Context-Based Automatic Local Image Enhancement

Sung Ju Hwang[1], Ashish Kapoor[2], and Sing Bing Kang[2]

[1] The University of Texas, Austin,TX, USA
sjhwang@cs.utexas.edu
[2] Microsoft Research, Redmond, WA, USA
{akapoor,sbkang}@microsoft.com

Abstract. In this paper, we describe a technique to automatically enhance the perceptual quality of an image. Unlike previous techniques, where global statistics of the image are used to determine enhancement operation, our method is local and relies on local scene descriptors and context in addition to high-level image statistics. We cast the problem of image enhancement as searching for the best transformation for each pixel in the given image and then discovering the enhanced image using a formulation based on Gaussian Random Fields. The search is done in a coarse-to-fine manner, namely by finding the best candidate images, followed by pixels. Our experiments indicate that such context-based local enhancement is better than global enhancement schemes. A user study using Mechanical Turk shows that the subjects prefer contextual and local enhancements over the ones provided by existing schemes.

1 Introduction

Recent advances in digital photography has made it possible for an amateur photographer to create professional-looking photos. Simple adjustments of color and contrast can be performed using photo retouch tools such as Adobe Photoshop. However, such manual adjustments do not scale up well with large collections of captured photos. An automatic image enhancement method would help resolve this problem; by "image enhancement," we mean improvement of image perceptual quality of images.

Most existing automatic enhancement techniques make use of global intensity transforms, either for color correction (white balancing) or contrast enhancement. For these global schemes, the mapping of color or intensity is one-to-one and is independent of pixel location or scene context. Such methods would not work well for images where different parts require different types of correction, e.g., the darker portions of an indoor scene requires higher contrast adjustment and different color correction than the window, or it is desirable to emphasize the subject by enhancing contrast between the subject and its background [1].

In our paper, we propose a new automatic technique for *locally* enhancing images based on context at two levels: coarse (scene) and fine (pixel). For each point in input image, we find the best local matches in a training database that

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part I, LNCS 7572, pp. 569–582, 2012.

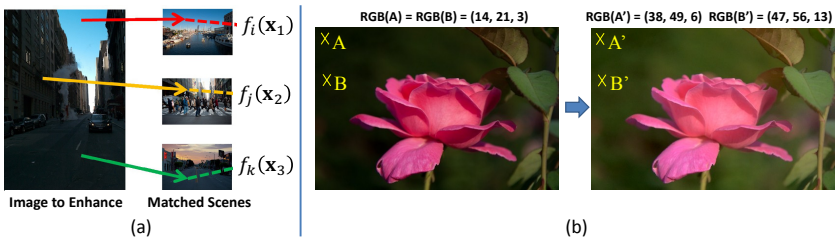


Fig. 1. (a) Illustration of basic concept. We enhance a given image by scene matching and use the recovered multiple candidates to combine enhancements. (b) An example highlighting the local enhancement. The pixels marked A and B in the input image have the same RGB values but are mapped to pixels A' and B' with different colors.

has original and enhanced image pairs. The matching relies on scene descriptors in addition to low-level image statistics. We then combine enhancement functions from multiple candidates over the entire image to generate an enhancement map with local variations, to enhance each pixel based on both scene and local context. Fig. 1(a) shows the concept of our idea where mapping for each input pixel is found by considering relevant matched scenes and then inheriting the enhancement operations from pixels in those matched images that are closest in terms of local context. Fig. 1(b) further highlights the effect of local enhancements achieves using the proposed scheme. In particular, we want to highlight that two identical pixels (marked as A and B) in the input image can get mapped to different values in the enhanced image. Note that the ratios (R/G, B/G) are also different for A' and B' in the output. This highlights our core contribution, namely an enhancement algorithm that considers local and contextual cues and that has both the elements of *tonal adjustment* and *color correction*.

2 Related Work

Various techniques have been proposed for automatic image enhancement. First, there are simple, heuristic-based methods, such as color correction based on the gray-world [2] or gray-edge [3] assumption, and enhancing the contrast by stretching the color histogram. While conceptually very simple, these strategies work reasonably well in practice. There are also Bayesian approaches (e.g., [4,5]) that model the illuminant as a random variable from a database of images, and then estimates it from the posterior distribution conditioned on image intensity data and/or feature descriptors such as direction filters.

