Regularizing Image Intensity Transformations Using the Wasserstein Metric

Magnus Oskarsson(✉)

Centre for Mathematical Sciences, Lund University, Lund, Sweden
magnuso@maths.lth.se

Abstract. In this paper we direct our attention to the problem of discretization effects in intensity transformations of images. We propose to use the Wasserstein metric (also known as the Earth mover distance) to bootstrap the transformation process. The Wasserstein metric gives a mapping between gray levels that we use to direct our image mapping. In order to spatially regularize the image mapping we apply anisotropic filtering and use this to steer our mapping. We describe a general framework for intensity transformation, and investigate the application of our method on a number of special problems, namely histogram equalization, color transfer and bit depth expansion. We have tested our algorithms on real images, and we show that we get state-of-the-art results.

Keywords: Image enhancement · Discretization · Wasserstein metric

1 Introduction

Image intensity transformations are used in many applications. When we need to change the color space in some way we can do this by applying a transformation function from input colors to output colors. One problem that has to be handled is the discrete nature of the signal. As an example consider the classic problem of histogram equalization. This process is used to increase the contrast, in an optimal way. This is achieved by applying a gray level transformation such that when it is applied to the input image, the output image will get a completely flat gray level distribution. This can be done exactly if we work with continuous gray levels, but for a real image we will get discretization effects, which is illustrated in Figure 1. To the left a dark input image is shown, with its corresponding gray level distribution underneath. If we apply standard histogram equalization on the image, we get the middle image, with its corresponding gray level distribution below. One can see that the overall distribution is more spread out, but it is by no means flat. This is due to the discrete intensity gray values of the input image. We will in this paper show how we can enforce a special output gray value distribution using the so-called Wasserstein metric (or Vasershtein). To the right in Figure 1 one can see the result of applying our proposed method in the case of histogram equalization. Notice that we get less discretization effects in the image and that the corresponding output gray value distribution is completely flat.

R.R. Paulsen and K.S. Pedersen (Eds.): SCIA 2015, LNCS 9127, pp. 275–286, 2015.
DOI: 10.1007/978-3-319-19665-7_23

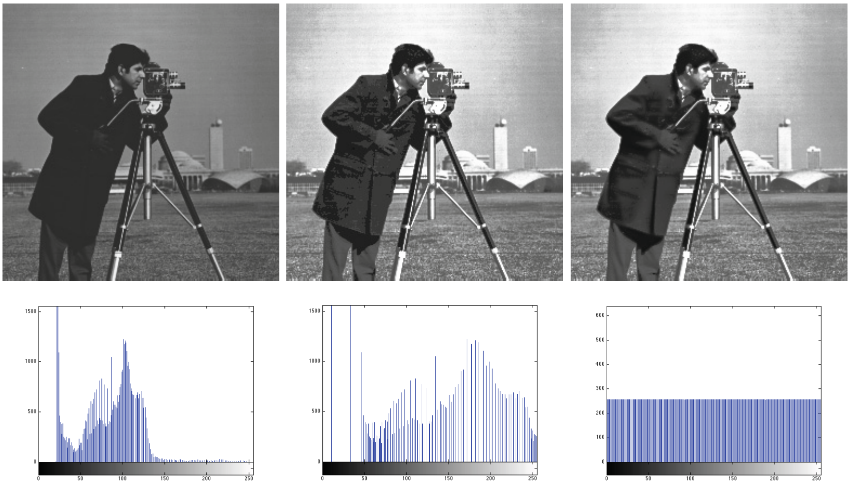

Fig. 1. Top row shows images, and bottom row shows the corresponding intensity distributions. To the left is a dark input image. The middle shows the result of standard histogram equalization. Due to the quantization of the input image, the distribution of the equalized won't be flat, and there will be gray levels that are not used. To the right is the result of applying our Wasserstein regularization using algorithm 1. The resulting distribution is flat, and we make use of all gray-level intensities. One can see that we get less discretization effects in the image, see e.g. the coat.

