Color image dequantization by Constrained Diffusion

Daniel Keysers and Christoph H. Lampert and Thomas M. Breuel
German Research Center for Artificial Intelligence (DFKI) GmbH
and Technical University of Kaiserslautern, Germany

ABSTRACT

We propose a simple and effective method for the dequantization of color images, effectively interpolating the colors from quantized levels to a continuous range of brightness values. The method is designed to be applied to images that either have undergone a manipulation like image brightness adjustment, or are going to be processed in such a way. Such operations often cause noticeable color bands in the images that can be reduced using the proposed Constrained Diffusion technique. We demonstrate the advantages of our method using synthetic and real life images as examples. We also present quantitative results using 8 bit data that has been obtained from original 12 bit sensor data and obtain substantial gains in PSNR using the proposed method.

Keywords: artifact reduction, quantization

1. INTRODUCTION

Many color images are quantized near the limit of just noticeable color differences; that is, colors are chosen such that objectionable color banding is barely perceptible. The 8 bit per channel quantization commonly used for color images is an example of this, as are the palette based color image representations.

When such images are manipulated or processed, objectionable banding often appears. For example, when we adjust the brightness of an image to lighten darker image regions or darken light image regions with an "adjust curves" or "gamma correction" operation, those regions often show banding, as the differences in color between two neighboring regions of constant (quantized) pixel values are increased.

One way of avoiding this problem is to capture and process images using more than 8 bit per channel, but that is often impractical because most current cameras only output 8 bit per channel images, even if their sensor captures more detail. We propose a simple and effective method for the dequantization of color images, effectively interpolating the colors from a quantized form to a continuous range of brightness values.

The method is a kind of smoothing operation, and changing the representation from 8 bits per channel to, say, 16 bits per channel followed by smoothing would dequantize an image, but it leads to image blurring. Our approach is to constrain the smoothing operation by limiting the difference to the original color (after the brightness manipulation) to a maximum value. We use smoothing by diffusion and restrict the maximum distance to the original image in each step of the diffusion process. For color images, in our current implementation, each channel is processed separately, but the method could also be used in combination with a color distance measure better corresponding to human perception as for example the distance in the CIE L*a*b color space. Furthermore, if the original operation is known, we can use that information to adapt the constraint, for example if a gamma correction with a gamma value greater than one has been used, we would allow less change for the brighter colors than for the darker ones, for which color differences have increased more during the operation.

1.1. Related Work

There is a vast amount of literature on quantization,[1] and as many different algorithms to counteract it by smoothing. While some of these methods might be suitable for color dequantization,[2] we know of no other simple methods for color dequantization in the setting described; our method can be implemented in only a few lines of code and has only a single parameter, the maximum difference between recovered values and original values. If appropriate, the algorithm can be adapted to include additional constraints, e.g. by using anisotropic diffusion.[3] From the underlying idea, our algorithm follows a similar approach as the adaptive video deblocking algorithm that is part of the MPEG-4 AVC standard[4] (aka H.264). In that algorithm, the smoothing filter adapts its strength to the quantization parameter as well, but is of much higher complexity than our algorithm.

email: {keysers,chl,tmb}@iupr.net; postal address: Erwin-Schroedinger-Str., 67663 Kaiserslautern, Germany

2. CONSTRAINED DIFFUSION

2.1. The Algorithm

The Constrained Diffusion algorithms works by iterating two simple steps, a linear diffusion step, where the image is smoothed, and a nonlinear constraint step, where the pixel values of the resulting image are clipped to not differ more than a desired threshold from the original. In detail, these steps are:

Diffusion: Set $I'_j = I_j * k$ where $k(x, y)$ is the linear diffusion kernel given by the matrix $\begin{pmatrix} 0 & \frac{1}{5} & 0 \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ 0 & \frac{1}{5} & 0 \end{pmatrix}$.

Constraint: Obtain the $(j+1)$th image I_{j+1} by evaluating pixel wise

$$I_{j+1}(x, y) := \max\left(I_0(x, y) - \delta,\ \min\left(I_0(x, y) + \delta,\ I'_j(x, y)\right)\right) \tag{1}$$

where I_0 is the original input image. The number of iterations n can be chosen arbitrarily. It is simple to check that for $n \to \infty$, the algorithm converges to a stable solution, but in some cases it can be beneficial to only run it for very few steps, even $n \leq 10$. The only other parameter is δ, which can either be treated as a free parameter, or derived from image properties.

Our method of Constrained Diffusion can be used in two different ways: as a preprocessing step before performing typical image processing operations like contrast enhancement, or as a postprocessing step after such operations have been performed.

2.2. Preprocessing

Assume one has an 8 bit per channel image, e.g. taken with a digital camera, and the image is not of optimal quality, e.g. too dark, or having a wrong gamma value. The usual procedure then is to apply image filters to improve the visual impression, working directly on the 8 bit data.[5]

Instead, one can first convert the image into 16 bit color depth space and perform the processing there. To do so, the distribution of the 8 bit data, which is very sparse in the 16 bit, has to be smoothed. One method for this is a Gaussian filter, but we will show that the Constrained Diffusion filter is a better choice. On the resulting 16 bit image, we run all necessary image filters to enhance the image quality and only then convert back to 8 bit. In preprocessing, we can fix $\delta = \frac{1}{256} \approx 0.0039$, because that is the quantization interval for 8 bit intensity values represented in 16 bit. Typically, it is safe to set n to a large value in that case.