In a similar vein, there are exemplar and learning-based methods that leverage databases of pre-enhanced images in order to enhance a new image. Kang et al. [6] describe such an examplar-based method. A metric is first learned for enhancement parameter similarity, then used to find the image in the training set that is closest to the given test image in terms of image enhancement. The

Fig. 2. Overview of the proposed coarse-to-fine local enhancement method. Enhancement maps are contrast enhanced for better visibility.

enhancement parameter associated with the "closest" image is then used. Since the database is trained for the use, the enhancement is personalized. Caicedo et al. [7] further exploited the personalization aspect of the method, and use collaborative filtering to discover clusters of user preferences.

While the methods of [6,7] are non-parametric, Bychkovsky et al. [8] took the parametric approach to reproduce the global tonal remapping function by training it on input-output image pairs, where the output images are enhanced by professional retouchers. However, this remapping function is only learned for the luminance channel, and is tested on a dataset where the amount of white balance is known, limiting its practicality. All these methods rely on mostly lower-level image statistics (such as color or intensity histogram [6]), and the enhancements are global. Bychkovsky et al. [8] use faces as one of their features, but do not exploit other high-level scene or object-level semantics.

Techniques for tonemapping high dynamic range (HDR) images [9] locally preserve detail in low dynamic range (LDR, or conventional 24-bit RGB) outputs for display. These techniques have also been used to enhance local contrast in single conventional RGB images. The approaches range from gradient-based [10], digital dodging-and-burning [11], and use of bilateral filtering [12] to tonal style remapping [13]. While local tonal adjustment is similar to our goal, regular tonemapping does not explicitly involve color correction (unless it is a byproduct of tonal adjustment). By comparison, we explicitly handle color correction as well, and the enhancement is done via example-based learning using scene descriptors and context (see Figure 1(b)).

The approach of Dale et al. [14] is the closest to ours. They used scene descriptors to match images that are similar in terms of scene context, and then use these images to search for the region match. Subsequently, each region in the input images is matched to the corresponding regions in the retrieved images using co-segmentation, and their color distributions are transferred. However, their method does not consider the similarity in enhancement space for those images while our method accounts for both scene semantics and enhancement parameters. Even though the transferred color distributions are local, the actual enhancement is done by fitting a *global* enhancement function to minimize the error between the globally enhanced image and the color-transferred image. This is to overcome the possible artifacts at region boundaries. Thus, it does not preserve one-to-many mappings that were present in the color-transferred image. On the other hand, our method produces a potentially different mapping operator for each pixel.

3 Approach

We view the image enhancement process as a search for local functions that would transform each i^{th} pixel y_i^o in the original image to an enhanced value y_i^e. The key idea is to learn such mapping from a training database that consists of input (original) and output (enhanced) image pairs. Since such training data set encompasses a wide variety of real images, we can hope to recover enhancement operations that are heteroscedastic and depend upon the local pixel context in addition to the image global statistics.

Fig. 2 provides an overview of the pipeline. Given an image to enhance, a retrieval step is first performed in order to focus on a subset of examples in the training set that are most similar to the input image in terms of enhancement requirements. Once such subset is retrieved, individual enhancement operation for each pixel in the input image is found by matching it to the pixels in the retrieved set of training examples. Finally, spatial smoothening is performed on the enhancement map in order to preserve spatial and visual regularity.

One of the key technical aspects of this work is the course-to-fine search, where the search over possible enhancements of each image is performed by first finding candidate images based on scene first and then over the space of pixels. Further, this matching is performed using different cues that incorporates knowledge at different levels of granularity, enabling us to consider both high-level statistics using the scene cues, and local semantics using low-level descriptors. Table 1 shows the list of features we used for each matching step. We describe the individual components of the pipeline in detail next.

Table 1. List of features used for matching

Retrieval type	Feature	Dimensionality
Image (scene)	L*ab color histogram	128
	GIST [15]	768
	Bag-of-words	512
Pixel	L*ab color	3
	Saliency map [16]	1
	y position	1
	Dense SIFT [17]	8

3.1 Retrieving Relevant Training Images

The first step in our pipeline is to find a set of candidate images from the training database that are most likely to be similar to the input test image in terms of enhancement requirements. (This is similar to the metric learning approach used in [6].) Instead of looking at the entire training database, the focus on the subset that is contextually similar to the test image is more likely to provide us with better enhancement choices. More specifically, given an input image $\mathbf{x}$ and a procedure to measure distances between images, the algorithm retrieves the top K images that are closest to the original images in the training set. In our work, $K = 10$.