Histogram equalization is one example where one would like to enforce a special type of intensity value distribution, but there are numerous other applications, within e.g. HDR imaging [3] and within Chromatic adaptation and Color constancy [1,10]. We will in Section 5 give qualitative results on using our method for histogram equalization and for color transfer between images. For other methods on color transfer see e.g. [9].We will also in more detail study how we can use our proposed method for bit depth increase. For most of today's digital devices for capture and display of image data the bit depth varies, from e.g. 6 to 10 bits per color channel, due to technical limitations. When transferring image data between different modalities the need for changes in bit depth is apparent. Decreasing the bit depth is in many cases quite straightforward, but increasing it involves some form of interpolation. This problem is in general not well posed, and we need to regularize it in some way. One often assumes conditions on spatially nearby pixels, that they vary smoothly or piece-wise smoothly. There has been a number of previous work done in the area of bit depth increase. Let's assume a p-bit input image, I_{in}, with gray values in the range $[0, 2^p - 1]$ and a q-bit output image, I_{out}, with range $[0, 2^q - 1]$. We further assume that $p < q$. A basic idea is to just zero pad the input image, or in some other way spread out the dynamic range linearly so that $I_{out}(x, y) = I_{in}(x, y) \cdot 2^{q-p}$. Variations of these methods include multiplication with an ideal gain and then round to nearest integer in the output range, or making a non-linear transformation, e.g.

using gamma expansion. See [12] for details on these methods. Such methods all construct a one to one mapping that is based solely on the gray level at a given pixel, that for most problems leads to contouring artifacts along smooth gradients. To overcome such shortcomings some form of spatial filtering can be applied. We will in the experimental section compare our method to those of [14] (based on flooding) and [7] (based on minimum risk).

2 Problem Formulation and Motivation

We will in this section describe the general problem that we address in this paper. Assume that we have an input (gray value) image $I_{in}(x, y)$ with $(x, y) \in D \subset \mathbb{R}^2$, whose intensity values we would like to transform in some way. One way is to apply a function $f : \mathbb{R} \to \mathbb{R}$ so that $I_{out}(x, y) = f(I_{in}(x, y))$. As described in the introduction, this often introduces discretization effects due to the discrete values of I_{in}. In order to remove these effects one might apply some form of spatial regularization so that f is not only a function of the intensity value but also of the position in the image, i.e. $f : D \times \mathbb{R} \to \mathbb{R}$ and $I_{out}(x, y) = f(x, y, I_{in}(x, y))$. Often this function f is not explicitly given, instead the spatial regularization is done using some form of filtering. We will now look at the specific case where we have a priori information about the gray value distribution of the output image. We will look at the following problem.

Problem 1. Given an input image I_{in} and an output intensity distribution $h_{out}(t)$ find an output image I_{out} with gray value distribution equal to $h_{out}(t)$ and such that $||I_{in} - I_{out}||$ is small.

This is a quite general formulation and we will see in Section 5 that many different problems can be cast in this way. We will in this paper decribe a constructive non-iterative method that solves problem 1. We will use the Wasserstein metric to find a function that prescribes how we transfer the gray values in an optimal way in order to get a specific output intensity distribution. We will then use non-linear anisotropic smoothing of the input image to find the closest output image.

The paper is organized as follows. In Section 3 we present the Wasserstein metric and its application to our problem. In Section 4 we describe the basic outline of our method. The method applied to histogram equalization, color transfer and bit depth expansion is described in Section 5. Using our general method, we show results comparable to state-of-the-art methods.

Our method could be incorporated as a step in many existing image processing applications where one wants to avoid discretization effects. Our aim in this paper is not to present a novel method – for say bit depth expansion – but rather show that using a novel combination of standard components – not specifically tuned to a special application – we can achieve results that are very much comparable to state-of-the-art methods. One of the major strengths that we see with our proposed method is the hard regularization effect we get from using the Wasserstein metric. In many cases one could get good results by applying spatial regularization using anisotropic smoothing, but without the Wasserstein regularization, problems with parameter choices and over-smoothing become apparent.

3 The Wasserstein Metric

We will in this section describe how we use the Wasserstein metric. The general metric was introduced in [13] and the name was coined in [4]. In computer science it is known as the Earth mover's distance (EMD). Informally the metric measures the minimum work of transforming one distribution into another.