2.3. Postprocessing

Often it is the case that one has only access to the resulting output of image enhancement operations, but not to the original. In this situation, the Constrained Diffusion can be applied as well, now as a postprocesing filter working directly on the 8 bit per channel data. In this case, the quantization effects that were caused by the earlier filters are reduced. The choice of δ is not obvious in this situation, it depends on the operations performed and on the image characteristics. One method to estimate a good choice is to use the average distance between intensities in the histogram of the image.

3. APPLICATIONS

In this section, we show how our method increases the quality of typical image processing tasks when acting as a pre- or postprocessing filter for 8 bit images.

To demonstrate the working principle more clearly, we use a synthetic 8 bit gray scale image (Figure 1) as input. It contains a smooth intensity gradient and a strongly structured area including some text.

For color images, the algorithms can be applied by just handling each color channel independently. We will present examples in Section 4. For image of different color depth, analogous arguments still hold, but to keep the description simple, we will stay with the most common case of 8 bit and 16 bit in our description.

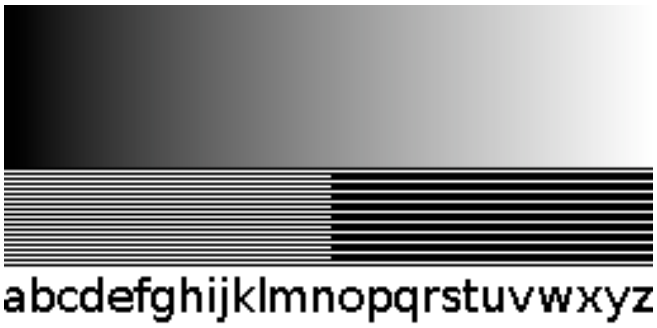


Figure 1: Original gray scale and texture image.

3.1. Quantization due to Linear Image Operations

Many important operations in digital image processing are linear, e.g. brightness adjustment and contrast enhancement. All state of the art programs for photo retouching like *Adobe Photoshop* or *The Gimp* have these integrated, often even in an automatic procedure without any user interaction.

One typical aim is to spread the range of color intensities over the whole span of possible values, yielding an optimal viewing impression. Because such operations are pixel wise functions, the total number of different values in the image is not increased, only the values themselves change.

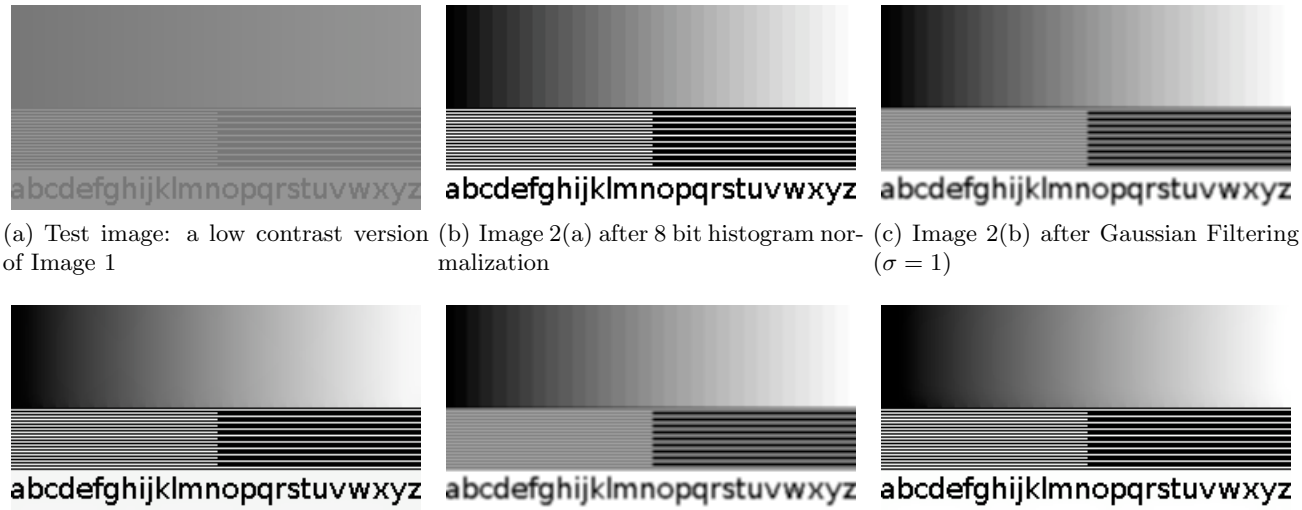
When the initial image was of low quality, usually too dark or of too low contrast, the actual number of intensities will be much lower than the possible 256, and after contrast enhancements, the small difference between two occurring intensity values could be large, even 10 units or more. The result is "banding", several clearly visible bands of one color per band without smooth transitions between them. In natural images, these are most visible in areas that are supposed to have a smooth color gradient, like blue sky. In our test image, the smooth gray scale gradient represents this image property.