Ideally, distance computation should be done such that the retrieved images reflect the content and enhancement requirement of the input images. Consequently, we rely on distance metric learning. In particular, we parameterize the distance between any two images $\mathbf{x}_i$ and $\mathbf{x}_j$ using an $n \times n$ matrix M_I that operates on the extracted n-dimensional feature vectors of the images:

$$d_{M_I}(\mathbf{x}_i, \mathbf{x}_j) = (\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j))^T M_I(\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)).$$

Here, $\psi(\mathbf{x}_i)$ is the extracted feature vector from image $\mathbf{x}_i$ and consists of a representation that capture the content and high-level image statistics. For our pipeline to work well, it requires that the distance-metric parameter M_I is set such that it considers images requiring similar enhancements closer to each other than the others that need different enhancement operations. Consequently, we learn the parameter M_I using a target distance function $d_t(\mathbf{x}_i, \mathbf{x}_j)$ that captures distance between enhancements applied to the images. Formally, given a training dataset of images and their enhancements, we seek to learn M_I by optimizing the following objective:

$$\arg\min_{M_I} \sum_{i,j} ||d_{M_I}(\mathbf{x}_i, \mathbf{x}_j) - d_t(\mathbf{x}_i, \mathbf{x}_j)||^2. \quad (1)$$

This objective looks at all pairs of images in the dataset. 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 (limited memory BFGS).

In our implementation, the image feature vector $\psi(\mathbf{x})$ is constructed by first concatenating L*ab histogram, GIST [15] descriptors, and bag-of-words (see Table 1) representation, and then reducing the dimensionality to 256 using PCA. Further, for the target distance we use $d_t(\mathbf{x}_i, \mathbf{x}_j) = \|g(\mathbf{x}_i) - g(\mathbf{x}_j)\|^2$, where $g(\cdot)$ denotes the enhancement function corresponding to the image. The enhancement function is represented as the histogram of $\delta y = y^e - y^o$ for each of the color components in the L*ab color space (20 bins per component), where y^o and y^e are pixels in the original and enhanced images, respectively.

3.2 Pixelwise Local Enhancement

Once the relevant subset of examples have been retrieved, the next stage in the pipeline is to individually estimate the "best" enhancement for every pixel in the test image. In particular, for every individual pixel we seek to map the input L*ab to output values that make use of local cues within the scene context defined by the set of retrieved images. Note that scene-level matching takes into consideration object-level context; subsequent to scene-level candidate selection, there is less ambiguity between object-level matches using the lower-level cues. For pixelwise search, the key idea is to compute a local feature representation of each pixel consisting of the original color and other local cues that describe local context (saliency, SIFT, and y position, see Table 1). This representation

is then used to measure similarity (or distance) between the pixels in terms of local context, thereby enabling us to identify pixels in the training subset that are most similar to the ones in the test image.

We again consider a distance metric parameterized by the matrix M_P that is used to compute distance between two pixels and retrieve similar pixels:

$$d_{M_P}(y_i, y_j) = (\tau(y_i) - \tau(y_j))^T M_P(\tau(y_i) - \tau(y_j)).$$

Here, $\tau()$ denotes feature representation of pixels. The parameter M_P can be learnt in a similar way before using an appropriate target distance. However, we expect that the distance metric would be sensitive to the scene context of the given input image. For example, consider an ocean scene with sky. We expect that there may be blue pixels in the ocean and sky regions that map to similar L*ab values but require different enhancements. Here, we expect that the y-position of a pixel is a more informative similarity feature. For images that contain people and faces, we expect saliency and appearance cues to be more useful. *Thus, it is very important to learn M_P on-the-fly and that we learn a distance metric over the space of pixels depending on the scene context of the given image.*

Given the set of retrieved training images, we can apply the same distance metric learning approach as before. As the number of pixels is relatively large, we use an efficient online metric learning method from [18] with the target distance computed as the sum of the L2 distance of enhancement pairs (y^o, y^e), computed as $d_{ij} = \|y_i^o - y_j^o\|_2 + \|y_i^e - y_j^e\|_2$. To further speed up the distance metric learning, we use 10,000 randomly sampled training points from each training example image. Since the search space for retrieving top k-nearest pixels is huge (10^6), we use locality sensitive hashing (LSH) as proposed in [19]. We set $k = 10$ in our work. The efficiency can be further improved using parallel processing as each search process is independent of each other. On a quad-core P4 2.4 GHz PC, it took about 1.5 minutes to perform the search (with parallel implementation) for an image of size 500×333.