We will start by defining the metric. It can be defined on an arbitrary metric space for which every probability measure is a Radon measure, but we will use it in a more simple form. We use the metric space $\mathbb{R}$ with the normal Euclidean distance. Let $P(\mathbb{R})$ denote the collection of all probability measures μ on $\mathbb{R}$ with finite second moment: for some s_0 in $\mathbb{R}$,

$$\int_{\mathbb{R}} (s - s_0)^2 \, \mathrm{d}\mu(s) < +\infty. \tag{1}$$

Then the second Wasserstein distance between two probability measures μ and ν in $P(\mathbb{R})$ is defined as

$$W(\mu,\nu) := \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}\times\mathbb{R}} (s-t)^2 \, \mathrm{d}\gamma(s,t) \right)^{1/2}, \tag{2}$$

where $\Gamma(\mu,\nu)$ denotes the collection of all measures on $\mathbb{R}\times\mathbb{R}$ with marginals μ and ν on the first and second factors respectively. The function $\gamma(s,t)$ describes how this transformation is done. The minimizing γ is called *optimal transport*.

Here we will use the Wasserstein metric between the gray value distributions of two images. We will represent these distributions as discrete histograms. There are efficient algorithms for calculating the Wasserstein distance between two discrete distributions. We have used the fast method described in [8]. By estimating the distance we also get the optimal transport γ, which in this case is a matrix. The number $\gamma(s,t)$ tells how many pixels of gray value s in the input image that should be transformed to gray value t in the output image. As an illustration consider again the example in Figure 1. We have calculated the Wasserstein metric between the middle and the right histogram in Figure 1. Three rows of the corresponding optimal transport γ are shown in Figure 2. The estimated γ gives a number of constraints on the transformation that we apply on the input image. These constraints are in general not enough to give a unique output image, i.e. there are many output images that will yield the same output histogram. So we will need to add some more constraints to regularize our solution. This could be done in a number of ways, e.g. minimizing the difference between the input and output image or minimizing the gradients of the output image to get a smooth solution. This could of course also be application dependent, but one important point is that we can in many cases easily use the Wasserstein metric to ensure that the output image follows the desired gray value distribution. We will in the next section describe our proposed method based on anisotropic filtering.

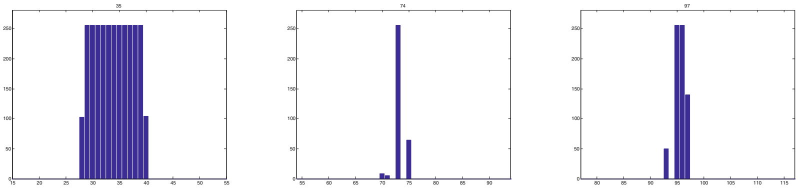

Fig. 2. Three rows of the transformation matrix γ between the middle histogram in Figure 1 and the histogram to the right in Figure 1. The bars depict from left to right $\gamma(35,:)$, $\gamma(74,:)$ and $\gamma(97,:)$. The sum of the bars equals the number of pixels with gray levels 35,74 and 97 in the input image in Figure 1.

4 Enforcing the Wasserstein Metric

We are now ready to describe how we can enforce the Wasserstein metric exactly. We will do this using a filtering approach, where we by smoothing the image steer our transformation.

We assume that we have calculated the optimal transport $\gamma(s,t)$ between the input and output gray levels. We then take the input image and smooth it anisotropically. You can use your favorite smoothing method; we will in Section 5 show results using three different smoothing methods, and the results are quite similar. The important thing is that we smooth enough to eliminate the discretization effects but so that not too much of the true edges in the image are lost. This means that some form of anisotropic filtering should be used. This will give us a smoothed version I_{sm} of the input image I_{in}. We can then for each pixel calculate the signed distance between the input image and the smooth image:

$$\varDelta I(x,y) = I_{sm}(x,y) - I_{in}(x,y), \quad (x,y) \in D. \tag{3}$$

This gives a weighting for each pixel if it wants to decrease or increase its gray level. By sorting these distances for a certain input gray value, we get a priority list on the pixels of that color. We then use γ to transform the gray values in the order of this priority list. The method is summarized in algorithm 1.

Regarding color, we propose to use our method on each color channel separately. Since we specify the output distribution, the risk of color artifacts is small.

One crucial part of algorithm 1 is the construction of the smoothed version of the input image. This should be done by some form of edge preserving noise filtering. In our experiments we have tried three well performing noise reduction methods, namely Bilateral filtering [11], BM3D [2] and anisotropic filtering based on the structure tensor [6]. The results vary slightly, with BM3D giving the best PSNR values. However BM3D gives also in some cases some unwanted artifacts. More details can be found in the experimental evaluation.

