# Angular-based pre-processing for image denoising

Javier Vazquez-Corral and Marcelo Bertalmío

***Abstract*—There is not a large research on how to use color information for improving results in image denoising. Currently, most of the methods modify the color space from sRGB to an opponent-like one as better results are obtained, but out of this conversion, color is mostly ignored in the image denoising pipelines. In this work, we propose a color decomposition to pre-process an image before applying a typical denoising. Our decomposition consists in obtaining a set of images in the spherical coordinate system, each of them with the origin of the spherical transformation in a different color value. These color values, that we call color centers, are defined so as to be far away from the dominant colors of the image. Once in the spherical coordinate system, we perform a mild denoising operation with some state-of-the-art method in the angular components. Then, we convert these images back to sRGB, and we merge them depending on the distance between the color of each pixel and the color centers. Finally, we denoise the pre-processed image with the same state-of-the-art method used in our pre-processing. Experiments show that our method outperforms the results of directly applying the denoising method on the input image for different state-of-the-art denoising methods.**

***Index Terms*—Image Denoising, Non-local methods, Angular representation.**

## I. Introduction

Noise is present in images due to the inherent physical and technological limitations of the cameras. The presence of noise degrades the quality of the captured images. Therefore, image denoising is a must-have step in the digital imaging processing pipeline.

Research into image denoising spans on more than three decades. We can split the different image denoising methods in two different categories: local and non-local. The former denoises a given pixel by taking into account only its neighbors. Examples of local methods are those based on the reduction of the Total Variation (TV) of the image such as the Rudin-Osher-Fatemi (ROF) model [1]. These methods are prone to removing details from edges and textures, and may introduce staircasing artifacts.

Non-local methods are considered as state-of-the-art approaches. In particular, patch-based methods use the self-similarity found in natural images for denoising the input image. Their main drawback is a high algorithmic complexity, as they combine information from patches all over the image domain. Examples of this type of methods are the Non-Local Means (NLM) algorithm [2], the UINTA algorithm [3], and the Block-Matching and 3D Filtering (BM3D) algorithm [4]. BM3D combines a patch-based approach with the use of frequency filtering. A number of extensions to non-local methods have been proposed. For example Zontak *et al.* [5] consider different scales in the search for patch correspondences. Zoran *et al.* [6] propose the EPLL method that considers the log likelihood of the patches to select the best ones to be used in the restoration. Talebi *et al.* [7] present a method that locally controls the denoising strength via a spatially adaptive iterative filter, and Tarebi and Milanfar [8] propose a method that computes a global filter to denoise the image. For a comprehensive review of image denoising algorithms we refer the reader to the work of Lebrun *et al.* [9].

Surprisingly, even when most of the images in the world are in color, very little research has been devoted in how to benefit from the color information for this problem. Almost all image denoising methods are based on applying the same algorithm in each channel of the color image, either on the original RGB space, or most commonly in recent times in the opponent space suggested by BM3D [4]. In this work we pre-process the input image to improve the capabilities of the denoising methods, with a focus on non-local ones. Our pre-processing is inspired by the fact that image colors can be clustered together, and that natural images are typically dominated by only a few colors [10]. In particular, our pre-processing applies a mild denoising (i.e. a denoising that considers a small standard deviation parameter) in a color decomposition of the image specifically designed for this purpose. This mild denoising is performed by the same method that will later be used to denoise the pre-processed image. The idea of the decomposition is to obtain a set of image representations in the spherical coordinate system, each of them having the origin in points that are far away from the dominant colors present in the image. In this way, all the pixels with color values close to the dominant color will have very similar angular components, thus allowing us to apply a mild denoising in this step in order to improve the final result of the whole denoising pipeline.

## II. Angular-based pre-processing

We explain our angular-based pre-processing in this section. The pipeline of our approach is presented in Figure 1. In a nutshell, we can divide our method in the following steps: i) obtaining a set of color values (we call them color centers and we denote them as $c_i$) at a large distance of the dominant colors, ii) for each $c_i$, converting the original color image to the spherical domain (being $c_i$ the origin of the spherical transformation) and denoising the two angular components of the representation in a mildly manner, iii) coming back to sRGB and merging all the different images depending on the distance between the color value of each pixel in the original image and the different color centers. The pre-processed image $I_{preproc}$ is finally used to feed a typical denoising method.