Once the set of k-nearest training pixels $z_1, .., z_k$ to an input pixel y_i^o has been retrieved, we then recover the enhancement mapping using the weighted combination of the transformation observed on the retrieved pixels, i.e.,

$$f_i(\mathbf{x}) = y_i^o + \sum_{j=1}^{k} w_j(z_j^e - z_j^o).$$

z_j^o and z_j^e correspond to the original and enhanced training pixel, respectively. The core idea is to consider the transformations $(z_j^e - z_j^o)$ that were applied to the training pixels and apply a blending of those transformations to the input pixel y_i^o. Furthermore, the weights that determine the blending of the transformations are computed using a softmax function:

$$w_j = \frac{\exp(-d_{M_P}(y_i^o, z_j^o))}{\sum_{j'=1}^{k} \exp(-d_{M_P}(y_i^o, z_{j'}^o))}.$$

This softmax transformation results in blending that considers the distance of each of the retrieved pixels and assigns higher weights to nearby pixels. This transformation is performed on all the three L*ab channels independently on every single pixel, resulting in the output image $f(\mathbf{x}^o)$ corresponding to the test input $\mathbf{x}^o$. Note that since the proposed approach recovers individual transformations for each individual pixel, two pixels having same values in the original image can be mapped to different output values under our scheme. *This is the fundamental difference between our approach and existing global enhancement schemes where such one-to-many mappings are infeasible.*

3.3 Regularizing the Enhancement Map

Up to this point, the enhancement operation for each of the pixel is estimated independently. Also, the enhancement recovered for pixels can sometimes be inexact, resulting in enhancement maps with discontinuity artifacts. Thus, we need to spatially regularize to preserve the piecewise smooth characteristics of the input image. In particular, the key intuition behind this step is that the enhancement operation being applied to two pixels need to be similar if those pixels are spatially and perceptually close in the original image. In other words, similar operations need to be applied to neighboring pixels unless there is a sharp edge between them in the original image.

We achieve spatial smoothening using a model motivated by Gaussian Random Fields. Given the original input image $\mathbf{x}^o$ and its pixelwise enhancement map $f(\mathbf{x}^o)$, we find

$$\min_{\mathbf{y}^e} \left[\sum_{i}^{N} ||y_i^e - f_i(\mathbf{x}^o)||^2 + \gamma \sum_{i,j \in Nbr} L_{ij} ||y_i^e - y_j^e||^2 \right].$$

Here $\mathbf{y}^e$ denotes the final enhanced image with y_i^e and y_j^e as enhanced pixel values (L*ab color) to be estimated for pixels i and j respectively. Nbr refers to the set of 4-connected pairs of pixels. $f_i(\mathbf{x}^o)$ corresponds to the individual pixelwise enhancement and L_{ij} is a similarity measure between pixels i and j, computed using the value of input pixels y_i^o and y_j^o using radial basis function (RBF): $L_{ij} = \exp - \frac{\|y_i^o - y_j^o\|^2}{2\sigma^2}$. We found that the procedure worked well for $0.05 \leq \sigma \leq 0.5$; we set $\sigma = 0.1$.

This formulation consists of the unary term that represent affinity to solutions that are close to pixelwise local enhancements ($f(\mathbf{x}^o)$) and a pairwise term that enforces spatial and perceptual smoothness. The pairwise term sums over all pairs of spatially neighboring pixels (specifically, 4-connected neighborhood). Thus, minimization of such objective should result in solutions that are both smooth as well as similar to the local enhancements. Note that γ acts as a regularization parameter which controls the degree of smoothness: a high value of γ puts more weight on the pairwise component resulting in higher degree of smoothening. In our work, $\gamma = 400$ and was found using cross-validation.

In principle, our formulation is very similar to approaches in computer vision that are based on Markov Random Fields (MRFs) or Conditional Random Fields (CRFs); it also has connections to manifold-based learning techniques. The objective presented above is convex on $\mathbf{y}^e$, and we solve it using limited memory BFGS. For a 500×333 image, the optimization takes about 1.5 minutes.

4 Results

We validate our system via a user study. The database we use is the MIT FiveK dataset [8], which has 5,000 pairs of original and enhanced images. Each enhanced version was generated by professionals who adjusted the color remapping curve using the Adobe Lightroom software. Thus, the "ground truth" images are generated using global transformations. To preserve as much detail as possible, we use the exposure-normalized set (Catalog AsShotZeroed in the Adobe Lightrooom catalogue provided with the dataset) as the input set. The original raw image data has 16 bits per channel, but to reduce the computational complexity, we export them to 8 bits per channel 24-bit sRGB JPEG images, each resized so the longer edge is 500 pixels.