Figure 2 shows the effect of a linear image processing operation using *Histogram Normalization* as an example. We start with a low contrast version (Figure 2(a)) of the test image. Performing Histogram Normalization transforms each pixel using the linear relationship $z \mapsto \frac{z-z_{min}}{z_{max}-z_{min}}$, with z being the pixel's intensity expressed in the range $[0, 1]$ and z_{min} and z_{max} are minimum resp. maximum over all intensities in the image. Histogram Normalization restores the full contrast range from black to white, but on the downside, it often causes quantizations effect where a smooth transition would be expected, see Figure 2(b).

To remove these artifacts, the image typically has to be smoothed afterwards. Image 2(c) shows the result of using a Gaussian filter for this. It restores smooth transitions, but also blurs the rest of the image. Instead, we can perform Constrained Diffusion as postprocessing after the Histogram Normalization. The result is presented in Image 2(d): smoothing occurs only up to a certain level, keeping the structure almost intact. Only the background ends up somewhat darker than it should be.

Better results are usually achieved by using preprocessing instead of postprocessing. Image 2(e) shows the result of one possibility for this. The 8 bit Image 2(a) is transformed to 16 bit color depth and smoothed with a Gaussian filter. Afterwards, Histogram Normalization is performed and the image is transformed back to 8 bit. However, at least in our situation, we observe no apparent advantage over the Gaussian as a postprocessing filter.

In contrast, Image 2(f) shows the result obtained when the Constrained Diffusion is used as preprocessing in 16 bit space, and then Histogram Normalization and the transformation back to 8 bit is performed. The result is much better, yielding a smooth gradient without blurring other parts. Visually, it is almost indistinguishable from the (usually unknown) original.



(a) Test image: a low contrast version of Image 1

(b) Image 2(a) after 8 bit histogram normalization

(c) Image 2(b) after Gaussian Filtering ($\sigma = 1$)

(d) Image 2(b) after Constrained Diffusion ($\delta = 0.025$, $n = 100$)

(e) Image 2(a) after Gaussian Filtering ($\sigma = 1$) as 16 bit preprocessing and Histogram Normalization

(f) Image 2(a) after Constrained Diffusion ($\delta = 0.0039, n = 100$) as 16 bit preprocessing and Histogram Normalization

Figure 2: Correction of Quantization due to Histogram Normalization

3.2. Quantization due to Nonlinear Image Operations

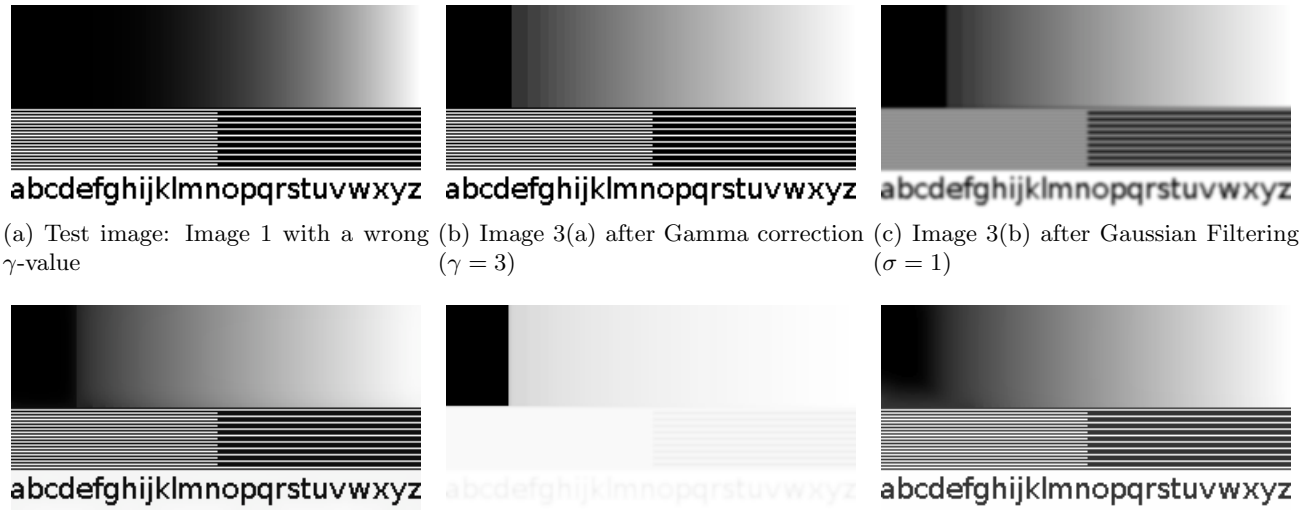
Also many nonlinear operations in image processing either cause quantization or do not improve on it. All methods of palette reduction or color clustering belong to the first kind of nonlinear operations. They usually work by combining several different colors into a single new one, thus reducing the total number of colors in the image. However, with modern image formats like JPEG and PNG, this is not as important an issue anymore as it was in the earlier days of the world wide web when the GIF format was most common. This image format can store only up to 256 different colors that must be chosen out of the roughly 16 million colors representable by 24 bit RGB.

