COLOR IMAGE DESATURATION USING SPARSE RECONSTRUCTION

Hassan Mansour, Rayan Saab, Panos Nasiopoulos, and Rabab Ward

The University of British Columbia
Department of Electrical and Computer Engineering
2332 Main Mall, Vancouver, BC Canada V6T 1Z4
{hassanm, rayans, panosn, rababw}@ece.ubc.ca

ABSTRACT

In this paper, we propose an algorithm to estimate the true values of saturated pixels in color images. Pixel saturation occurs when at least one color channel is clipped at some value below the full dynamic range of the scene, resulting in a loss in image fidelity. The proposed algorithm is based on the assumptions that images are nearly sparse in an appropriate transform domain, and that saturated pixels can be inferred from the structure of non-saturated neighboring pixels. Consequently, we use a hierarchical windowing algorithm which selects image regions containing relatively few saturated pixels for processing. Starting with small sized regions, and progressively increasing the size, we solve a sparsity promoting constrained ℓ_1 minimization problem for each selected region to recover the saturated pixels. Moreover, we provide simulation results to show the effectiveness of our algorithm.

Index Terms— Color image desaturation, sparse signal reconstruction.

1. INTRODUCTION

The saturation of pixels in color images is a phenomenon that occurs in many image capturing and processing applications when one or more of the Red, Green, and Blue pixel values is clipped at a maximum threshold. One example of saturation in image capturing occurs when a camera with an 8-bit quantizer receives an amount of incident light at one or more of the RGB color sensors that exceeds the 255 threshold. Another example of pixel saturation can be found in the tone-mapping of a high dynamic range (HDR) image to a low dynamic range (LDR) version. In this case, the very dark and very bright pixels in the HDR image are often clipped to 0 and 255, respectively, to compress the contrast in the LDR image [1].

Pixel saturation often leads to the loss of image texture in the clipped regions. Moreover, it causes color distortions in some images when the ratio between the color channels is offset in the saturated regions. Fig. 1 shows an image in which pixels are saturated at a threshold of 200. The figure shows that saturated regions exhibit a loss in image texture, which makes the reconstruction process all the more important in applications that rely on texture processing. These applications include depth map estimation in multiexposure stereo imaging [2] and LDR to HDR inverse tone-mapping [1, 3].

The correction of saturated image regions can be viewed as an image restoration problem, the goal of which is to reverse the distortions caused by imaging systems. In fact, several techniques have been proposed for image restoration in various contexts, including inpainting, disocclusion, inverse tone-mapping, and denoising [4–6].

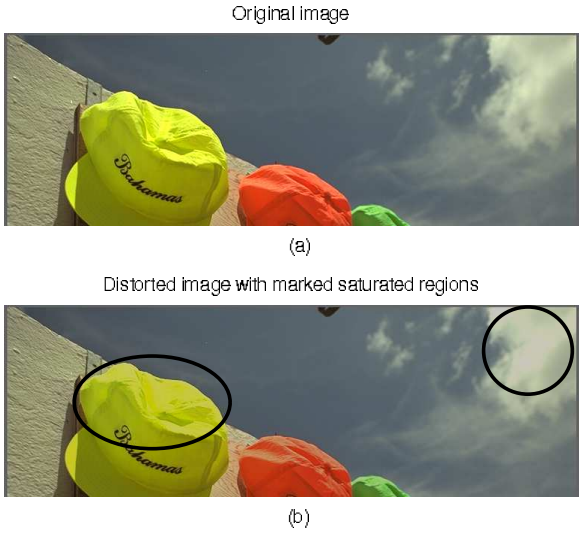


Fig. 1: Illustration of an image (Kodim03) with a saturation threshold of 200. The original image (a) is clipped at the threshold level resulting in the distorted image (b) with the marked clipped regions.

Most of these techniques share an underlying assumption that images can be represented using a probabilistic graphical model such as a Markov Random Field (MRF) [7, 8]. The reconstruction of the missing regions is then achieved via Maximum A-Posteriori (MAP) estimation. We refer the reader to [4] and the references within for more information about image disocclusion and inpainting techniques. Zhang et al. [9] tackle a closely related problem where image pixels are saturated due to the overexposure of the camera sensors. They present an algorithm based on Bayesian estimation that uses the sensor responses from non-saturated channels to correct the saturated regions.