We generate two test sets for our experiments: (1) "Random 250": 250 randomly selected images from the entire dataset, and (2) "High Variance 50": 50 images manually selected by three individuals, based on the conjectured difficulty in performing global light/color correction. We compare our enhanced images against the input images, those enhanced using Picasa, and those enhanced using Dale et al.'s [14] method. We did not compare with [8] since their system adjusts only brightness/contrast.

In addition to reporting the L2 error, we ran a user study for comparative assessment of image perceptual quality. For the user study, we randomly selected 25 images from "Random 250" dataset and 25 images from "High Variance 50" dataset. Then, we presented random pairs, where one image is selected from a baseline containing either the input images, images enhanced using Picasa, images enhanced using Dale et al.'s method, and images enhanced using our method (3 pairs per image).

We used Mechanical Turk for our user study. We ran parallel mini-studies, with each study involving about 100 subjects, each subject looking at 3 different images, i.e., 9 comparisons. The order of images was randomized to avoid bias. Our only instruction to the subject is to select the image he/she thinks looks better (or "no preference"). 417 unique subjects responded, with each comparison taking 19.4 seconds on average.

4.1 Effect of Local Enhancement

Fig. 3 shows an example image enhanced using our method compared with the images enhanced globally (by Picasa, Dale et al.'s method, and a human expert). The given example is a typical case where global enhancement may not be adequate. The sky in the background is saturated both in intensity and color, while

the foreground train and the ground appears very dark. We want to enhance the foreground to appear brighter, but since the histogram already spans the full range, the global heuristic-based method (Picasa) cannot change the given image. Dale et al.'s method and a human expert enhance the foreground by increasing the overall brightness, but there is loss of sky color.

On the other hand, our method increases the brightness and contrast of the foreground, while the sky is only moderately changed. This is because the enhancements for the grass, tree, train, and sky originate from different regions and images in the training database. The bars on the right of each retrieved image is the weighting of each image, computed from the number of pixels retrieved from the image. It shows that we actually make use of all the 10 retrieved images to reconstruct the enhancement map for the given image; partial matches with different scene contexts are used to enhance each region differently. For the example shown in the first row of Fig. 4, the enhancement palette for the building comes from the house matched in the last row, while the trees and the grass benefit from matched images of trees and mountains. Also, for the last example of the outdoor night scene, some of the matched scenes are similar (either night scenes and/or having white buildings).

Dale et al.'s method also uses enhancements from multiple images by transferring colors from matched regions before finally fitting a global enhancement operation. However, it is less effective on the FiveK dataset because of the difficulty in finding a good matches. The dataset appears to be too diverse in lighting and color temperature variation; furthermore, their method does not consider the similarity in the enhancement space as ours does. See, for example, their result in Fig. 4.

The RGB remapping curve in Fig. 3 shows that our method is truly local. For our method, each RGB input is mapped to wider range of outputs[1]. The main advantages of our local enhancement method are: (1) We can deal with images that have spatially-varying lighting and color conditions; with global tonal adjustment, enhancing one region would have an adverse effect on another. (2) Using visual cues related to object importance such as saliency map, face, or objectness, we can emphasize the subject by enhancing those regions differently from the background. The example shown in the top row of Fig. 4 illustrates this point. Here, Picasa and Dale et al.'s method enhance the image so that details in the cloud are preserved, while the expert enhanced the foreground but lost some detail in the background. Our method is able to preserve the cloud details while enhancing the appearance of the grass and building.

4.2 Quantitative Evaluation

In this section, we report comparisons of results using L2 errors as a measure of performance. We also show that the L2 error does not necessarily correlate with perceptual image quality.

[1] The mapping for global enhancement is not exactly one-to-one due to quantization errors from resampling and color space conversion errors.

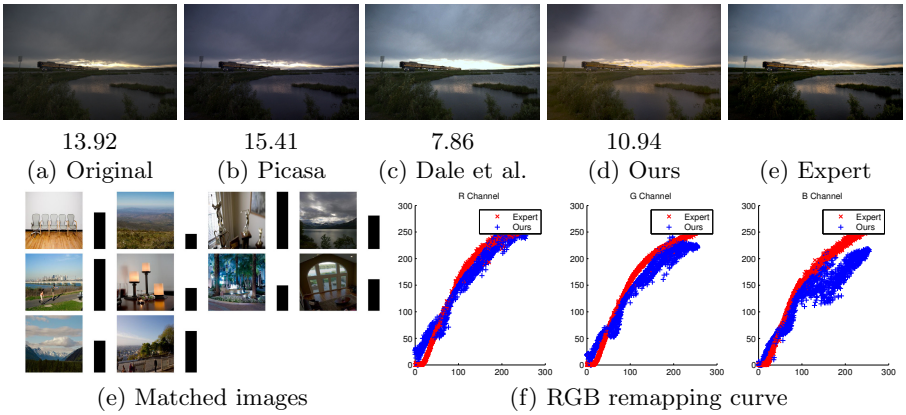


Fig. 3. Illustration of local enhancement effect. (a) Input image. (b)-(d) Image enhanced using different enhancement methods. The numbers are L2 errors with respect to the expert enhanced image. (e) Retrieved database images. The height of the bar to the right of each image represents its relative weight. (f) RGB remapping curve. For the expert enhanced images, the mapping is one-to-one, while our enhanced images are not, demonstrating local enhancement.

