Image Enhancement and Reconstruction Using Photo Collections

Anthony Herwaldt
&
Qifeng Hu

Overview

- Enhancing and reconstructing a noisy image using a photo collection database
- Based of idea presented in paper below
  - Image restoration using online photo collections. K. Dale, M.K. Johnson, K. Sunkavalli, W. Matusik, and H. Pfister. International Conference on Computer Vision, 2009.

Figure 6: Performance across database size. We average errors across all 100×10 trials, for each database size. Moderately sized databases perform comparably to the full 1M image database for single-parameter estimation in (b) exposure and (c) contrast correction, while the 1M image database shows a significant improvement over smaller databases for (a) white balance correction.

(a) White balance (b) Exposure correction

Input Image

Color Transferred Image

Restoration

Result

Image Database

Matched Images

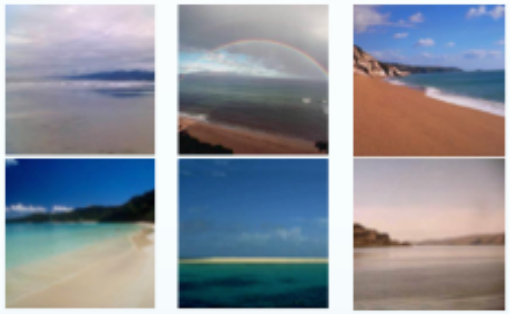


Image Database

- First, we simulated an image database using around 1,000 images divided into 5 categories
  - Flamingos, Sunflowers, Chairs, Beaches, and Mountains
- We did this in order to simplify the computation and save time



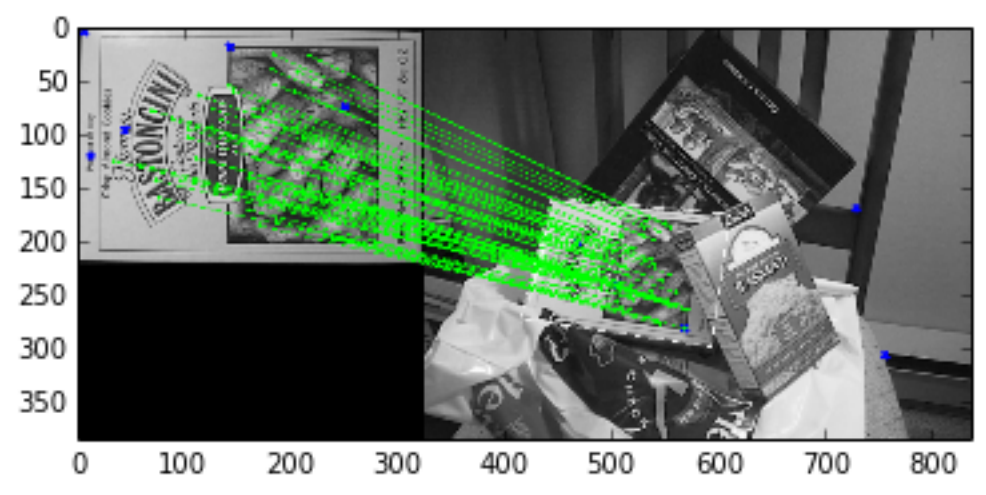


Query (Input) Image

- Then we found a picture of a similar image that is not one of the pictures in the database
  - We called this picture our “query” image, or the one that we want to find a match of in our database

Method: SIFT

- *Scale-Invariant Feature Transform*
- In order to compare images, and find a match with the Query image, we generated SIFT key points and descriptors
- The SIFT algorithm detects and describes local features in the image



FLANN Matcher

- *Fast Approximate Nearest Neighbor Search Library*
- For each image, after calculating their SIFT characteristics, we acquire the feature vectors, as in key points and descriptors
- By looking at the nearest neighbors around the target image in terms of Euclidean distance between different feature vectors, we pick the 10 closest neighbors

Color Transfer

- Take a source and target as input. Source contains the color space that will be transferred to target.
- Use L*a*b color space to transfer color. A small change in the L*a*b space produces an equal perceptual change in the human eye
- Split among the L*a*b channels and uses the mean and standard deviation of each channel to transfer color
  - Implementation and algorithm borrowed from Adrian Rosebrock at http://www.pyimagesearch.com/2014/06/30/super-fast-color-transfer-images/