Authors are with Universitat Pompeu Fabra, E-08018, Barcelona, Spain (e-mails: {javier.vazquez, marcelo.bertalmio}@upf.edu). They are supported by the European Research Council, Starting Grant ref. 306337, by the Spanish government FEDER Fund, grant ref. TIN2015-71537-P(MINECO/FEDER,UE), and by the Icrea Academia Award. The work of J. Vazquez-Corral was supported by the Spanish government under Grant IJCI-2014-19516)

Manuscript received

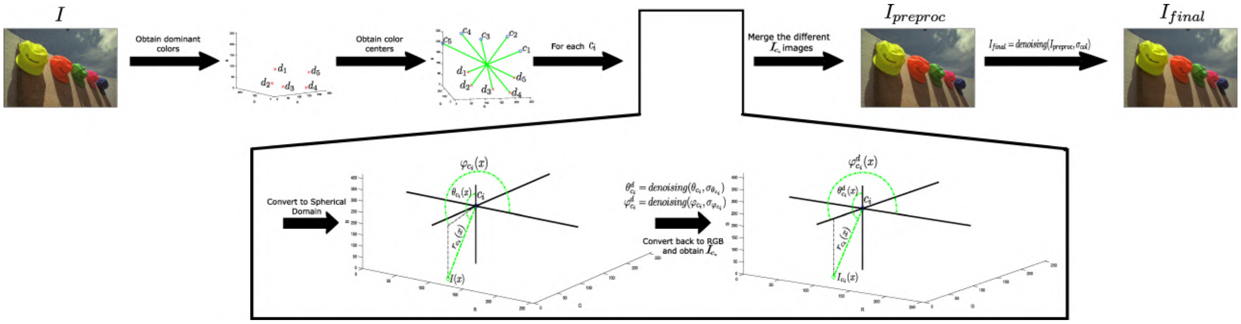
$I$
Obtain dominant colors
$d_1$ $d_2$ $d_3$ $d_4$ $d_5$
Obtain color centers
$c_1$ $c_2$ $c_3$ $c_4$ $c_5$
For each $c_i$
Merge the different $I_{c_i}$ images
$I_{preproc}$
$I_{final} = denoising(I_{preproc}, \sigma_{col})$
$I_{final}$
Convert to Spherical Domain
$\varphi_{c_i}(x)$
$\theta_{c_i}(x)$
$c_i$
$r_{c_i}(x)$
$I(x)$
$\theta^d_{c_i} = denoising(\theta_{c_i}, \sigma_{\theta_{c_i}})$
$\varphi^d_{c_i} = denoising(\varphi_{c_i}, \sigma_{\varphi_{c_i}})$
Convert back to RGB and obtain $I_{c_i}$
$\varphi^d_{c_i}(x)$
$\theta^d_{c_i}(x)$
$I_{c_i}(x)$
R
G
B

Fig. 1. Pipeline of our method. Given an input image $I$ we obtain the color centers; for each of them, we convert the input image to the spherical domain (centering the origin of the spherical transformation on the color center) and we denoise the two angular components of this representation. We come back to sRGB and we merge all the different images. Finally, the pre-processed image is used to feed a typical denoising method.

### A. Obtaining the set of color centers

The first step in our approach deals with the obtention of the dominant colors of the image (we call them $d_i$). To this end, we follow the procedure outlined in [11] to obtain what they call color pivots. In short, the work of [11] uses the RAD method from Vazquez *et al.* [12]. Given an image $I$, RAD provides us with different sets of color values. Each of these sets is called a ridge of the color histogram and it is composed by local maxima values of a creaseness operator that are connected following the lines of shallowest gradient descent (representing in a single ridge small variations of an object color due to specular and shadow effects). Then, for a particular ridge $R = \{r_1, \cdots, r_n\}$, the dominant color $d_i$ is the one that fulfills

$$d_i = \arg\max_{c \in R_i} H(c), \tag{1}$$

where $R_i$ denotes the $i$-th ridge, and $H(\cdot)$ stands for the color histogram (i.e. the dominant color is the color value of the ridge with maximum histogram value). Once the dominant colors have been obtained, we look for the color centers $c_i$. As outlined before, $c_i$ should be far away from the dominant color. To this end, given a particular dominant color $d_i$ we first call $p$ the line that passes through $d_i$ and the center of the color space (in case of sRGB the vector $[127.5\ 127.5\ 127.5\ ]$). Then, we define the color center $c_i$ as the point of the RGB cube that belongs to $p$ and is at the maximum distance of $d_i$. Mathematically,

$$p = d_i + \eta([127.5 \quad 127.5 \quad 127.5] - d_i), \tag{2}$$

$$c_i = \arg\max_{x \in p, x \in RGB_{cube}} d(x, d_i), \tag{3}$$

where $RGB_{cube}$ denotes the RGB color cube, and $d$ denotes the Euclidean distance.