A typical example of the second kind of operations is *Gamma Correction*. Here, the intensity of each pixel in an image is transformed nonlinearly using the format $z \mapsto z^{1/\gamma}$ where z is the pixel's intensity in the range $[0, 1]$ and γ the Gamma-value. Obviously, if in the input image z takes values from only a small subset of the possible values, the output image has the same property.

This is illustrated in Figure 3. As test image, we use the gray scale and texture Image 1, but with a wrong γ-value, overemphasizing the dark parts of the image. Applying Gamma correction in 8 bit color space restores the visual impression of brightness, but quantization occurs especially in the dark areas (Image 3(b)). Image 3(c) shows the effect of trying to compensate this by applying a Gaussian filter. The procedure smoothes the gradient, but the texture becomes blurred. Performing Constrained Diffusion on Image 3(b) yields better results: except for the darkest part, the gradient is smooth afterwards. Since it is possible to choose δ rather small, the text and pattern region does not suffer much from blurring as can be seen in Image 3(d).

Again, one should expect even better results from prefiltering. However, for the Gaussian this totally fails, see Image 3(e). Our explanation is the following: transforming Image 3(a) to 16 bit and then applying a Gaussian Filter creates many intensities in the medium brightness range. If Gamma Correction is performed afterwards, this effect is nonlinearly emphasized, causing large parts of the image to become too bright.

In contrast to this, for the diffusion method, preprocessing works very well. We first transform Image 3(a) to 16 bit color depth and then apply Constrained Diffusion. The result is Gamma corrected in the 16 bit color space. Image 3(f) shows the result after transforming back to 8 bit: the gradient is almost perfectly restored, and the textured part stays sharp.



(a) Test image: Image 1 with a wrong γ-value

(b) Image 3(a) after Gamma correction ($\gamma = 3$)

(c) Image 3(b) after Gaussian Filtering ($\sigma = 1$)

(d) Image 3(b) after Constrained Diffusion ($\delta = 0.01$, $n = 10$)

(e) Image 3(b) when using Gaussian Filtering ($\sigma = 1$) as preprocessing in 16 bit and then Gamma Correction

(f) Image 3(b) after Constrained Diffusion ($\delta = 0.0039$, $n = 100$) as 16 bit preprocessing and subsequent Gamma Correction

Figure 3: Correction of Quantization due to Nonlinear Image Operations

3.3. Quantization due to Image Compression

Of course, quantization occurs due to many other operations than just linear or nonlinear pixel wise transforms. One prominent example that we will examine here are blocking and ringing artifacts in JPEG and MPEG encoding. They occur when the DCT coefficients of JPEG or MPEG are quantized too coarsely, i.e. when targeting a too high compression ratio. The result is a visible block structure in the image (blocking), and small dots of wrong intensity close to edges (ringing).

Many specialized filters have been proposed to remove these effects,[6,7] but most of them are very complex, requiring much more computational power than the JPEG decoding itself. Our algorithm is much simpler, and it can serve as a simple deblocking and deringing filter when used as a postprocessing step. We demonstrate this in Figure 4.

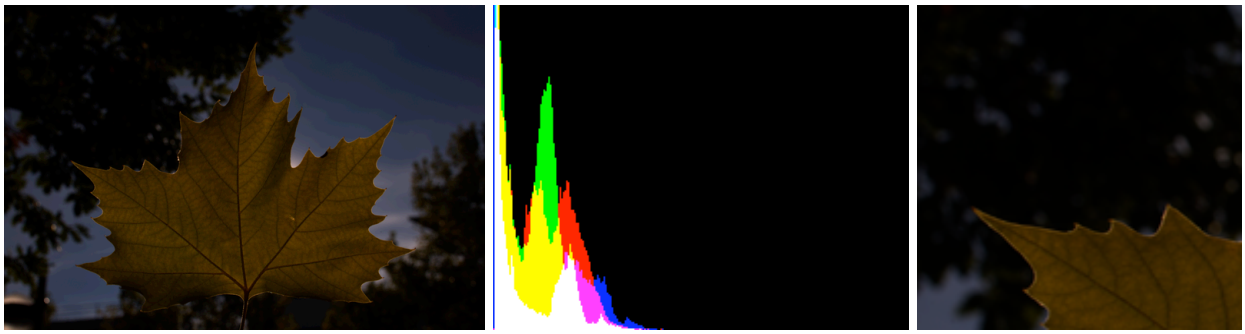
To obtain a test image, we compressed Image 1 in JPEG format with a high compression ratio and decompressed it again. Because of the way the JPEG coefficients are quantized, afterwards the gradient is not smooth anymore and not even monotonous. Also, at high contrast areas, the typical ringing or "mosquito noise" is visible. To remove the ringing, we apply a Gaussian Filter. In the result (Image 4(b)), the gradient is smoothed, but the monotony is not restored. Also as in the previous examples, the textured part is strongly blurred.