In this paper, we address the problem of recovering pixel values that have been clipped at a known saturation threshold. Specifically, we will make use of the following three assumptions in designing our desaturation algorithm:

A1. Natural images admit sparse representations in appropriate transform domains (e.g., wavelets, DCT),

A2. The values of the reconstructed pixels should be greater than or equal to the saturation threshold, and

A3. The structure of the saturated regions is similar to that of the unsaturated neighboring pixels.

Using these prior assumptions, we formulate the desaturation problem as a set of constrained optimization problems where we min-

imize the ℓ_1-norm of the discrete cosine transform (DCT) coefficients of image regions containing some saturated pixels. The regions are selected based on a hierarchical windowing approach that ensures a small percentage of saturated pixels per window. The constraints force the desaturated pixels to exceed the saturation threshold while not distorting the remaining pixels. The idea behind this is to use small windows to recover pixels located near the boundaries of clipped regions, then using larger windows, to recover the remaining clipped pixels. Note that although the application we target in this paper is pixel desaturation, the techniques we propose can be used in other image restoration applications with limited modification.

The remainder of the paper is organized as follows. In Section 2 we formulate the desaturation problem in the context of sparse signal reconstruction. Specifically, a constrained ℓ_1 minimization approach is proposed to recover saturated pixels in local image regions. In Section 3 we present a hierarchical windowing approach that identifies local regions of progressively increasing sizes on which to solve the optimization problem. We then propose in Section 4 modifications to the above approach where the ℓ_1 cost function is replaced by an image dependent *weighted* ℓ_1 function. A post-processing stage is then proposed where the reconstructed pixels are smoothed and scaled. Finally, the performance of the proposed algorithm is evaluated in Section 5.

2. THE SPARSE RECONSTRUCTION PROBLEM

The problem of recovering sparse signals in $\mathbb{R}^N$ - or sparse representations of signals in $\mathbb{R}^N$ - from fewer measurements has received a lot of attention lately. This has come due to the advent of compressed sensing and related applications (see, e.g., [10–13]), as well as due to the role of sparsity in applications such as transform coding (see e.g., [14, 15]) among many others.

2.1. Formal definition

Let $\Sigma_s^N := \{x \in \mathbb{R}^N : |supp(x)| \leq s\}$ be the set of all s-sparse signals and define compressible signals as those vectors that can be well approximated in Σ_s^N. Henceforth, A denotes a matrix in $\mathbb{R}^{M\times N}$ with $M < N$ and denote by

$$b = Ax + e \tag{1}$$

the (possibly noisy) measurement vector from which we must recover $x \in \Sigma_s^N$. Thus, the system of equations in (1) is underdetermined and (if consistent) admits infinitely many solutions from which we must choose the "correct solution". In fact, when $e = 0$, and if A is in general position (i.e., non of its $M \times M$ submatrices are rank deficient) and $s < M/2$, then $x_0 = x$ [16] where

$$x_0 := \arg\min_y \|y\|_0 \text{ subject to } Ay = b. \tag{2}$$

Above, $\|y\|_0$ is the so-called ℓ_0-norm of y and is simply the number of non-zero entries of y. While (2) is a combinatorial problem and thus extremely hard to solve for large N and M, a large body of research has recently been focused on the properties of x_1,

$$x_1 := \arg\min_y \|y\|_1 \text{ subject to } Ay = b. \tag{3}$$

Contrary to x_0, x_1 is obtained via a convex optimization problem (3) which is easy to solve. Under appropriate conditions on A and if s is small enough, one can show that $x_1 = x$. More generally, if x is compressible, and if A obeys certain conditions, then one can obtain reasonable bounds on $\|x_1 - x\|_2$. For a flavor of some of these motivating results see, e.g., [10, 11, 13]).

2.2. Sparse reconstruction of saturated images

Let $I \in \mathbb{R}^{m\times n}$ be an image and suppose that $Z \in \mathbb{R}^{m_1\times n_1}$ is a region of the image with saturated pixels. Let $z \in \mathbb{R}^N, N = m_1 n_1$, be a lexicographic rearrangement of Z into a column vector. Thus, z_j, the jth entry of z corresponds to one pixel in Z. Suppose that the linear operator $T : \mathbb{R}^N \to \mathbb{R}^N$ mimics the action of the two-dimensional DCT (on an image that has been lexicographically rearranged into a column as above). Then $x = Tz$ is a compressible vector, i.e., it is almost sparse. This observation drives such compression algorithms as JPEG (see, e.g., [7]).