Algorithm 1. Wasserstein regularized intensity transformation

1: Given I_{in} and a desired output intensity distribution $h_{out}(t)$.
2: Calculate the histogram $h_{in}(s)$ of I_{in}.
3: Estimate $\gamma(s,t)$ from the Wasserstein metric between $h_{in}(s)$ and $h_{out}(t)$
4: Estimate a smoothed version I_{sm} by anisotropically filtering I_{in} spatially.
5: Calculate the distance function $\Delta I(x,y) = I_{sm}(x,y) - I_{in}(x,y) \quad (x,y) \in D$.
6: **for** each gray level s **do**
7: Sort the pixels with $I_{in} = s$ according to ΔI to get a priority list.
8: Redistribute these pixels according to $\gamma(s,:)$ in the order of the priority list into the output image I_{out}.
9: **end for**

5 Applications

We will in this section describe how the method presented in the previous sections can be used in a number of applications. We give examples on histogram equalization, color transfer and bit-depth increase.

5.1 Histogram Equalization

We have already in the introduction seen an example of histogram equalization. We simply start by performing standard histogram equalization on the input image. We then run algorithm 1 with this image as input, and a flat target histogram h_{out}. Since histogram equalization normally is used to increase the intensity constrast, when running it on a color image one usually first transforms the image to a new colorspace (e.g. HSV) that separates color and intensity information. One then runs the algorithm on the intensity channel alone and transfers back to the original colorspace. The details are summarized in algorithm 2. In Figure 3 we show the result of running algorithm 2 on an over

Algorithm 2. Wasserstein regularized histogram equalization

1: Given an input image I_{in} convert the image to an HSV representation I_{HSV}.
2: Perform traditional histogram equalization on the V-channel I_V to get I_{Vhq}.
3: Estimate I_{Vhq2} using algorithm 1 with input I_{Vhq} and a flat h_{out}.
4: Convert I_{HSV2}, with V-channel equal I_{Vhq2}, to RGB to get I_{out}.

saturated image. To the left is the input image, in the middle is traditional histogram equalization and to the right is the result running our method. One can see that the discretization effects are much smaller.

5.2 Color Transfer

In this section we look into how we can use our method to transfer the color distribution of one image to another. One application where this can be used is

Fig. 3. The result of running algorithm 2 on an over saturated image. To the left is the input image, in the middle is the rersult of running traditional histogram equalization and to the right the result of our method. One can see that we get less quantization effect in the right hand image. The results are best viewed on screen.

the case when we have taken multiple images of the same scene with different illunination. An example where we have used our method is shown in Figure 4. To the left and in the middle two images of the same scene are depicted. To the left the image is taken without flash and in the middle with flash. In many cases one gets a more desirable illumination without flash but one also gets a more noisy image since the illumination is poor. There has been a number of previous work on how to combine two images taken with and without flash, see e.g. [9]. Here we just simply transfer the color distribution from the image with no flash to the image with flash. We do this by first calculating the distribution of the no flash image. Since we want to transfer color information we do this for each color channel separately. We then proceed in the same manner as we did for the histogram equalization, i.e. we start by applying a discrete histogram transformation on the flash image using the calculated color distribution from the no flash image. We do this by finding the tranformation $f(s)$ that minimizes

$$|| \int_0^s h_{in}(u) du - \int_0^{f(s)} h_{out}(u) du ||_2. \tag{4}$$

We have used the Matlab function `histeq` to do this. We then use $f(I_{in}(x,y))$ together with the desired distribution h_{out} as input to algorithm 1. The result can be seen to the right in Figure 4. One can see that we get an image without the noise from the no flash image but with the same color tone.

5.3 Bit Depth Increase

We will now show how our ideas can be used in bit depth expansion. We will base our method on the assumption that the high bit image should have the same gray level distribution as the low bit output image, but on a finer scale. In order to ensure this we start by estimating the high bit distribution from the low bit distribution. This is in many case a much better posed problem

Fig. 4. An example of color transfer using our proposed method. To the left is a noisy image taken without flash. In the middle is an image of the same scene taken with flash. To the right is the middle image where the colors have been transformed to match the distribution of left hand image.