Image Enhancement

- We use the color transferred image to develop parameters that are then used to enhance the original image
- White Balance

$$R(I(p),\theta) = \begin{pmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_g & 0 \\ 0 & 0 & \alpha_b \end{pmatrix} \begin{pmatrix} I_r(p) \\ I_g(p) \\ I_b(p) \end{pmatrix}$$

$$\alpha_k = \frac{\sum_{p \in k} I(p) I^c(p)}{\sum_{p \in k} I(p)^2}$$

- Contrast Enhancement

$$R(I(p),\gamma) = I(p)^{\gamma}$$

$$\gamma = \frac{\sum_p \omega_p (\log I^c(p))(\log I(p))}{\sum_p \omega_p (\log I(p))^2}$$

Image Enhancement (Cont.)

- Exposure Correction

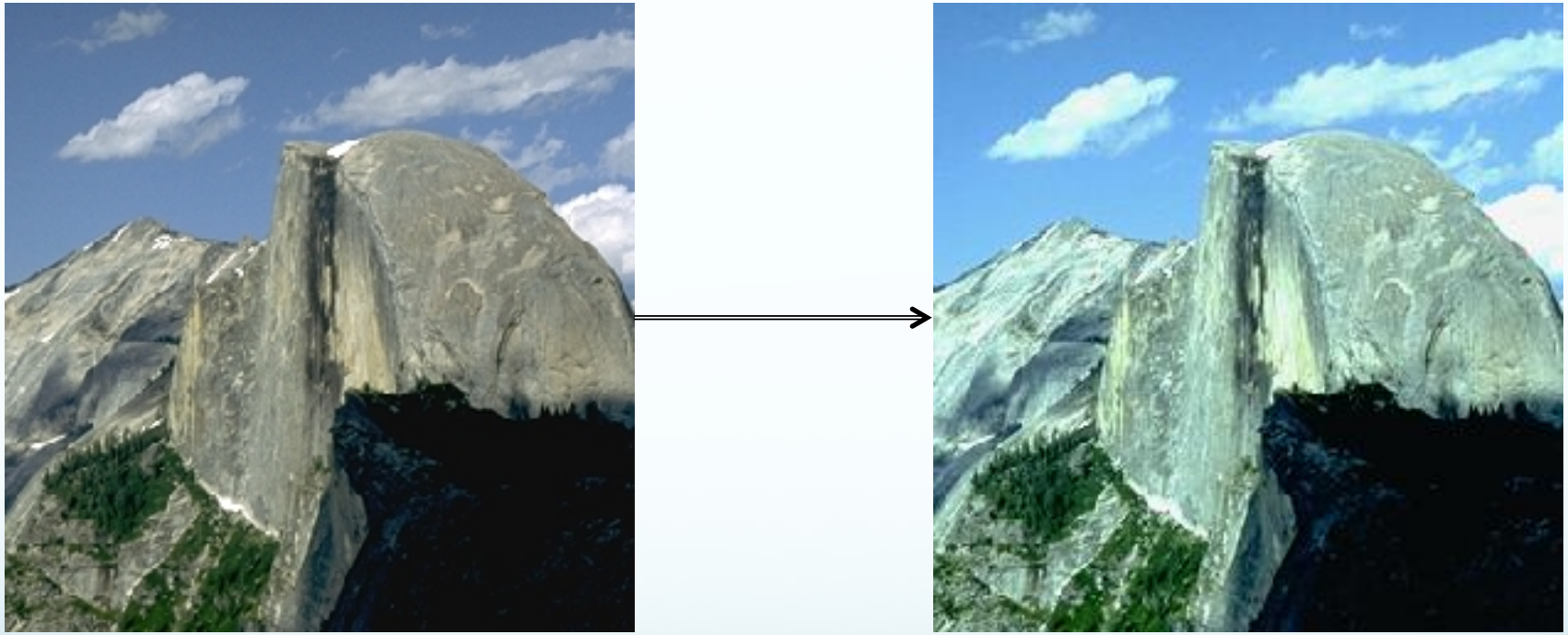
$$R(I(p), \alpha) = \alpha I(p)$$

$$\alpha = \frac{\sum_p I(p) I^c(p)}{\sum_p I(p)^2}$$

- Brightness Enhancement
  - $R(I(p),\beta) = I(p) + \beta$

Demo 1

Demo 2



Demo 3

White Balance Distortion with t =.5
t is parameter of distortion, between 0 (none) and 1 (max)