### B. Obtaining the new set of images and denoising them

Given the original image $I$ and any of the color centers $c_i$, the spherical coordinates centered in $c_i$ for a pixel $x$ in the image $I$ are obtained as

$$r_{c_i}(x) = \sqrt{(I^r(x) - c_i^r)^2 + (I^g(x) - c_i^g)^2 + (I^b(x) - c_i^b)^2}$$

$$\theta_{c_i}(x) = \tan^{-1}\left(\frac{\sqrt{(I^r(x) - c_i^r)^2 + (I^g(x) - c_i^g)^2}}{(I^b(x) - c_i^b)}\right)$$

$$\varphi_{c_i}(x) = \tan^{-1}\left(\frac{I^g(x) - c_i^g}{I^r(x) - c_i^r}\right), \tag{4}$$

where $r$,$g$, and $b$ denote the different color channels. Then, we perform a mild denoising (i.e. a denoising with a small standard deviation parameter) in the two angular components

$$\begin{aligned} \theta^d_{c_i} &= denoising(\theta_{c_i}, \sigma_{\theta_{c_i}}) \\ \varphi^d_{c_i} &= denoising(\varphi_{c_i}, \sigma_{\varphi_{c_i}}), \end{aligned} \tag{5}$$

where $d$ stands for denoised, $\sigma_{\theta_{c_i}}$, $\varphi_{\theta_{c_i}}$ are the noise standard deviation parameters, and the denoising method can be any present in the literature.

### C. Back to sRGB and merging

In this point, for each $c_i$ we come back to the original sRGB space by using the denoised quantities ($\theta^d_{c_i}$ and $\varphi^d_{c_i}$). Thus, we obtain a set of different sRGB images $I_{c_i}$

$$\begin{aligned} I^r_{c_i}(x) &= r_{c_i}(x)\sin(\theta^d_{c_i}(x))\cos(\varphi^d_{c_i}(x)) + c^r_i \\ I^g_{c_i}(x) &= r_{c_i}(x)\sin(\theta^d_{c_i}(x))\sin(\varphi^d_{c_i}(x)) + c^g_i \\ I^b_{c_i}(x) &= r_{c_i}(x)\cos(\theta^d_{c_i}(x)) + c^b_i. \end{aligned} \tag{6}$$

We merge the different $I_{c_i}$ in a single pre-processed image as follows. Let us note that $r_{c_i}(x)$ marks the distance between the color center $c_i$ and the color value at the specific pixel $x$. Our hypothesis is that pixels with values at larger distances are better denoised. The rationale behind this idea is that at larger distances, the values in the angles $\theta_{c_i}$ and $\varphi_{c_i}$ for the colors close to that of pixel $x$ are closer, and therefore easier to denoise. Then, for each pixel $x$ we define the weights $w_{c_i}(x)$ as

$$w_{c_i}(x) = \left(\frac{r_{c_i}(x)}{\sum_{c_i} r_{c_i}(x)}\right)^{\alpha}, \tag{7}$$

and we obtain the pre-processed image as follows

$$I_{preproc}(x) = \sum_{c_i} \frac{w_{c_i}(x)}{\sum_{c_i} w_{c_i}(x)} I_{c_i}(x). \tag{8}$$

The last step of our method consists on denoising the pre-processed image with the same method used in Eq. 5

$$I_{final} = denoising(I_{preproc}, \sigma_{col}) \tag{9}$$

where $\sigma_{col}$ is the noise standard deviation parameter for the color image, and the denoising method should be the same one used in Eq.5.

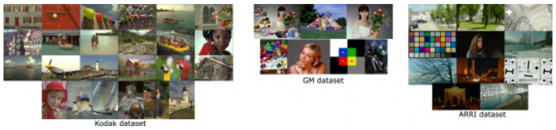

Fig. 2. Datasets used in our experiments.