than the actual bit expansion problem. We do this by interpolating the low bit distribution to a continuous distribution, and then resampling this distribution to the desired bit depth. From the low-bit input image we calculate the histogram, $h_{in}(s), s = 0, \dots, (n-1)$, where $n = 2^p$. We assume that the input image is sampled from some ideal image with continuous probability distribution $h_0(u)$ so that

$$h_{in}(s) = \int_{s+0.5}^{s-0.5} h_0(u)du, \quad s = 0 \dots (n-1). \tag{5}$$

In order to estimate h_0 from h_{in} we need to regularize the problem somewhat. We will use a spline-based approach. To this end we assume that h_0 can be approximated by a piece-wise polynomial of degree two,

$$h_0(u) = a_s + b_s u + c_s u^2, \quad (s-0.5) \leq u \leq (s+0.5). \tag{6}$$

Equation (5) together with the assumption that h_0 and its derivative are continuous, gives a linear system of equations in a_s, b_s and c_s. This in turn gives us an estimate of $h_0(u), -0.5 \leq u \leq n-0.5$, from which we can discretize to get $h_{out}(t)$ with the desired output number of bins.

We now have all the components of our bit depth expansion. From a low bit input image we start by expanding it to the desired bit depth using ideal gain. From the gray value distribution of this image we can estimate the desired output gray value distribution using the method just described. We now are in the setting described in Section 4 and we can use algorithm 1 in order to estimate an output image that follows the desired gray value distribution. The steps in our bit expansion method are summarized in algorithm 3.

We have tested our algorithms on a number of real images. The originals were 8-bit color images, with gray values between 0 and 255. We then constructed low

Algorithm 3. Wasserstein regularized bit depth increase

1: Given an input image I_{in} with p bits.
2: Calculate the ideal gain q bit image I_g.
3: Calculate the gray value distribution $h_{in}(s)$ of I_g.
4: Estimate the output distribution $h_{out}(t)$ from $h_{in}(s)$ using a spline based interpolation.
5: Estimate I_{out} using algorithm 1 with input I_g and h_{out}.

Table 1. Peak Signal to Noise-ratio for a number of test images with different input bit-depth. See text for details.

	6 to 8 bit					4 to 8 bit				
Image	IG	MR	CA	BF	BM	IG	MR	CA	BF	BM
pepper	**47.06**	45.11	47.32	43.64	44.27	34.67	33.89	**37.31**	36.19	36.99
building	**47.14**	44.39	46.36	44.67	45.36	34.84	34.84	35.08	34.20	**35.27**
tree	**47.14**	44.49	46.14	44.59	45.05	34.89	34.94	35.57	34.89	**35.98**
flower	46.88	46.77	**49.64**	47.84	48.97	34.40	32.18	**37.37**	36.31	35.91
spider	47.10	41.19	**48.71**	46.70	47.80	34.86	34.47	**38.03**	38.08	37.43

bit versions by dividing the gray values by 2^k and rounding, to get an $(8-k)$-bit image. As a first example, the result from running algorithm 3 using bilateral filtering on the pepper image can be seen in Figure 5. The top row shows close ups of the ideal gain version of the input image. From left to right we have 5, 4 and 3 bit input images. The bottom row shows the output of our algorithm using bilateral filtering. In Figure 6 the resulting color distributions for 5 bit expansion is shown. It also shows the corresponding ground truth distributions in black. One can see that they follow each other very well. We have also conducted a test on a number of other real test images. These are taken from the database described in [5]. In this case the original images were 8-bit, that were converted to low bit images. We have also run two state-of-the-art bit expansion algorithms for comparison, namely the method of [7], based on minimum risk and the content adaptive bit-depth expansion algorithm of [14]. We have constructed 4-bit and 6-bit versions of the test images and run the algorithms to – in all cases – produce 8-bit results. The output images have then been compared to the ground truth 8-bit images. The results from this evaluation can be seen in table 1 where the peak signal to noise ratio is presented. We include results from ideal gain (IG), Mittal et al. (MR), Wan et al. (CA) and our approach using bilateral filtering (BF), BM3D (BM) and the structure tensor anisotropic filtering (ST). The corresponding results for the structure similarity index, SSIM [15] can be found in table 2. A number of observations can be made. We see that for small bit increase problems, the ideal gain actually performs very well. This is due to the fact that the error is always bounded by a small amount in this case. However when looking at the images one can see that the other algorithms output more pleasing results. This serves as a reminder that just looking at the PSNR values