Input	Distorted	Output	MSE
			Distorted: 6820.1068 Output: 5342.476

Contrast Distortion with t = .8 (t range from .5 (black) to 2 (white))

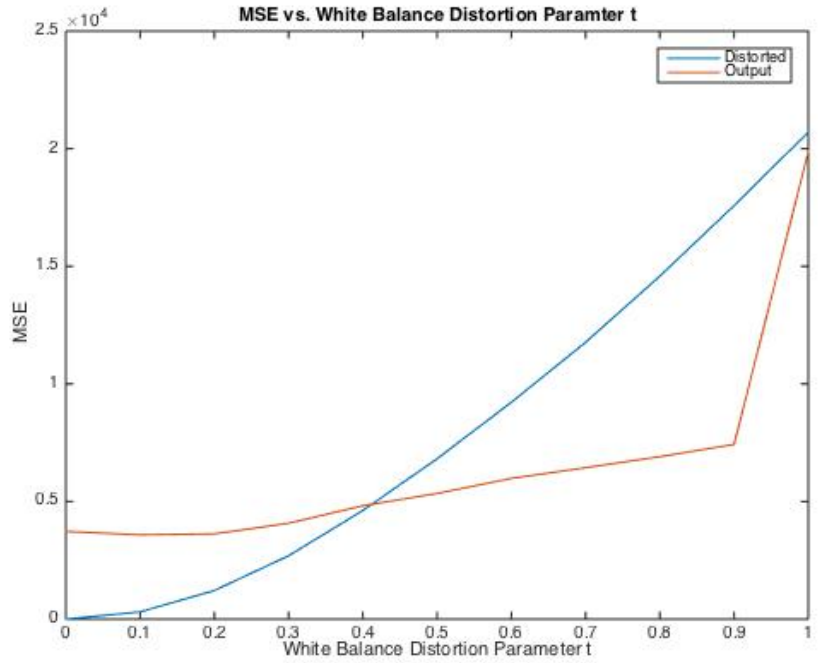
			Distorted: 19161.416 Output: 1241.766

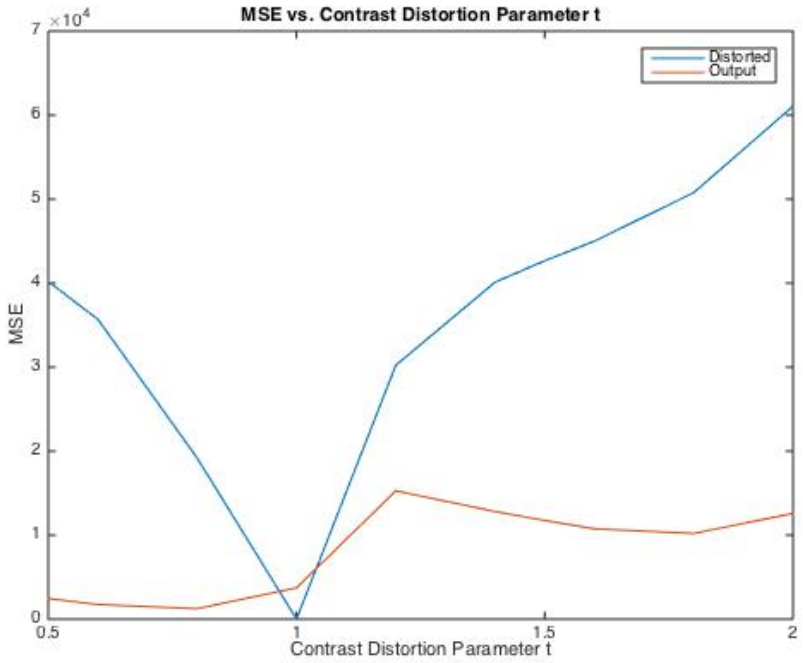
Demo 3 (Cont.)

