(e) Expert rendition (f) Prediction with error 7.6

(g) Expert rendition (h) Prediction with error 10.6

Figure 3. Sample prediction results for our method provided for visual calibration of error values in Figure 2.

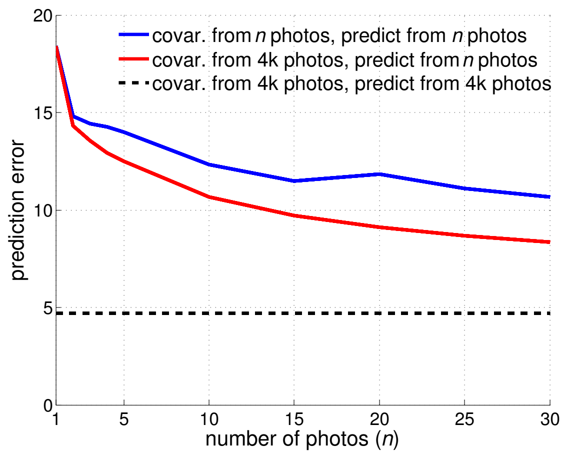


Figure 4. Using pre-trained covariance function improves the accuracy of prediction when only a few examples are available. In the above example directly learning from 30 images results in the same error as learning from 10 images and using pre-trained covariance.

yields significantly better prediction given the same small number of run-time data. This highlights the importance of the covariance function in the prediction process since it models the structure of the photograph space. We build upon this insight in the following sections.

4.2. Transferring a User's Adjustments

The technique described in the previous section is suitable for off-line training. However, adjusting 5,000 images requires several weeks of work, normal users cannot reasonably train this algorithm for their own style. In this section, we leverage the fact that we already have a large dataset $\mathcal{L}$ of 5000 images adjusted by the *reference* retoucher to enable learning from only a small set of examples $\mathcal{S}$ by a new photographer.

4.2.1 Experimental Setup

To evaluate our approach, we implemented the following algorithm. We run GPR on the large set $\mathcal{L}$ to compute a covariance function. Akin to Kang et al. [10], we use sensor placement [11] to select a small set $\mathcal{S}$ of images to be adjusted by the new photographer. For the covariance matrix needed to compute the mutual information, we use $\Sigma_{\mathcal{L}}$ (see [11] for detail). To predict the new photographer's adjustment on an unseen photo, we use the covariance function trained on the large set $\mathcal{L}$ to run GPR interpolation on the labels of the small set $\mathcal{S}$.