To make the discussion more pertinent to the problem at hand, let t be the known pixel saturation threshold and suppose that $\Lambda = \{j : z_j \geq t\}$, the set of indices corresponding to saturated pixels, is of cardinality $N - M$. Similarly, let $\Lambda^c = \{1, ..., N\} \setminus \Lambda$ be the set of indices corresponding to the M non-saturated pixels. Clearly, any reconstruction algorithm must not distort the values of the unsaturated pixels. Therefore, we define a restriction matrix $R_{\Lambda^c} \in \mathbb{R}^{M\times N}$ that when applied to the image Z returns only the unsaturated pixels in the image. This matrix is constructed by setting the rows of R_{Λ^c} equal to the rows of the identity matrix indexed by Λ^c. Similarly, define R_Λ, the restriction matrix corresponding to the saturated pixels. Thus, from a sparse reconstruction point of view (3) where $Ay = b$ serves as constraint, we identify the analogous constraint $Ay = R_{\Lambda^c} z$, with $A = R_{\Lambda^c} T$. Moreover, since the saturated pixels are clipped at a threshold t, we impose a second constraint that a reconstruction algorithm must satisfy: $R_\Lambda Ty \geq t$. Here, the inequality is meant to be understood element-wise. Thus, for every image region with clipped pixels, we propose to solve

$$x^* := \arg\min_y \|y\|_1 \text{ s.t. } R_{\Lambda^c} Ty = R_{\Lambda^c} z, \quad R_\Lambda Ty \geq t \tag{4}$$

to recover an estimate x^* of the DCT coefficients of the selected region. Next, we recover the pixel values via $z^* = T^{-1}x^*$.

Note that in justifying our use of (4), we have used assumptions A1 and A2 of Section 1. It is also worth noting that (4) can be derived as the MAP estimator of the DCT coefficients of the image region, under the assumption that they are (sparsity promoting) independent identically distributed Laplacian random variables. We omit the standard derivation for lack of space.

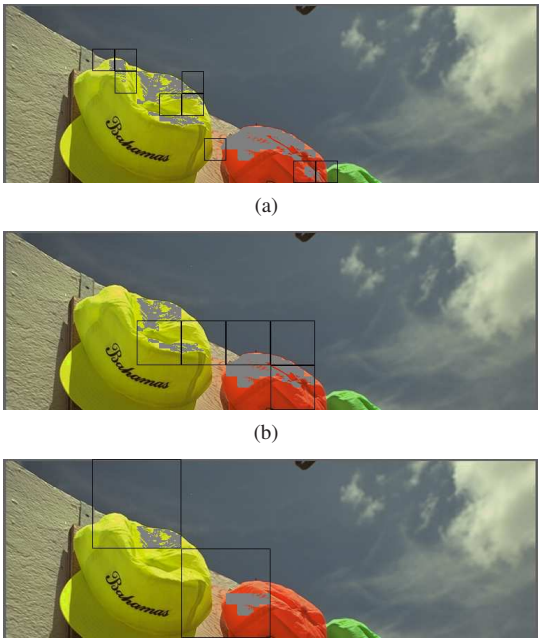
3. HIERARCHICAL WINDOWING

In this section, we propose a hierarchical algorithm that applies (4) first on small partially saturated square image regions. Having reconstructed the saturated pixels in the small windows, the algorithm then seeks larger square regions where (4) is solved. This procedure is repeated until no saturated pixels remain in the image.

More formally, define the hierarchical levels $l = 1, 2, ..., L$, where $L \leq \log_2(\min(m, n)) - 1$, and let the window size at each level be $p_l = 2^{l+1}$. At each level l, divide I into non overlapping $p_l \times p_l$ blocks and select the ones where the percentage of clipped pixels is less than a predefined threshold ν. On each such block Z, solve the optimization problem (4) to obtain estimates of the saturated pixels. Note that once saturated pixels are corrected for, they are no longer considered saturated in future steps. Finally once $l = L$ solve (4) on all $p_l \times p_l$ region with any saturated pixels remaining.

Remark. Throughout this section, we made the simplifying assumption that $\log_2(m), \log_2(n) \in \mathbb{N}$. If that were not the case, Z can be symmetrically extended (for example) by reflecting an appropriate number of pixels, first horizontally then vertically across the right and top boundaries.

It is worth noting that in keeping with assumption A3, when the image contains a large region with saturated pixels the algorithm will often use small windows to recover the boundary pixels of the region first, then it will move inward to fill in the central pixels using larger windows. This observation is highlighted in Figure 2.