Exposure Distortion with t = 2 (t range from .5 (dark) to 2 (bright)

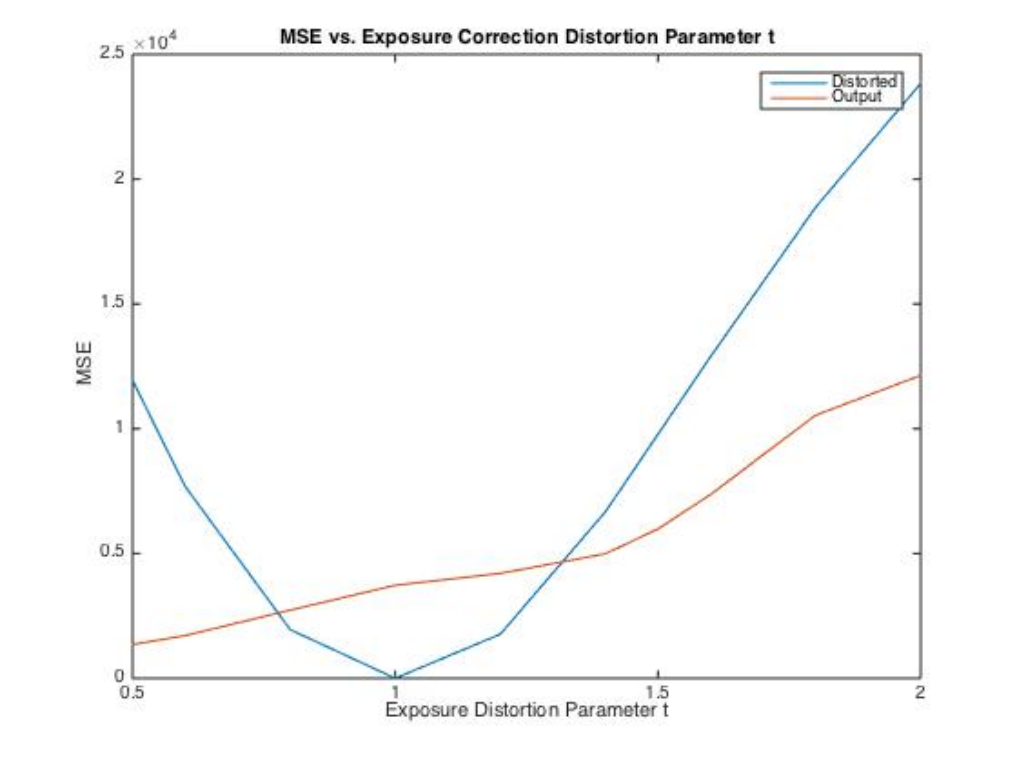
Input	**Distorted**	**Output**	**MSE**
			Distorted: 23837.887 Output: 12137.446

MSE vs. Distortion Parameters





MSE vs. Distortion Parameters (2)



Issues

- 1. Image search and matching method is not as accurate as we would like
  - Best solutions: multiple results sharing the same # of key-point pairs, resulting in a top 10
  - A possible solution is combining with histogram matching
- 2. In contrast & exposure correction, there is log(0) handling.
  - Added a small non-zero factor to black pixels
- 3. In enhancement, there is overflow in the output
  - Fixed with data type conversion and setting [0,255] boundary
- 4. Debate on matched image or original image
  - Applying enhancement on the color transferred matched image and the original image produce different results

Applications

- This type of algorithm is very applicable to many modern photo collection websites
- Companies like Facebook, Flickr, and Instagram maintain large image databases
  - They could offer an image enhance feature, which could use this algorithm to take advantage of the large database sets
  - Google Earth is another application that could make use of database matching when generating consistent satellite images of a region

Future Work

- Future modifications to the algorithm include extending the search algorithm to incorporate histogram matching
  - Could lead to better matching based on colors present in the image
- Extend the database to more categories and measuring performance
- Add noise suppression techniques and other enhancement procedures