(a) Test image: a version of Image 1 that has been strongly compressed using the JPEG format

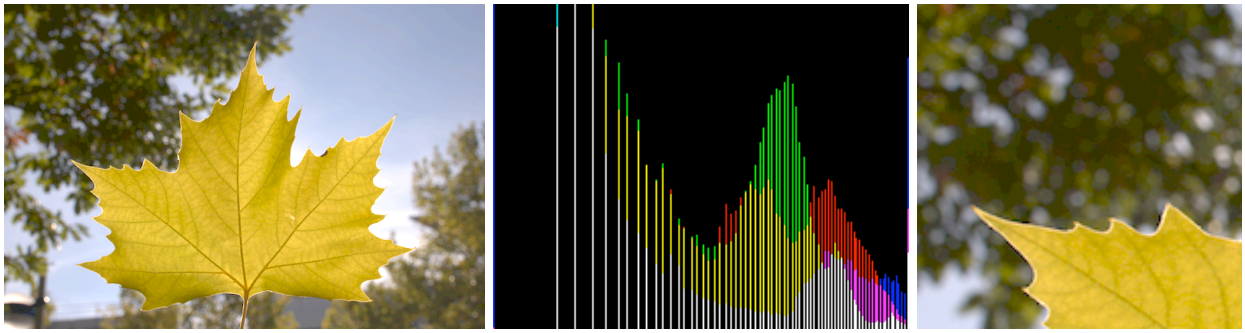
(b) Image 4(a) after Gaussian Filtering ($\sigma = 1$)

(c) Image 4(a) after Constrained Diffusion ($\delta = 0.05$, $n = 100$)

Figure 4: Correction of Quantization due to too strong JPEG Compression



(a) The original image in 8 bit representation



(b) After 8 bit contrast enhancement and gamma correction

Figure 5: Image Enhancement Example (*maple leaf* image)

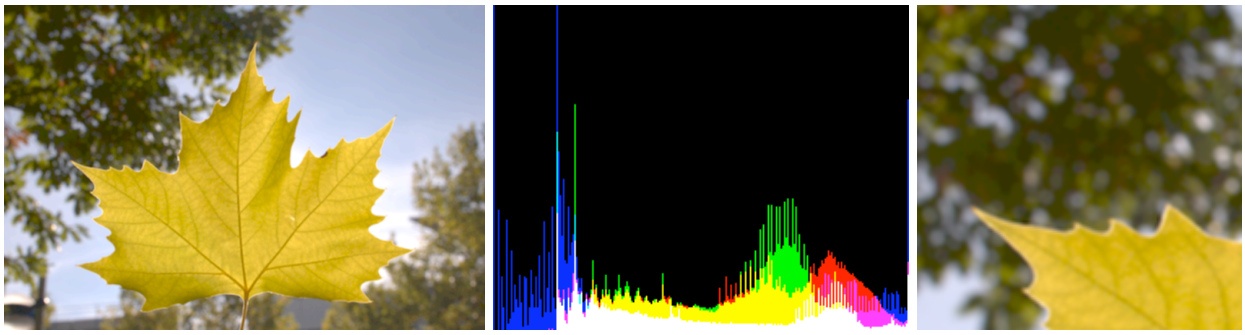
Image 4(c) shows the result of applying Constrained Diffusion instead. The gradient is smooth and monotonous again. Also, the ringing noise is strongly reduced. As only negative side effect, the background in the textured parts is significantly darkened. This is a consequence of the fact that δ has to be chosen rather large. For the JPEG example, it is not possible to apply the Constrained Diffusion as a preprocessing step.

4. EVALUATION

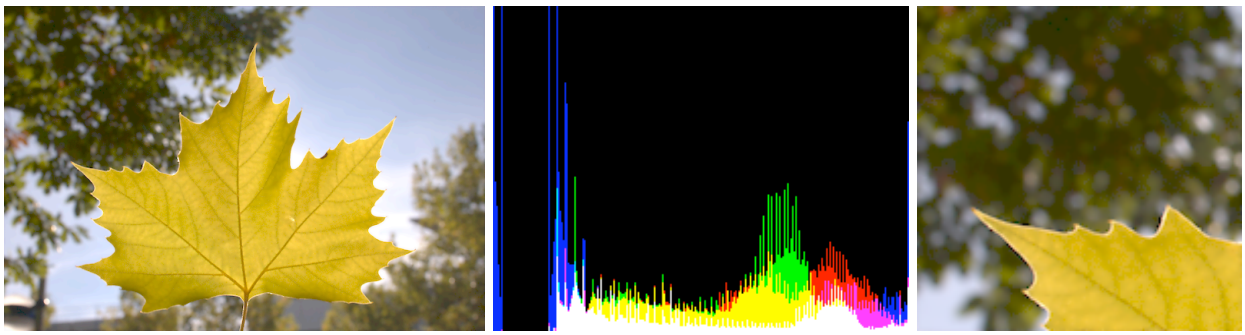
In the previous examples we used synthetic images to demonstrate the effect of Constrained Diffusion as pre- or postprocessing filter. In this section, we present some results on actual camera images that were obtained from a high quality digital camera Canon EOS 1Ds Mark II. Normally, this camera writes JPEG images with 8 bit per color channel, but internally it uses a 12 bit sensor and it also allows to extract the raw data from it. This makes it possible for us to compare the images that are obtained from restoring an 8 bit image with those that are created from the image with higher color depth where better quality restoration is possible.