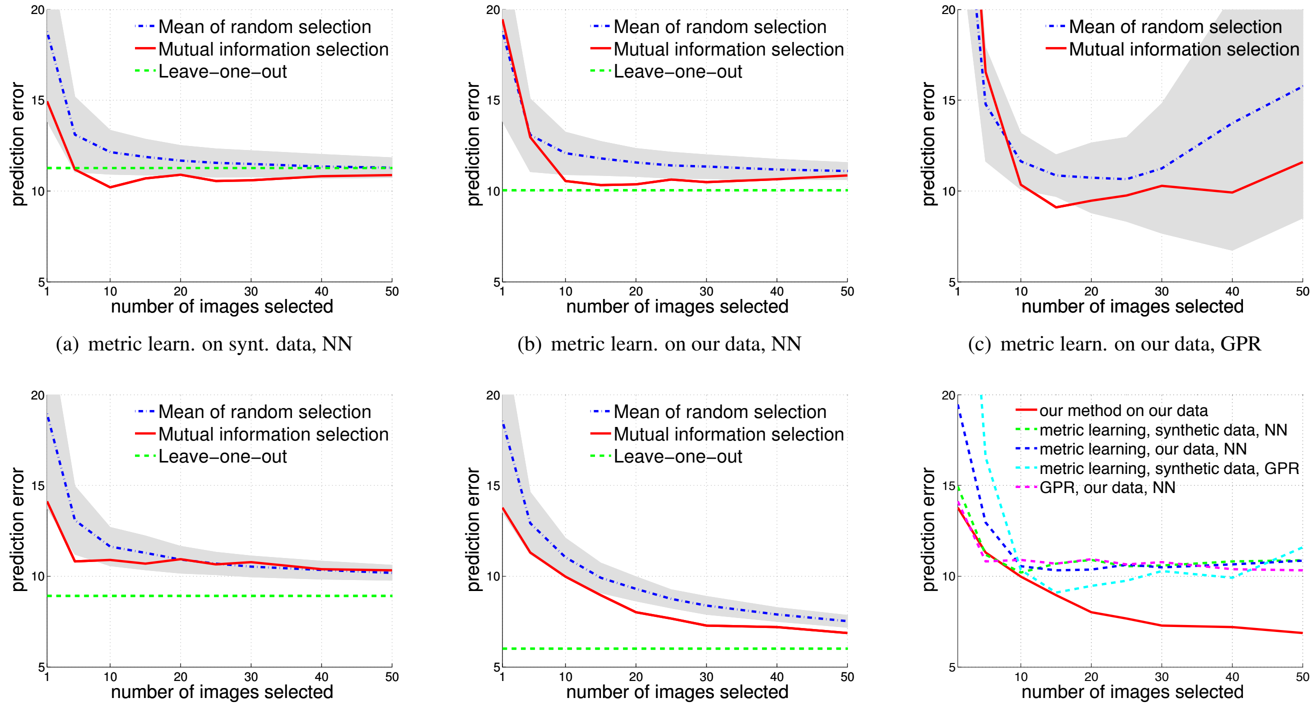
For comparison, we also implemented the method of Kang et al. We reproduced the automatic adjustment procedure that generates 4D vectors for each image of $\mathcal{L}$. We implemented the photo similarity functions that are proposed, and linearly combined them to approximate in a least-squares sense the L_2 distance on the 4D coefficient vectors. We ran sensor selection [11] to select $\mathcal{S}$ using the described covariance matrix. Given an unseen image, we search its nearest neighbor in $\mathcal{S}$ according to the learned metric, and apply its tone curve onto the new image. We also implemented variants to evaluate specific aspects. We trained the metric of Kang et al. on a photographer's curve instead of the original synthetic data. We also replaced nearest-neighbor search by GPR based on the covariance matrix used for sensor placement.

4.2.2 Results

Figure 5 reports the results for several options. For each scenario, we plot the accuracy as a function of the size of $\mathcal{S}$. In this figure, we compare a random selection with sensor placement selection; we also indicate the leave-one-out bound for reference. For all options but ours, the accuracy quickly reaches 10 and then plateaus (a, b, and d) or degrades (c). Using our dataset instead of synthetic data improves the leave-one-out performance (a vs. b). A similar improvement happens when using the metric learned with GPR instead of metric learning using a least-squares fit [10] (a vs. d). Using GPR with a covariance matrix optimized with metric learning yields poor results when $\mathcal{S}$ grows (c). In all cases, sensor placement using mutual information produces better results on average than a random selection. Although comparisons with Picasa have a limited scope because Picasa is not trained on any data, our results are consistent with the findings of Kang et al.: the difference between Picasa and their method using 25 images is below 1 (11.4 vs 10.6), which is marginal. In comparison, our approach performs significantly better than Picasa with an improvement of almost 4 (11.4 vs 7.6). Most importantly, our tests show that using GPR with our dataset yields results equivalent to other options up to 10 images, and performs significantly better than them for larger sizes of $\mathcal{S}$, producing an accuracy of about 7.6 with 25 images, instead of about 10.6 for the other techniques, i.e. an improvement on the order of 30% (f).