L2 Error on Globally Enhanced Dataset: The FiveK dataset was created through global tonal adjustment and our method performs local enhancement. By enhancing different parts of the image differently, we do not expect our method to perform significantly better than global methods for this experiment; however, since our method uses the globally enhanced training dataset, we still want to show that our method is able to learn and predict reasonably accurate enhancement functions based on this dataset. Table 2 shows the L2 error (in L*ab space) for the baseline systems and our method. The average L2 error we obtained for the "Random 250" set is slightly worse than Picasa, and on the "High Variance 50", the average L2 errors are similar. However, for the "High Variance 50" set, a significant amount of the errors comes from over-enhancing dark regions (e.g., second row Fig. 4). The L2 errors for Dale et al.'s method are higher than those for Picasa and ours, largely due to the difficulty of finding good matches in the small but visually diverse dataset. In the original experiments

Table 2. Comparisons of average L2 errors on two test sets, with the standard error for 95% confidence interval

Method	Random 250	High variance 50
Input	17.07 ± 0.93	14.85 ± 1.95
Picasa	13.39 ± 0.80	11.99 ± 1.46
Dale et al. [14]	20.43 ± 1.13	13.27 ± 1.67
Ours	15.01 ± 0.82	12.03 ± 1.27

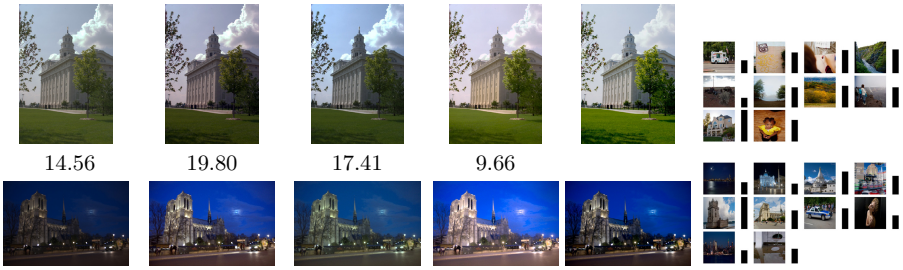


Fig. 4. Examples showing the advantage of the proposed method. The bar on the right side of each retrieved image denote the weighting for each image, calculated by computing the number of pixels the system used to enhance each image. The images with higher weights are the one that contribute most towards the enhancement of the input.

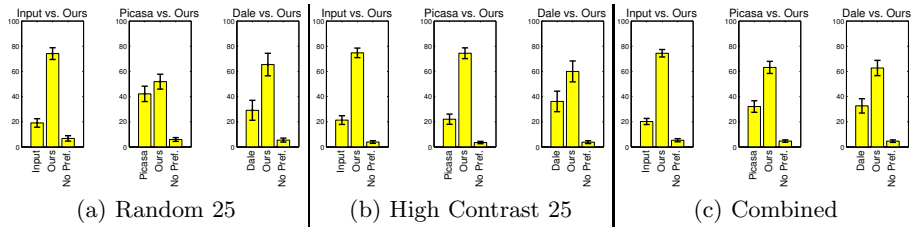


Fig. 5. User study results for the two different test sets. The graph denotes the proportion of user preference amongst the three choices. The error bars denote the range for 95% confidence. The proposed method outperforms global enhancement baselines by significant margins.

described in [14], the database used is large (about one million images) and contains only images of natural scenes.

L2 error is not a good predictor of quality: In most previous techniques, the quality of an automatic image enhancement method is evaluated by computing the L2 error between the enhanced image and the "ground truth" images, mostly out of convenience. However, it is not a reliable measure, since it is not directly related to the perceptual quality of the image. Fig. 6 illustrates this point; here, perceptual quality was measured based on the user study. The L2 error results in Table 2 are also not compatible with those of the user study shown in Fig. 5. For both testsets, while the mean L2 errors for our results are worse than those for Picasa, on average, more subjects prefer our results.