Image 5(a) shows an original camera image in 8 bit representation. The image is much too dark, and of low contrast. The histogram shows that only a small part of the scale of possible gray scales is used. After 8 bit contrast enhancement and gamma correction, the overall visual impression seems okay (Image 5(b)). But, in the detail image, one sees large areas of uniform color without smooth transitions between them. The histogram shows that only few bins are filled, i.e. only few intensity values actually occur.

A Gaussian Filtering applied to Image 5(b) smoothes these dark transitions and the histogram, but it also blurs the image. This can be seen well in Image 5(c), where the specular highlights have lost their brilliance. Applying the Constrained Diffusion as postprocessing filter to Image 5(b) also smoothes the transitions, but it leaves the structure almost intact (Image 5(d)). From the histogram it can be seen that now almost all possible gray scales occur, but fewer pixels changed their value.



(c) Gaussian Filtering ($\sigma = 1$) as postprocessing



(d) Constrained Diffusion ($\delta = 0.05, n = 10$) as postprocessing

Figure 5: Image Enhancement Example (Cont.)

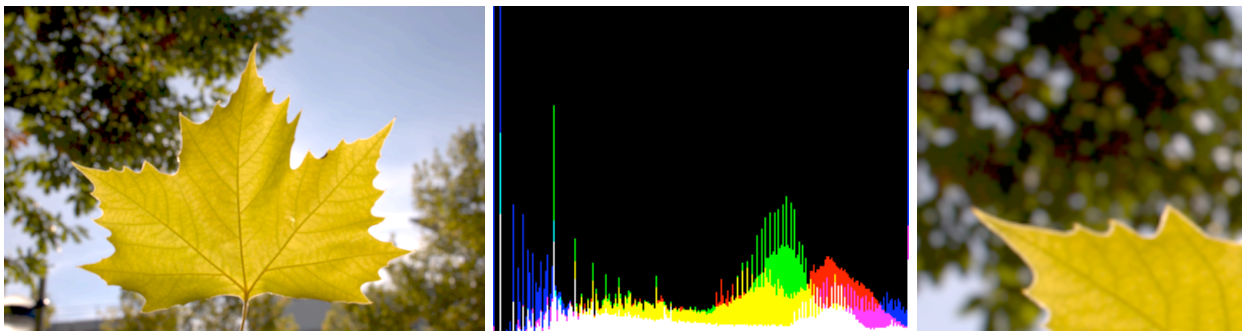
Since we have access to the original image, we can try to use preprocessing again. Using a Gaussian Filtering (Image 5(e)) does not show a significant improvement over the postprocessing setup. Using Constrained Diffusion Filtering as preprocessing on Image 5(a) shows a slightly improved result (Image 5(f)). Since δ can be chosen smaller, the contrast is better preserved in dark areas than for postprocessing, although also here some structure is lost. None of the methods can recreate the full image data, as can be seen when comparing with Image 5(g) that is the result of restoring the image from the camera's 12 bit sensor data.

We have applied our method to many other images. Some of them are presented in Figure 6. Constrained Diffusion always improves the visual quality of the image, although not all image artifacts can be avoided or removed, respectively.

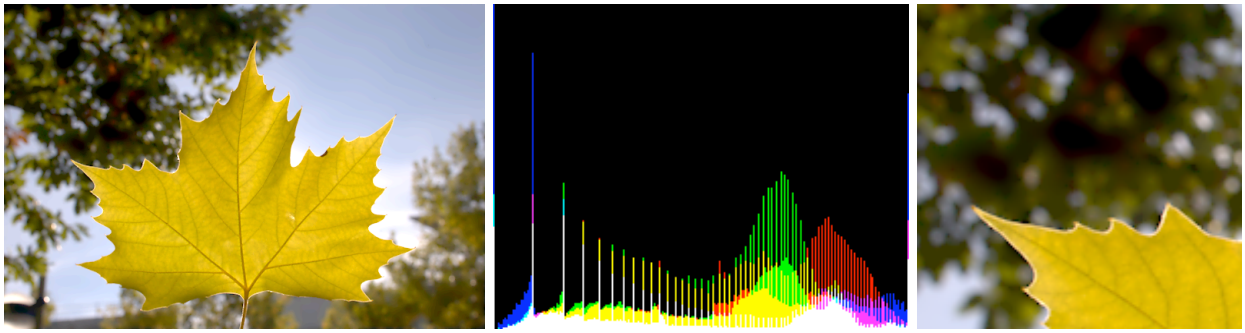
As an additional test, for the four images (as shown in Figure 5 and Figure 6), we calculated the numerical image similarity between the optimal image and the restored version, using either Gaussian Filtering or Constrained Diffusion, and testing both preprocessing or postprocessing. As a measure we used the Peak Signal to Noise Ratio (PSNR), which has a logarithmic scale with larger values meaning higher similarity between images. The PSNR is defined as a function of the Mean Squared Error (MSE), i.e. the average squared pixel value difference, as follows:

$$PSNR = -10 \log_{10} \frac{MSE}{255^2} \tag{2}$$

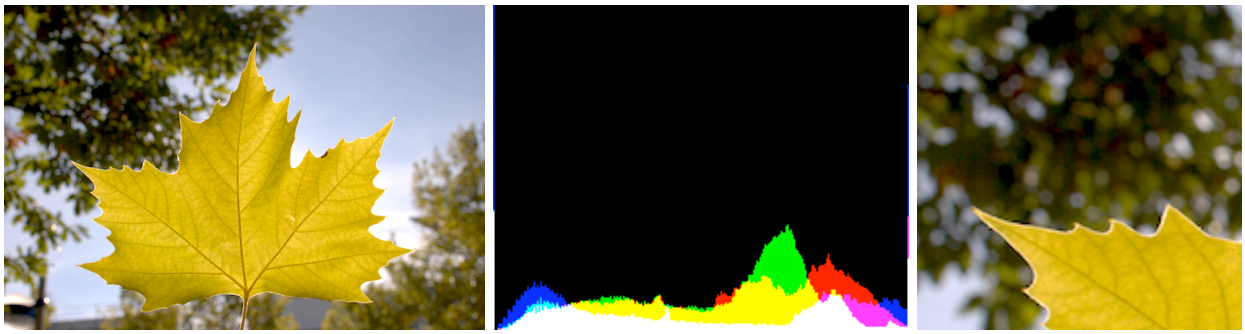
The numbers presented in Table 1 are the average over the PSNR of the red, green and blue color channel of the different versions that are restored from 8 bit. As reference, always the 12 bit restored version was used. As one can see, Constrained Diffusion often increases the PSNR, especially when used as a preprocessing operation. For Gaussian Filtering the effect can be positive or negative. The use of Constrained Diffusion as a preprocessing step improved the PSNR the most, on the average by 3.14dB.



(e) Gaussian Filtering ($\sigma = 1$) as preprocessing



(f) Constrained Diffusion ($\delta = 0.0039, n = 100$) as preprocessing



(g) Reconstruction from 12 bit camera data

Figure 5: Image Enhancement Example (Cont.)

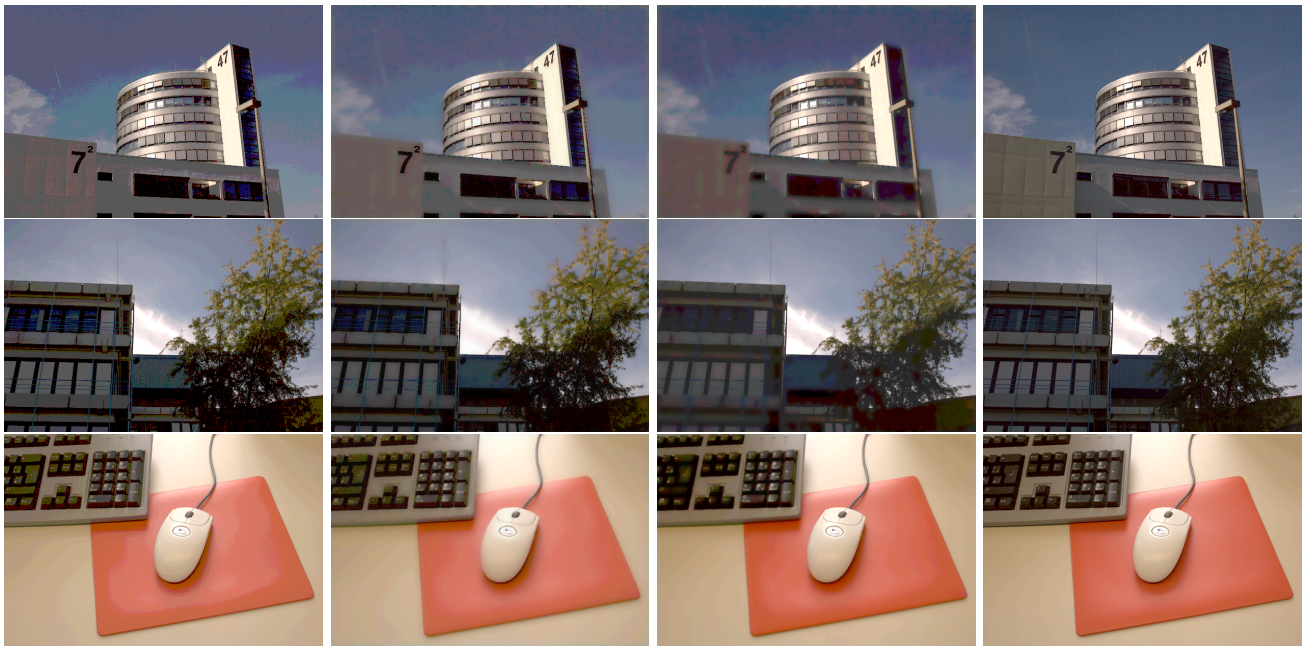


Figure 6: Example images for Constrained Diffusion Dequantization as postprocessing or preprocessing filter. For the three images *building* (top),*tower* (middle) and *mouse pad* (bottom) Histogram Normalization and Gamma Correction are performed. The images show from left to right: usual 8 bit procedure, Constrained Diffusion for pre- and for postprocessing and the optimal image, restored from 12 bit data.