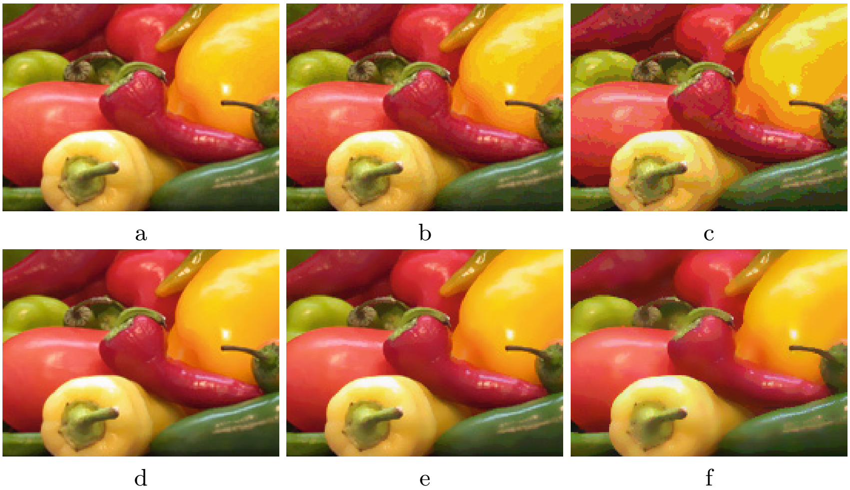

Fig. 5. Close-up results from running algorithm 3 on the pepper image, for a number of different bit depth inputs. In these examples bilateral filtering was used for the smoothing step. Top row shows, from left to right, five, four and three bit color input. Bottom row shows the resulting output images. The results are best viewed on screen.

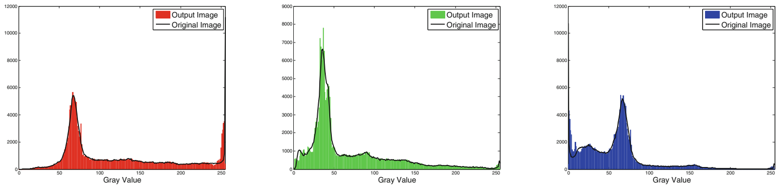

Fig. 6. The figure shows the result from running algorithm 3 using bilateral filtering. The resulting gray value distributions of the output image, for the red, green and blue channel respectively are shown. Also shown are the corresponding distributions for the ground truth 8-bit image. One can see that they follow each other well.

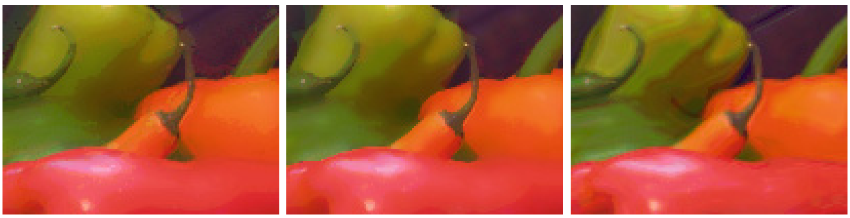

Fig. 7. Magnified results from running algorithm 3 using the three different filtering methods. From left to right it shows the result using the structure tensor adaptive filtering, using bilateral filtering and using BM3D. The results are best viewed on screen.

Table 2. SSIM for a number of test images with different input bit-depth. See text for details.

	6 to 8 bit						4 to 8 bit					
Image	IG	MR	CA	BF	BM	ST	IG	MR	CA	BF	BM	ST
pepper	0.99	0.98	0.99	0.98	0.98	0.99	0.87	0.89	**0.93**	**0.93**	0.91	**0.93**
building	0.99	0.99	0.99	0.99	0.99	0.99	0.93	0.93	**0.94**	0.93	0.92	**0.94**
tree	0.99	0.99	0.99	0.99	0.99	0.99	0.91	0.92	0.92	0.92	0.91	**0.93**
flower	0.97	0.99	0.98	0.98	0.98	0.98	0.84	**0.91**	0.90	**0.91**	0.90	0.87
spider	0.99	0.95	0.99	0.99	0.99	0.99	0.89	0.91	**0.96**	**0.96**	0.95	0.93

is not always a good way of evaluating the results. The SSIM measure gives a somewhat better way of comparing, but also suffers to some extent from the same problems. Setting this caveat aside one can see that our simple algorithm performs on par with the highly specialised state-of-the-art methods for 4-6 bit input.

In Figure 8 close ups for the results on the test images are shown for our method using bilateral filtering. From table 1 and 2 one can see that running our method using bilateral filtering or BM3D gives comparable results to each other. In Figure 7 a comparison for our method using the different filtering types is shown. One can see that we get some undesirable artifacts near edges using BM3D.