4.3 User Study

Since the L2 error is not a good predictor of quality, we rely on the user study to compare different enhancement methods. The results of our user study is summarized in Fig. 5. For the "Random 25" test set, our method received slightly

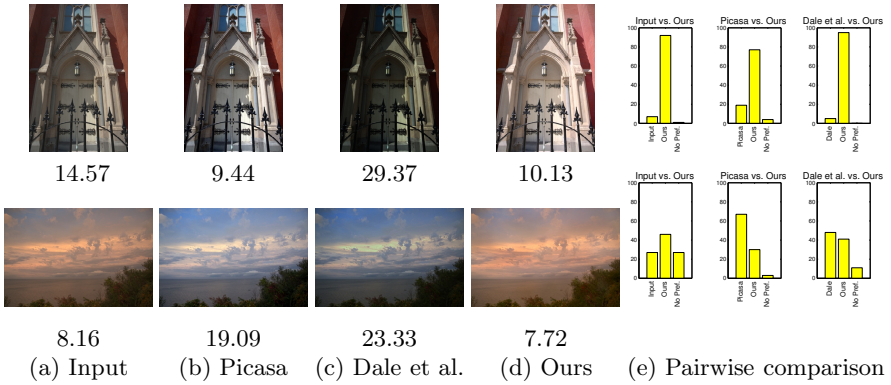


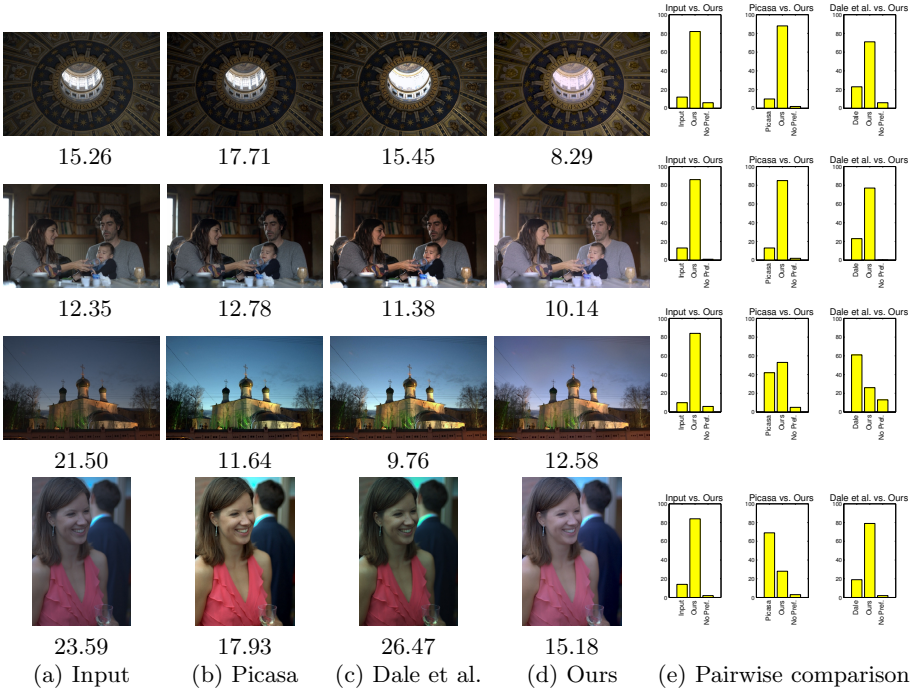
Fig. 6. Examples showing why L2 error is not totally reliable for predicting image quality. The numbers below the images are L2 errors. Top row: our method produced a larger L2 error but subjects prefer our result. Bottom row (also a failure example): our method produced a smaller L2 error but subjects prefer Picasa's and Dale's results.

more votes on average than both global enhancement baselines, for 16/25 images for Picasa, and 18/25 images for Dale et al.'s. The difference is not statistically significant for Picasa; in many cases, we received split votes for a given image. However, our method outperforms Dale's method significantly (for $p \leq 0.01\%$).

On the high-contrast images, the variance on the votes is significantly less, and subjects predominantly prefer the images enhanced by our method (more votes on 24/25 images for Picasa, and 18/25 images for Dale et al.'s). Dale et al.'s method does relatively well on these images, as it can brighten darker regions, though it also loses detail in the brighter regions (Fig. 7, first row). Picasa appears to be unable to effectively handle high-contrast images whose histograms already span the whole intensity range. We performed single-tailed t-test on the all 50 images using the number of votes on each image as the score, and the result confirms that the gains we get from using our method over Picasa and Dale et al. are statistically significant (for $p \leq 0.01\%$). This makes sense as finding a single global enhancement function for images with high variance could be difficult.