Table 1: PSNR improvement for Gaussian and Diffusion preprocessing and postprocessing methods. Histogram Normalization and Gamma Correction is always performed, either in 8 bit or 16 bit color depth. Bold differences mean a substantial gain (> 0.5dB), italics values indicate a substantial loss (< -0.5dB). The column "none" refers to only performing Histogram Normalization and Gamma Correction in 8 bit color depth. The reference image is the one obtained from the original, 12 bit color depth image.

	none	postprocessing		preprocessing	
		Gaussian	Diffusion	Gaussian	Diffusion
maple leaf	27.25 dB	26.67 (*−0.58*) dB	27.04 (−0.21) dB	32.80 (+**5.55**) dB	33.18 (+**5.93**) dB
building	26.82 dB	26.85 (+0.03) dB	29.78 (+**2.96**) dB	26.90 (+0.08) dB	28.95 (+**2.13**) dB
tower	27.29 dB	26.95 (−0.36) dB	28.97 (+**1.68**) dB	26.58 (*−0.71*) dB	28.99 (+**1.70**) dB
mouse pad	30.86 dB	31.55 (+**0.69**) dB	31.60 (+**0.74**) dB	32.07 (+**1.21**) dB	33.67 (+**2.81**) dB
average	28.06 dB	28.01 (−0.05) dB	29.35 (+**1.29**) dB	29.59 (+**1.53**) dB	31.20 (+**3.14**) dB

5. CONCLUSION

We presented a method for the dequantization of color images, that is both simple and fast. A program that performs the algorithm can be implemented in a few lines of Matlab code, or a few dozen lines of C/C++. The method works by interpolating the colors from their quantized form to a continuous spectrum, using a linear diffusion process, but constraining the maximal allowed deviation from the original by a simple threshold.

As one application we showed postprocessing of images that have been manipulated using typical image processing operations like Histogram Normalization, Gamma Correction or JPEG compression. Constrained Diffusion can to some degree compensate the artifacts that those operations introduce, often clearly improving the visual impression of the image.

Even more promising is the use of our method as a preprocessing filter to avoid that such artifacts occur at all. To do so, an image that is typically present in a representation with 8 bits per color channel is extended to 16 bit and the quantization that this causes is removed by the Constrained Diffusion filter. The resulting image can be processed with arbitrary image processing filters, obtaining very good results.

We showed several examples of synthetic and real life images, comparing our method to the usual way of smoothing by a Gaussian Convolution filter. We also presented measurements that show that our method improves not only the subjective visual impression, but also the PSNR as an objective quality measure, and does so better than Gaussian Filtering. Using the Constrained Diffusion technique as preprocessing, the PSNR could be improved by 3.14dB averaged over four images for the processing with Histogram Normalization and Gamma Correction.

Many improvements of the algorithm come to mind. For one thing, the algorithms contains a free parameter δ, that is chosen as a constant for all parts of the image. Adaptivity in this parameter could clearly improve the results e.g. by choosing larger values in dark areas than in bright ones, when the target operation is Gamma Correction. Also, our algorithm relies on isotropic diffusion, thus assuming that a pixel's intensity should be influenced equally by its neighboring pixels. Often, this assumption is wrong, e.g. close to edges in the image. So, a method that is based on an anisotropic diffusion filter could achieve even better results than the one we presented.

ACKNOWLEDGMENTS

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project IPeT (01 IW D03).

REFERENCES

1. R. M. Gray and D. L. Neuhoff, "Quantization.," *IEEE Transactions on Information Theory* **44**(6), pp. 2325–2383, 1998.
2. A. Rizzi, C. Gatta, and D. Marini, "A new algorithm for unsupervised global and local color correction," *Pattern Recognition Letters* **24**, pp. 1663–1677, 2003.
3. J. Weickert, *Anisotropic Diffusion in Image Processing*, Teubner Verlag, Stuttgart, 1998.
4. P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Techn.* **13**(7), pp. 614–619, 2003.
5. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
6. M.-Y. Shen and C.-C. J. Kuo, "Review of postprocessing techniques for compression artifact removal," *Journal of Visual Communication and Image Representation* **9**(1), pp. 2–14, 1998.
7. M. Yuen and H. R. Wu, "A survey of hybrid MC/DPCM/DCT video coding distortions," *Signal Process.* **70**(3), pp. 247–278, 1998.