4.3. Difference Learning

The method described in the previous section reduces the number of training examples to a few tens. However, new users may prefer to train the system using their own photos instead adjusting a predefined set of example to train the system. In this section, we explore the scenario where the preferences are learned on-the-fly using adjustments on random pictures for training. Instead of learning the adjustment of the new photographer directly, we propose to learn the difference between the reference photographer and the new adjustment. For a new photo, we first predict the reference adjustment and then predict its difference with the new photographer's version. Our experiments described in the results section show the benefits of difference learning.



(a) metric learn. on synt. data, NN (b) metric learn. on our data, NN (c) metric learn. on our data, GPR

(d) GPR metric on our data, NN (e) GPR metric (our data), GPR **our method** (f) summary (lower is better)

Figure 5. Performance of various options to predict Retoucher C's adjustments using a small set $\mathcal{S}$ of his examples and a large set $\mathcal{L}$ of examples, either synthetic or from Retoucher D. We report the prediction error in CIE-Lab units as function of the size of $\mathcal{S}$. We plot the accuracy the sensor-placement selection (in red), of a random selection (average in blue, up to one standard deviation in gray), and of a leave-one-out bound (in green). See text for detail.

Our Approach We first trained GPR on the large training set $\mathcal{L}$. Then, we predict the reference curves for each photo of the small training set $\mathcal{S}$ and compute their difference with the curves of the new photographers. This gives a series of *adjustment offsets*. Given a new photo, we first predict the reference adjustment $\mathbf{r}$ using the covariance trained on $\mathcal{L}$ and the adjustments in $\mathcal{L}$. We also predict an adjustment offset $\mathbf{o}$ using the $\mathcal{L}$ covariance and the offsets computed on $\mathcal{S}$, and add it to the reference adjustment $\mathbf{r}$. Finally, we apply this combined adjustment $\mathbf{r} + \mathbf{o}$ to the photo.

Results Figure 6 shows that our approach using Retoucher E for $\mathcal{L}$ and Retoucher C for $\mathcal{S}$. In this case, Retoucher E and Retoucher C adjust photos similarly and transfer learning as described in the previous section (§ 4.2) does not predict C's adjustments better than using GPR directly (§ 4.1) using only E's photos. That is, the curves predicted using of E's data only are already a good approximation of C's adjustments, and transfer learning is unable to improve over this baseline. In comparison, our difference learning approach yields better predictions than this baseline, even if the available photos of C are randomly selected. On average, as few as 3 examples photos are enough to produce better results. Although the crossing point may depend on the considered photographers, we believe that the ability of difference learning to learn preferences from only a few examples and its accuracy make it highly practical.

5. Conclusion

We have built a high-quality reference dataset for automatic photo adjustment, which addresses a major need and will enable new research on the learning of photographic adjustment. In particular, we include data from five different users to enable not only training but also comprehensive validation. We have demonstrated that our photo collection is a powerful tool to learn photo adjustment and study various aspects of it. We have shown that with high-quality data, supervised learning can perform better than existing techniques based on simple rules or synthetic training sets. We have also found that regression with our new set of image features outperforms previous methods. We have performed transfer learning and shown that our dataset enables better selection through sensor placement. We have also shown that difference learning enables preference learning in a on-the-fly context where the training photos are not predetermined. In addition to enabling these applications, our dataset proves invaluable for validation.