## III. Experiments

We run our method on 3 different datasets: Kodak [13], which is the standard denoising dataset, a dataset that we call Gamut Mapping (GM) as it contains images usually used for addressing the gamut mapping problem [14], and the 11 images from the ARRI dataset [15] captured using a Bayer Pattern. These different datasets are shown in Figure 2. For all the datasets, given the clean ground-truth image, we add to it white noise with a Gaussian distribution of different standard deviations $\sigma$ to create our input images. In this paper we consider $\sigma = 10, 20, 30, 40, 50$ and 60. For all the different methods and noise levels we follow the same strategy. We search for the optimal parameters in the standard Kodak dataset, and we later process the other datasets with these same parameters, therefore following a train/test paradigm. We make use of the color peak signal to noise ratio (CPSNR) as the measure to optimize because it is a color metric widely used in image denoising. For all the tested image denoising methods, we compare two results: i) the result obtained by applying a particular method directly to the input color image, ii) the result obtained by first pre-processing the image as proposed in section II followed by applying exactly the same procedure of i) with the same parameters to our pre-processed image. We now discuss the results for each specific method considered in the paper. Regarding the parameters of our method, the values for $\sigma_{\theta_{c_i}}$ and $\sigma_{\varphi_{c_i}}$ have been obtained by optimizing the results for the BM3D method in the Kodak dataset, and we set the value $\alpha = 10.6$ in Eq.(7), and the maximum number of color centers in the image to 8 for all the cases.

### A. BM3D

BM3D [4] is considered one of the state-of-the-art methods in image denoising. In this paper, we use the Matlab code provided by the authors, and run it in the opponent color space for the sRGB images. In the top part of Table I we show the parameters used by BM3D, for both the color images and for our angular components (third and fourth row), as well as the results obtained by the original method, and our method. (fifth and sixth row). We can see that our approach outperforms the original BM3D method. We can see that the improvement is larger for higher noise levels, and that it is smaller in the case of the Kodak dataset. Let us also note that for the noise level $\sigma = 60$ our improvement reaches up to 0.67, 0.98, and 1.18 dBs for the Kodak, GM, and ARRI datasets, respectively. Finally, in Figure 3 we can see the effect of applying our pre-processing. From left to right we present the original noisy image, the ground truth, the result of the original BM3D method, our result, and the difference images for both the BM3D method and our result. The difference image is computed as follows: i) we calculate the absolute difference between the denoised and the original images and ii) we add the values obtained for the three color channels. Therefore, a darker difference image indicates a better denoising. The difference images have been scaled per scene so that the value of 1 is given to the maximum error, and the gamma non-linearity of sRGB has been added for visualization purposes. We can see that the BM3D denoised image presents a more blurred and washed-up appearance in comparison to the results given by applying our preprocessing. This is specially evident in the color of the swimming pool water and in the seats in the tribune for the top image, and in the color of the sky for the bottom image.

### B. Non-local Means

The Non-Local Means (NLM) algorithm [2] made a breakthrough in the image denoising literature. NLM introduced the idea of combining redundant information from patches that span all the image domain. It is still one of the most used methods. Also, it has been shown to perform comparable to BM3D in psychophysical tests [16]. In this paper, we run the NLM algorithm in the opponent color space since it is the color space where better results are obtained. Different parameters for the luminance and chrominance channels are used in the search for the best results. In the bottom partt of Table I we present the parameters used, the results of the original NLM method, and the results of applying NLM to pre-process our angular components. As in the case of BM3D, our improvement is larger for higher noise levels. Our improvement reaches up to 0.47, 0.71, and 1.07 dBs at the higher noise level for the Kodak, GM, and ARRI datasets, respectively.

### C. Other methods

Finally, we test to see if our approach can also be used for other methods. In particular we try the EPLL [6] method and the Field-of-Experts (FOE) [17] method. The later one was selected to check if our approach can also be used for methods that are not non-local. Results for these two methods are shown in Table II. In this case, we reduce the size of the input image for computational reasons by performing an uniform sampling in order to respect the original colors of the image. Images in the Kodak and the GM dataset are reduced by a factor of 2 in each dimension, while images in the ARRI datasets are reduced by a factor of 4 in each dimension. As in the case of NLM, the parameters for the EPLL method are obtained separately for the luminance and chrominance components in the opponent color space. We can see that our pre-processing approach in the EPLL method improves the original one for almost all the cases, reaching to a maximum improvement of $0.38, 0.54$, and $0.65$ dBs for the Kodak, GM and ARRI datasets, respectively. In the case of the FOE algorithm, we run the method in the YCbCr color space using the code released by the authors. We can see that our method once again improves the results for the three different datasets, reaching up to a $0.35, 0.27$, and $0.49$ dBs improvement, respectively.