(a) (b) (c)

Fig. 2: Illustration of the level-based block extraction employed in the hierarchical windowing algorithm corresponding to (a) 32x32, (b) 64x64, and (c) 128x128 block sizes. The gray shaded areas indicate the saturated regions at each level.

4. MODIFICATIONS AND POST-PROCESSING

In this section, we modify the cost function of (4) by introducing image dependent weighting. We also propose post-processing the image by smoothing and scaling the de-saturated pixels.

4.1. Weighted ℓ_1 minimization

The hierarchical algorithm proposed in the previous section ensures that reconstruction is only performed on image regions Z with at most a small (but non-zero) percentage ν of saturated pixels. The idea behind this is to use the local structure in the vicinity of the saturated pixels of Z during reconstruction. Here, we loosely use the term structure to refer to the dominant transform subbands. On the other hand, the ℓ_1 cost function of (4) assigns equal weights to all the transform coefficients y. In other words, it does not emphasize the structure imposed by the large majority of the pixels of Z. We propose to remedy this by introducing a block adaptive weighting that penalizes deviations from the existing local structure imposed by the non-saturated pixels. Let w be a vector of positive weights,

$$w = (\log(|Tz| + 1) + c)^{-1}, \tag{5}$$

for some constant c. For example, in this paper $c = 0.5$ is used. Consequently, the hierarchical windowing algorithm now solves, instead of (4), the constrained *weighted* ℓ_1 minimization problem

$$x^* := \arg\min_y \|y\|_{1,w} \text{ s.t. } R_{\Lambda^c} T y = R_{\Lambda^c} z,\ R_\Lambda T x \geq t. \tag{6}$$

where $\|y\|_{1,w} = \sum_{i=1}^{N} |y_i w_i|$ is the weighted ℓ_1 norm of y.

4.2. Smoothing and scaling

Let I^* be the image obtained as a result of the processing described in the previous sections. To deal with high frequency noise in I^*, we propose to smooth the reconstructed saturated regions using a circular symmetric Gaussian filter. This is performed by extracting the reconstruction residual $I_r = (I^* - I)$ and filtering it with a Gaussian filter g , with

$$g(n_1, n_2) = \frac{e^{-\frac{n_1^2+n_2^2}{2\sigma^2}}}{\sum_{n_1,n_2\in\Omega} e^{-\frac{n_1^2+n_2^2}{2\sigma^2}}}, \tag{7}$$

for $(n_1, n_2) \in \Omega$. Here, Ω is a square region centered around zero. Following this smoothing operation, the residual is scaled to fill the range between the saturation threshold and the maximum value allowed by the color bit-depth and then added to I to obtain $\hat{I}$. Thus,

$$\hat{I} = I + a \cdot (I_r * g)\,, \tag{8}$$

where $*$ is the convolution operator and a is the scaling parameter. To finish the process, let $S = \{(i, j) : I(i, j) \text{ is saturated}\}$. We obtain the final desaturated image $\widetilde{I}$, with

$$\widetilde{I}(i,j) = \begin{cases} I(i,j), & (i,j) \notin S \\ \hat{I}(i,j), & (i,j) \in S. \end{cases} \tag{9}$$

All the steps involved in the proposed approach are summarized in Algorithm 1.

Algorithm 1 Hierarchical Windowing Algorithm

for $l = 1$ to L **do**
  Set $p_l = 2^{l+1}$.
  if $l < L$ **then**
    Using non-overlapping windows, locate all $p_l \times p_l$ blocks with less than $\nu\%$ saturated pixels.
  else if $l = L$ **then**
    Using non-overlapping windows, locate all $p_L \times p_L$ blocks with any number of saturated pixels.
  end if
  For each such block solve (6) and set $z^* = T^{-1}x^*$.
  Consider the recovered pixels unsaturated from now on.
  if No more saturated pixels **then**
    Terminate algorithm.
  end if
end for
Post process the results as described in Section 4.2 to obtain $\widetilde{I}$.