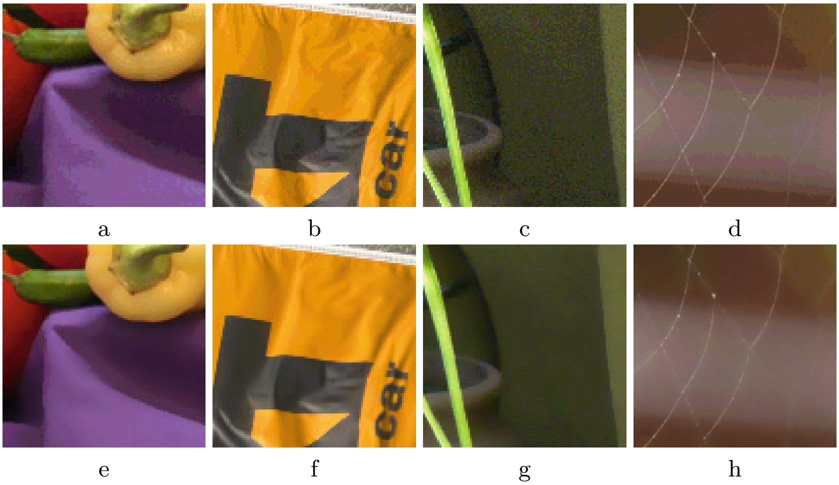

Fig. 8. Results from algorithm 3 for the test images from table 1. The top row (a-d) are close ups of parts of the four-bit input images. The optimal gain transformed images are shown. The bottom row (e-h) shows the resulting output 8-bit images, using algorithm 3 with bilateral filtering. The results are best viewed on screen.

6 Conclusion

We have in this paper introduced the notion of using the Wasserstein metric to regularize image intensity transformations. Using the optimal transport function γ and anisotropic filtering of the image we can steer the intensity transformation in a robust way and avoid discretization effects. We specifically tested our approach on the problem of bit depth expansion with state-of-the-art results.

Acknowledgments. This work was supported by ELLIIT and eSSENCE.

References

1. Brill, M.H.: The relation between the color of the illuminant and the color of the illuminated object. Color Research & Application **20**(1), 70–76 (1995)
2. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on Image Processing **16**(8), 2080–2095 (2007)
3. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: ACM SIGGRAPH 2008 classes, p. 31. ACM (2008)
4. Dobrushin, R.L.: Prescribing a system of random variables by conditional distributions. Theory of Probability & Its Applications **15**(3), 458–486 (1970)
5. http://www.imagecompression.info/test_images/ (2013)
6. Malm, H., Oskarsson, M., Warrant, E., Clarberg, P., Hasselgren, J., Lejdfors, C.: Adaptive enhancement and noise reduction in very low light-level video. In: IEEE 11th International Conference on Computer Vision, ICCV 2007, pp. 1–8. IEEE (2007)
7. Mittal, G., Jakhetiya, V., Jaiswal, S.P., Au, O.C., Tiwari, A.K., Wei, D.: Bit-depth expansion using minimum risk based classification. In: VCIP, pp. 1–5 (2012)
8. Pele, O., Werman, M.: Fast and robust earth mover's distances. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 460–467. IEEE (2009)
9. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., Toyama, K.: Digital photography with flash and no-flash image pairs. In: ACM Transactions on Graphics (TOG), vol. 23, pp. 664–672. ACM (2004)
10. Smithson, H.E.: Sensory, computational and cognitive components of human colour constancy. Philosophical Transactions of the Royal Society B: Biological Sciences **360**(1458), 1329–1346 (2005)
11. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision, 1998, pp. 839–846. IEEE (1998)
12. Ulichney, R.A., Cheung, S.: Pixel bit-depth increase by bit replication. In: Photonics West 1998 Electronic Imaging, pp. 232–241. International Society for Optics and Photonics (1998)
13. Vasershtein, L.N.: Markov processes over denumerable products of spaces, describing large systems of automata. Problemy Peredachi Informatsii **5**(3), 64–72 (1969)
14. Wan, P., Au, O.C., Tang, K., Guo, Y., Fang, L.: From 2d extrapolation to 1d interpolation: Content adaptive image bit-depth expansion. In: 2012 IEEE International Conference on Multimedia and Expo (ICME), pp. 170–175. IEEE (2012)
15. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)