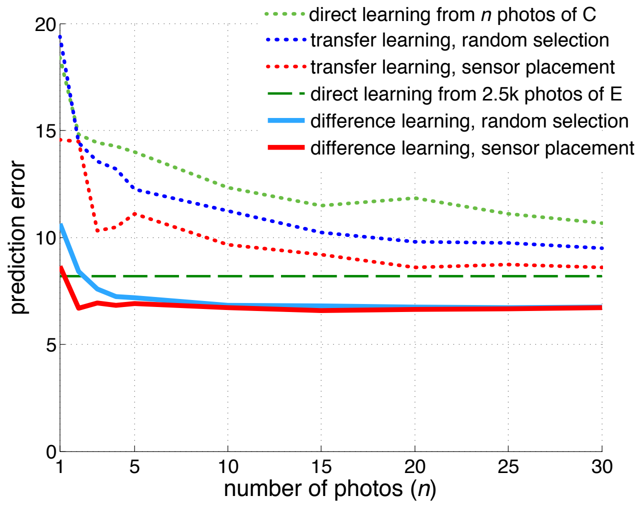


Figure 6. Several strategies to predict Retoucher C's adjustments from only n of his or her photos. We can directly train GPR on these examples only but the predictions are poor (first plot from the top). To improve the results, we can use transfer learning and precompute the GPR covariance function using a large dataset by Retoucher E (§ 4.2). This significantly improves the result (second plot) and if the we can select which photos of Retoucher C are available, sensor placement further improves the result (third plot). However, in this case, C and E produce adjustments similar enough so that applying GPR directly on E's photos without using any data from C better predicts C's adjustment than the previously mentioned options (fourth plot). This means that if our system was trained off-line with E's photos, the previous options would not allow C to get predictions closer to his or her preferences. In comparison, learning differences between C and E (§ 4.3) yields better results. If the photos of C are random, the improvement starts when 3 or more of C's photos are available (fifth plot). If we can select the photos with sensor placement, two example photos are sufficient to see an improvement (bottom plot).

Acknowledgments

We are grateful to Katrin Eismann and Jeff Schewe for providing invaluable advice and for introducing us to the community of professional photographers. We thank Todd Carroll, David Mager, Jaime Permuth, LaNola Stone, and Damian Wampler for their incredible patience in creating the dataset. This work was supported in part by grants from Foxconn and NSF (0964004) and a gift from Adobe.

References

[1] A. Adams. *The Negative: Exposure and Development Basic Photo 2*. Morgan and Lester, 1948. 98

[2] A. Adams. *The Print: Contact Printing and Enlarging*. Morgan and Lester, 1950. 98

[3] S. Bae, S. Paris, and F. Durand. Two-scale tone management for photographic look. *ACM Transactions on Graphics*, 25(3):637 – 645, 2006. Proceedings of the ACM SIGGRAPH conference. 98

[4] K. Dale, M. K. Johnson, K. Sunkavalli, W. Matusik, and H. Pfister. Image restoration using online photo collections. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009. 97, 98

[5] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Studying aesthetics in photographic images using a computational approach. In *Proceedings of the European Conference on Computer Vision*, 2006. 98

[6] K. Eisman and W. Palmer. *Adobe Photoshop Restoration & Retouching*. New Riders Press, 2006. ISBN: 0321316274. 98

[7] P. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *Proceedings of the conference on Computer Vision and Pattern Recognition*, 2008. 98

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2009. ISBN: 0387848576. 100

[9] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340, New York, NY, USA, 2001. ACM. 98

[10] S. B. Kang, A. Kapoor, and D. Lischinski. Personalization of image enhancement. In *Proceedings of the conference on Conference on Computer Vision and Pattern Recognition*, 2010. 97, 98, 100, 101, 102

[11] 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*, 9:235–284, 2008. 98, 102

[12] Y. Luo and X. Tang. Photo and video quality evaluation: Focusing on the subject. In *Proceedings of the European Conference on Computer Vision*, 2008. 98

[13] D. Margulis. *Photoshop LAB Color: The Canyon Conundrum and Other Adventures in the Most Powerful Colorspace*. Peachpit Press, 2005. ISBN: 0321356780. 98

[14] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. 100

[15] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufman, 2010. 98

[16] G. Sharma. *Digital Color Imaging Handbook*. CRC Press, Inc., Boca Raton, FL, USA, 2002. 100

[17] J. B. Tenenbaum and W. T. Freeman. Separating style and content. In *Proceedings of the conference on Advances in Neural Information Processing Systems*, 1997. 98