Table 1: Comparison of the Reconstruction Performance

Image	Saturated Image			Proposed Algorithm			Zhang et al.'s Algorithm [9]		
	SNR	PSNR	Avg. SSIM	SNR	PSNR	Avg. SSIM	SNR	PSNR	Avg. SSIM
kodim03	30.79	38.33	0.9959	35.09	42.63	0.9978	31.53	39.06	0.9964
kodim07	39.20	46.30	0.9993	43.54	50.64	0.9997	42.70	49.80	0.9996
kodim17	36.95	45.95	0.9997	39.87	48.87	0.9999	39.40	48.40	0.9999
kodim23	27.47	34.21	0.9886	30.14	36.88	0.9907	29.76	36.50	0.9905

5. PERFORMANCE EVALUATION

In order to demonstrate the effectiveness of our algorithm, we use four images from the Kodak Lossless True Color Image Suite[1]. The images are 24bit/pixel RGB color images where each color channel can hold a value between 0 and 255.

For testing purposes, we impose a saturation threshold of 220 on these unsaturated images. We apply our algorithm (with $L = 7$, $\nu = 25\%$, $\sigma = 3.5$, and Ω covering a 27x27 pixel smoothing window) to recover the saturated pixels. We then compare the performance with that of the algorithm proposed by Zhang et al. [9] and report the results in Table 1. We show the reconstructed image quality in terms of the signal-to-noise-ratio (SNR), peak-signal-to-noise-ratio (PSNR), and the average Structural SIMilarity index (SSIM) over the three color channels. The results show that our algorithm achieves significant gains over the saturated image (average of 3.56dB in SNR, 3.56dB in PSNR, and 0.002 in SSIM). Our algorithm also outperforms Zhang et al.'s algorithm.

As a final remark, it is worth mentioning that for this paper instances of the constrained ℓ_1 minimization problem (6) are solved using a modification of the SPGL1 solver [17] while the SPARCO [18] environment was used for prototyping the various operators.

Acknowledgment

The authors would like to thank Ewout Van Den Berg for providing the modified SPGL1 code as well as Di Xu and Colin Doutre for providing a version of Zhang et al.'s algorithm and for fruitful discussions.

6. REFERENCES

[1] Rafal Mantiuk, Grzegorz Krawczyk, Radoslaw Mantiuk, and Hans P. Seidel, "High dynamic range imaging pipeline: perception-motivated representation of visual content," 2007, vol. 6492, SPIE.

[2] A. Troccoli, Sing Bing Kang, and S. Seitz, "Multi-view multi-exposure stereo," in *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, June 2006, pp. 861–868.

[3] Allan G. Rempel, Matthew Trentacoste, Helge Seetzen, H. David Young, Wolfgang Heidrich, Lorne Whitehead, and Greg Ward, "Ldr2hdr: on-the-fly reverse tone mapping of legacy video and photographs.," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 39, 2007.

[4] Z. Tauber, Ze-Nian Li, and M.S. Drew, "Review and preview: Disocclusion by inpainting for image-based rendering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 4, pp. 527–540, July 2007.

[5] Francesco Banterle, Patrick Ledda, Kurt Debattista, and Alan Chalmers, "Inverse tone mapping," in *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, 2006, pp. 349–356.

[6] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[7] Volkan Cevher, Marco F. Duarte, Chinmay Hegde, and Richard G. Baraniuk, "Sparse signal recovery using markov random fields.," in *NIPS*. 2008, pp. 257–264, MIT Press.

[8] Patrick Perez, "Markov random fields and images," 1998, CWI quarterly, volume 11.

[9] Xuemei Zhang and David H. Brainard, "Estimation of saturated pixel values in digital color imaging," *J. Opt. Soc. Am. A*, vol. 21, no. 12, pp. 2301–2310, 2004.

[10] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[11] E. J. Candès and T. Tao, "Decoding by linear programming.," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 489–509, 2005.

[12] Y. Tsaig and D. Donoho, "Extensions of compressed sensing," *Signal Process.*, vol. 86, no. 3, pp. 549–571, 2006.

[13] D. Donoho, "Compressed sensing.," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[14] S. Chen, D. Donoho, and M.A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.

[15] R.M.F. Ventura, P. Vandergheynst, and P. Frossard, "Low-rate and flexible image coding with redundant representations," *IEEE Transactions on Image Processing*, vol. 15, no. 3, 2006.

[16] D. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 5, pp. 2197–2202, 2003.

[17] E. van den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[18] Ewout van den Berg, Michael P. Friedlander, Gilles Hennenfent, Felix J. Herrmann, Rayan Saab, and Özgür Yilmaz, "Algorithm 890: Sparco: A testing framework for sparse reconstruction," *ACM Trans. Math. Softw.*, vol. 35, no. 4, pp. 1–16, 2009.

[1] Available from http://r0k.us/graphics/kodak/