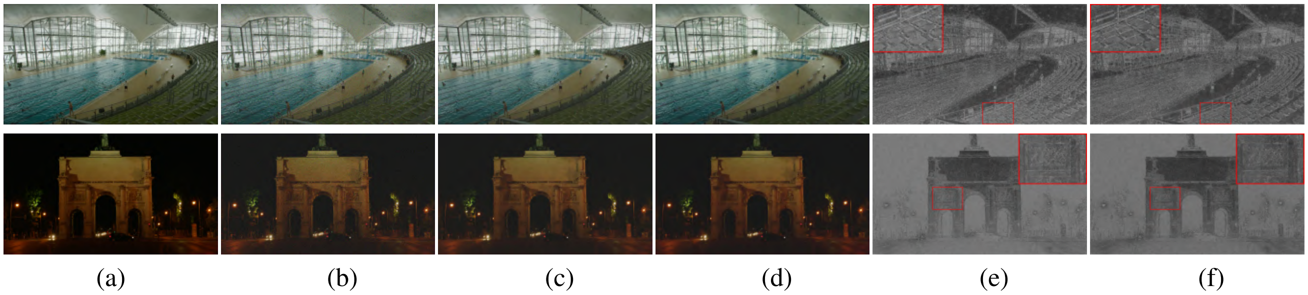

Fig. 3. An example of our result for the BM3D case. (a) Original image, (b) noisy image, (c) result of BM3D, (d) result of our approach, (e) difference image for BM3D, (f) difference image for our result.

TABLE I
CPSNR RESULTS FOR THE BM3D [4] CASE (TOP), AND THE NLM [2] CASE (BOTTOM). AVERAGE FOR EACH DATASET.

| | Kodak dataset (training data) | | | | | | GM dataset | | | | | | ARRI dataset | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Noise | 10 | 20 | 30 | 40 | 50 | 60 | 10 | 20 | 30 | 40 | 50 | 60 | 10 | 20 | 30 | 40 | 50 | 60 |
| $\sigma$(Color) | 10 | 19 | 29 | 41 | 45 | 53 | 10 | 19 | 29 | 41 | 45 | 53 | 10 | 19 | 29 | 41 | 45 | 53 |
| $\sigma(\theta, \varphi)$ | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 |
| BM3D | 36.45 | 32.63 | 30.39 | 28.57 | 27.36 | 26.12 | 37.19 | 33.06 | 30.41 | 28.23 | 26.65 | 25.12 | **38.19** | 34.47 | 31.71 | 29.50 | 27.56 | 25.86 |
| Ours-BM3D | **36.46** | **32.70** | **30.49** | **28.73** | **27.77** | **26.79** | **37.20** | **33.16** | **30.60** | **28.56** | **27.29** | **26.10** | 38.18 | **34.61** | **32.01** | **30.04** | **28.44** | **27.04** |
| Improv. | 0.01 | 0.07 | 0.10 | 0.16 | 0.41 | 0.67 | 0.01 | 0.10 | 0.19 | 0.33 | 0.64 | 0.98 | -0.01 | 0.14 | 0.30 | 0.54 | 0.88 | 1.18 |
| $\sigma$(Lum, Chr) | 5,6 | 10,12 | 15,20 | 19,29 | 23,40 | 27,49 | 5,6 | 10,12 | 15,20 | 19,29 | 23,40 | 27,49 | 5,6 | 10,12 | 15,20 | 19,29 | 23,40 | 27,49 |
| $\sigma(\theta, \varphi)$ | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 |
| NLM | 35.21 | 31.40 | 29.19 | 27.60 | 26.21 | 24.97 | 35.87 | 31.81 | 29.15 | 27.15 | 25.48 | 24.04 | 36.46 | 32.79 | 30.23 | 28.21 | 26.47 | 24.92 |
| Ours -NLM | **35.22** | **31.44** | **29.28** | **27.80** | **26.52** | **25.44** | **35.88** | **31.88** | **29.31** | **27.45** | **25.95** | **24.75** | **36.47** | **32.90** | **30.51** | **28.75** | **27.26** | **25.99** |
| Improv. | 0.01 | 0.04 | 0.09 | 0.20 | 0.31 | 0.47 | 0.01 | 0.07 | 0.16 | 0.30 | 0.47 | 0.71 | 0.01 | 0.11 | 0.28 | 0.54 | 0.79 | 1.07 |