Fig. 7 shows some images used in the experiment, along with the votes received. The first two rows show examples where our method is preferred. For these images, it is not possible to effectively enhance those regions using a global enhancement method. The third row is a case where our method and a baseline received split votes. The split votes may be a result of random personal preferences, as it is hard to tell that one enhanced version is better than the other. In the last row, Picasa is preferred by more subjects; here our technique did not properly color correct the skin. Using a skin detector (not implemented) should handle this problem. Our framework is able to accommodate additional object or scene knowledge.

We expect our technique to fail when scene matching fails. For the image shown at the bottom of Fig. 6, even though our L2 error is significantly lower



(a) Input (b) Picasa (c) Dale et al. (d) Ours (e) Pairwise comparison

Fig. 7. Example images for the user study and the response. The numbers below each image is the L2 error. The first two examples are from the "High variance 50" testset, and the rest are from the "Random 50" testset.

than those for global enhancement baselines, subjects much prefer the other results. Here, it appears that our technique did not sufficiently color correct (although it is possible the sky has an orange hue).

5 Concluding Remarks

In this paper, we propose an automatic local image enhancement method based on a coarse-to-fine (image, then pixel) search for the optimal enhancement function for each pixel. Pixelwise local enhancement is accomplished by combining global enhancement functions from several matched images, searching the closest pointwise enhancement operator using local cues, and regularizing the resulting enhancement prediction map. Our method is truly local as both the prediction and enhancement is performed at the pixel level. We show the advantage of using our local enhancement method over global enhancement, through qualitative analysis and a user study. Extensions include having region-to-region matching between the image and pixel match step, and use of higher-level semantics.

Acknowledgment. We would like to thank Kevin Dale for providing code used in Dale et al. [14].

References

1. Luo, Y., Tang, X.: Photo and Video Quality Evaluation: Focusing on the Subject. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 386–399. Springer, Heidelberg (2008)
2. Finlayson, G., Trezzi, E.: Shades of gray and colour constancy. In: 1st Conf. on Color Imaging, pp. 37–41 (2004)
3. van de Weijer, J., Gevers, T., Gijsenij, A.: Edge-based color constancy. IEEE Trans. on Image Processing 16(9) (2007)
4. Gehler, P.V., Rother, C., Blake, A., Minka, T., Sharp, T.: Bayesian color constancy revisited. In: CVPR (2008)
5. Rosenberg, C., Minka, T., Ladsariya, A.: Bayesian color constancy with non-Gaussian models. In: NIPS (2003)
6. Kang, S.B., Kapoor, A., Lischinski, D.: Personalization of image enhancement. In: CVPR (2010)
7. Caicedo, J.C., Kapoor, A., Kang, S.B.: Collaborative personalization of image enhancement. In: CVPR (2011)
8. Bychkovsky, V., Paris, S., Chan, E., Durand, F.: Learning photographic global tonal adjustment with a database of input/output image pairs. In: CVPR (2011)
9. Reinhard, E., Ward, G., Pattanaik, S., Debevec, P., Heidrich, W., Myszkowski, K.: High Dynamic Range Imaging: Acquisition, Display, and Image-based Lighting, 2nd edn. Elsevier (Morgan Kaufmann) (2010)
10. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. ACM TOG and SIGGRAPH 21, 249–256 (2002)
11. Reinhard, E., Stark, M., Shirley, P., Ferwerda, J.: Photographic tone reproduction for digital images. ACM TOG and SIGGRAPH 21 (2002)
12. Durand, F., Dorsey, J.: Fast bilateral filtering for the display of high-dynamic-range images. ACM TOG and SIGGRAPH 21 (2002)
13. Bae, S., Paris, S., Durand, F.: Two-scale tone management for photographic look. ACM TOG and SIGGRAPH 25 (2006)
14. Dale, K., Johnson, M.K., Sunkavalli, K., Matusik, W., Pfister, H.: Image restoration using online photo collections. In: ICCV (2009)
15. Torralba, A.: Contextual priming for object detection. IJCV 53, 169–191 (2003)
16. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: NIPS (December 2006)
17. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT Flow: Dense Correspondence across Different Scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 28–42. Springer, Heidelberg (2008)
18. Jain, P., Kulis, B., Dhillon, I., Grauman, K.: Online metric learning and fast similarity search. In: NIPS (2008)
19. Jain, P., Kulis, B., Grauman, K.: Fast image search for learned metrics. In: CVPR (2008)