TABLE II
CPSNR RESULTS FOR THE EPLL [6] AND FOE [17] CASES. AVERAGE FOR EACH DATASET. KODAK AND GM IMAGES HAVE BEEN REDUCED BY A FACTOR OF 2 IN EACH DIMENSION. ARRI IMAGES HAVE BEEN REDUCED BY A FACTOR OF 4 IN EACH DIMENSION.

| | Kodak dataset (training data) (1/4) | | | | | | GM dataset (1/4) | | | | | | ARRI dataset (1/16) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Noise | 10 | 20 | 30 | 40 | 50 | 60 | 10 | 20 | 30 | 40 | 50 | 60 | 10 | 20 | 30 | 40 | 50 | 60 |
| $\sigma$(Color) | 5 | 10 | 17 | 25 | 40 | 148 | 5 | 10 | 17 | 25 | 40 | 148 | 5 | 10 | 17 | 25 | 40 | 148 |
| $\sigma(\theta, \varphi)$ | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 |
| FOE | 33.64 | 29.56 | 27.41 | 25.96 | 24.75 | 23.62 | 34.30 | 30.02 | 27.42 | 25.53 | 24.00 | 22.69 | 34.05 | 30.03 | 27.60 | 25.83 | 24.34 | 23.03 |
| Ours-FOE | **33.68** | **29.59** | **27.45** | **26.03** | **24.90** | **23.97** | **34.33** | **30.09** | **27.50** | **25.62** | **24.12** | **22.96** | **34.08** | **30.09** | **27.72** | **26.05** | **24.67** | **23.52** |
| Improv. | 0.04 | 0.03 | 0.04 | 0.07 | 0.15 | 0.35 | 0.03 | 0.07 | 0.08 | 0.09 | 0.12 | 0.27 | 0.03 | 0.06 | 0.12 | 0.22 | 0.33 | 0.49 |
| $\sigma$(Lum, Chr) | 7,7 | 13,15 | 19,22 | 26,30 | 32,37 | 36,43 | 7,7 | 13,15 | 19,22 | 26,30 | 32,37 | 36,43 | 7,7 | 13,15 | 19,22 | 26,30 | 32,37 | 36,43 |
| $\sigma(\theta, \varphi)$ | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 | 1,0 | 2,1 | 3,2 | 5,3 | 7,5 | 9,7 |
| EPLL | **34.40** | **30.39** | 28.18 | 26.49 | 25.21 | 24,23 | 35,15 | 30,85 | 28,26 | 26,20 | 24,64 | 23.40 | **34.78** | 30.84 | 28.34 | 26.32 | 24.72 | 23.46 |
| Ours-EPLL | **34.40** | 30.38 | **28.20** | **26.56** | **25.41** | **24.61** | **35.16** | **30.87** | **28.33** | **26.35** | **24.97** | **23.94** | 34.77 | **30.86** | **28.43** | **26.54** | **25.15** | **24.11** |
| Improv. | 0.00 | -0.01 | 0.02 | 0.07 | 0.20 | 0.38 | 0.01 | 0.02 | 0.07 | 0.15 | 0.33 | 0.54 | -0.01 | 0.02 | 0.09 | 0.22 | 0.43 | 0.65 |

TABLE III
CPSNR RESULTS FOR THE TNRD CASE [19]. AVERAGE FOR EACH DATASET.

| | Kodak dataset | GM dataset | ARRI dataset |
|---|---|---|---|
| Noise | 50 | 50 | 50 |
| $\sigma(Color)$ | 50 | 50 | 50 |
| $\sigma(\theta, \varphi)$ | 5,5 | 5,5 | 5,5 |
| TNRD-RGB | 26.09 | 25.79 | 26.74 |
| Ours | **26.26** | **26.14** | **27.28** |
| Improv | 0.17 | 0.35 | 0.54 |

*1) Adaptability of our approach to CNN-like methods:* Recently, methods based on convolutional neural-networks (CNN) have become state-of-the-art in image denoising [18], [19]. To check the adaptability of our approach to these methods, we have used the data provided by Chen and Pock [19]. In their work, they consider standard deviation noise levels of $\sigma = 5, 15, 25,$ and 50 for gray-level images. We have run their method (called TNRD) for $\sigma = 50$ in each of the channels of the RGB color space, as it is the most similar space to the grey-level images they trained for. We have also run our preprocessing with $\sigma = 5$ in both angular components, followed by the running of the method for $\sigma = 50$ in each of the RGB channels. Results are presented in Table III. We can see that our preprocessing also improves the result of the original method. Let us note that in this particular case, the TNRD method does not provide better results than BM3D because we are not using an opponent-like color space (due to the training data considered).

## IV. CONCLUSIONS

We proposed a color decomposition framework to pre-process an image before the typical denoising pipeline is used. Our decomposition consists in obtaining a set of images in the spherical coordinate system, each of them considering the origin of the spherical transformation in a different color value. Once in the spherical coordinate system, we perform a mild denoising operation with some state-of-the-art method in the angular components and we come back to the sRGB space. Then, we apply the same state-of-the-art denoising method in this pre-processed image. Results show that our method consistently improves over the direct denoising of the input image by the same state-of-the-art method.

## References

[1] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Phys. D*, vol. 60, no. 1-4, pp. 259–268, Nov. 1992. [Online]. Available: http://dx.doi.org/10.1016/0167-2789(92)90242-F

[2] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 2 - Volume 02*, ser. CVPR ’05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 60–65. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.38

[3] S. P. Awate and R. T. Whitaker, “Higher-order image statistics for unsupervised, information-theoretic, adaptive, image filtering,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, 2005, pp. 44–51. [Online]. Available: https://doi.org/10.1109/CVPR.2005.176

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3d transform-domain collaborative filtering,” *IEEE TRANS. IMAGE PROCESS*, vol. 16, no. 8, p. 2080, 2007.

[5] M. Zontak, I. Mosseri, and M. Irani, “Separating signal from noise using patch recurrence across scales,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, 2013, pp. 1195–1202. [Online]. Available: https://doi.org/10.1109/CVPR.2013.158

[6] D. Zoran and Y. Weiss, “From learning models of natural image patches to whole image restoration,” in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV ’11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 479–486. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126278

[7] H. Talebi, X. Zhu, and P. Milanfar, “How to saif-ly boost denoising performance,” *IEEE Trans. Image Processing*, vol. 22, no. 4, pp. 1470–1485, 2013. [Online]. Available: https://doi.org/10.1109/TIP.2012.2231691

[8] H. Talebi and P. Milanfar, “Global image denoising,” *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 755–768, Feb 2014.

[9] M. Lebrun, M. Colom, A. Buades, and J. M. Morel, “Secrets of image denoising cuisine,” *Acta Numerica*, vol. 21, p. 475576, 2012.

[10] I. Omer and M. Werman, “Color lines: Image specific color representation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 946–953.

[11] I. Rafegas, J. Vazquez-Corral, R. Benavente, M. Vanrell, and S. Alvarez, “Enhancing spatio-chromatic representation with more-than-three color coding for image description,” *J. Opt. Soc. Am. A*, vol. 34, no. 5, pp. 827–837, May 2017. [Online]. Available: http://josaa.osa.org/abstract.cfm?URI=josaa-34-5-827

[12] E. Vázquez, R. Baldrich, J. van de Weijer, and M. Vanrell, “Describing reflectances for colour segmentation robust to shadows, highlights and textures,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 917–930, May 2011. [Online]. Available: http://www.cat.uab.cat/Public/Publications/2011/VBV2011

[13] “http://r0k.us/graphics/kodak/,” accessed 6th July 2017.

[14] J. Morovič, *Color gamut mapping*. Wiley, 2008, vol. 10.

[15] S. Andriani, H. Brendel, T. Seybold, and J. Goldstone, “Beyond the kodak image set: A new reference set of color image sequences,” in *2013 IEEE International Conference on Image Processing*, Sept 2013, pp. 2289–2293.

[16] G. Ghimpeteanu, D. Kane, T. Batard, S. Levine, and M. Bertalmío, “Local denoising based on curvature smoothing can visually outperform non-local methods on photographs with actual noise,” in *IEEE International Conference on Image Processing*, 2016.

[17] S. Roth and M. J. Black, “Fields of experts: A framework for learning image priors,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, Jun. 2005, pp. 860–867.

[18] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, July 2017.

[19] Y. Chen and T. Pock, “Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, June 2017.