Texture Sensitive Denoising for Single Sensor Color Imaging Devices

Angelo Bosco[1], Sebastiano Battiato[2], Arcangelo Bruna[1], and Rosetta Rizzo[2]

[1] STMicroelectronics, Stradale Primosole 50, 95121 Catania, Italy
[2] Università di Catania, Dipartimento di Matematica ed Informatica, Viale A. Doria 6, 95125 Catania, Italy
angelo.bosco@st.com, battiato@dmi.unict.it, arcangelo.bruna@st.com, rosetta.rizzo@dmi.unict.it

Abstract. This paper presents a spatial noise reduction technique designed to work on *CFA* (*Color Filter Array*) data acquired by *CCD/CMOS* image sensors. The overall processing preserves image details by using heuristics related to *HVS* (*Human Visual System*) and texture detection. The estimated amount of texture and HVS sensitivity are combined to regulate the filter strength. Experimental results confirm the effectiveness of the proposed technique.

Keywords: Denoising, Color Filter Array, HVS, Texture Detection.

1 Introduction

The image formation process through consumer imaging devices is intrinsically noisy. This is especially true using low-cost devices such as mobile-phones, PDA, etc., mainly in low-light conditions and absence of flash-gun.

In terms of denoising, linear filters can be used to remove Gaussian noise (*AWGN*), but they also significantly blur edge structures of an image. Many sophisticated techniques have been proposed to allow edge preserving noise removal such as: [12] and [13] that perform multiresolution analysis and processing in the wavelet domain, [3] that uses anisotropic non-linear diffusion equations but work iteratively, [1] and [10] that are spatial denoising approaches.

In this paper we propose a novel spatial noise reduction method that directly processes the raw CFA data, combining together *HVS* (Human Visual System) heuristics, texture/edges preservation techniques and sensor noise statistics, in order to obtain an effective adaptive denoising.

The proposed algorithm introduces the concept of the usage of *HVS* properties directly on the CFA raw data from the sensor to characterize or isolate unpleasant artifacts. The complexity of the proposed technique is kept low by using only spatial information and a small fixed-size filter processing window, allowing real-time performance on low cost imaging devices (e.g., mobile phones, PDAs, …).

The paper is structured as follows. In the next section some details about the CFA and *HVS* characteristics are briefly discussed; in Section 3 the overall details of the

A. Trémeau, R. Schettini, and S. Tominaga (Eds.): CCIW 2009, LNCS 5646, pp. 130–139, 2009.

proposed method are presented. An experimental section reports the results and some comparisons with other related techniques. The final section tracks directions for future works.

2 CFA Data and HVS Properties

In typical imaging devices a color filter is placed on top of the imager making each pixel sensitive to one color component only. A color reconstruction algorithm interpolates the missing information at each location and reconstructs the full *RGB* image. The color filter selects the red, green or blue component for each pixel; the most common arrangement is known as Bayer pattern [4].

In the Bayer pattern the number of green elements is twice the number of red and blue pixels due to the higher sensitivity of the human eye to the green light, which, in fact, has a higher weight when computing the luminance.

The *HVS* properties are a complex phenomenon (highly nonlinear) not yet completely understood involving a lot of complex parameters. It is well known that the *HVS* has a different sensitivity at different spatial frequencies [15]. In areas containing mean frequencies the eye has a higher sensitivity. Furthermore, chrominance sensitivity is weaker than the luminance one.

The *HVS* response does not entirely depend on the luminance value itself, rather, it depends on the luminance local variations with respect to the background; this effect is described by the Weber-Fechner's law [7].

These properties of the *HVS* have been used as a starting point to devise a CFA filtering algorithm, that providing the best performance if executed as the first algorithm of the *IGP* (*Image Generation Pipeline)* [2]. Luminance from CFA data can be extracted as explained in [11], but for our purposes it can be roughly approximated by the green channel values before gamma correction.

3 Algorithm

3.1 Overall Filter Block Diagram

A block diagram describing the overall filtering process is illustrated in Fig. 1. Each block will be separately described in detail in the following sections.

The fundamental blocks of the algorithm are:

- ***Signal Analyzer Block*:** computes a filter parameter incorporating the effects of human visual system response and signal intensity in the filter mask. (Section 3.2 for further details)
- ***Texture Degree Analyzer*:** determines the amount of texture in the filter mask using information from the *Signal Analyzer Block*. (Section 3.4)
- ***Noise Level Estimator*:** estimates the noise level in the filter mask taking into account the texture degree. (Section 3.5)
- ***Similarity Thresholds Block*:** computes the thresholds that are used to determine the weighting coefficients for the neighborhood of the central pixel.

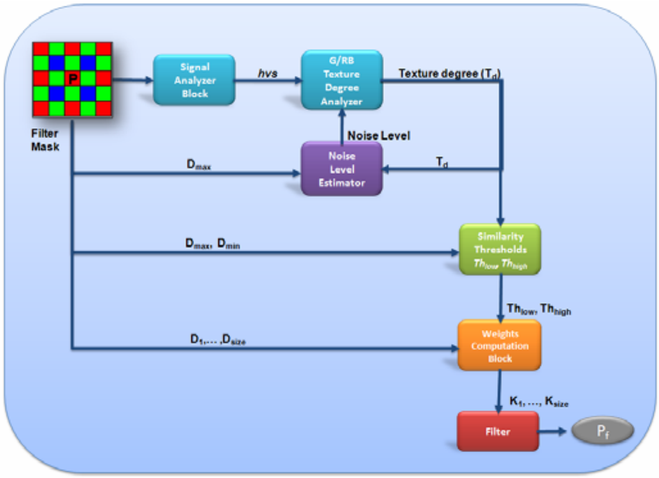


Fig. 1. Overall Filter Block Diagram

- ***Weights Computation Block***: uses the thresholds computed by the *Similarity Thresholds Block* and assigns a weight to each neighborhood pixel, representing the degree of similarity between pixel pairs. (Section 3.6)
- ***Filter Block***: actually computes the final weighted average generating the final filtered value. (Section 3.7)

3.2 Signal Analyzer Block

As noted in [5] and [8] it is possible to approximate the minimum intensity gap that is necessary for the eye to perceive a change in pixel values. This phenomenon is known as luminance masking or light adaptation. Higher gap in intensity is needed to perceive a visual difference in very dark areas, whereas for mid and high pixel intensities a small difference in value between adjacent pixels is more easily perceived by the eye [8].

It is also crucial to observe that in data from real image sensors, the constant *AWGN* model does not fit well the noise distribution for all pixel values. In particular, as discussed in [6], the noise level in raw data is predominantly signal-dependent and increases as the signal intensity raises; hence, the noise level is higher in very bright areas.

We decided to incorporate the above considerations of luminance masking and sensor noise statistics into a single curve as shown in Fig. 2. The shape of this curve allows compensating for lower eye sensitivity and increased noise power in the proper areas of the image, allowing adaptive filter strength in relation to the pixel values.

A high *HVS* value (HVS_{max}) is set for both low and high pixel values: in dark areas the human eye is less sensitive to variations of pixel intensities, whereas in bright areas noise standard deviation is higher. *HVS* value is set low (HVS_{min}) at mid pixel intensities.

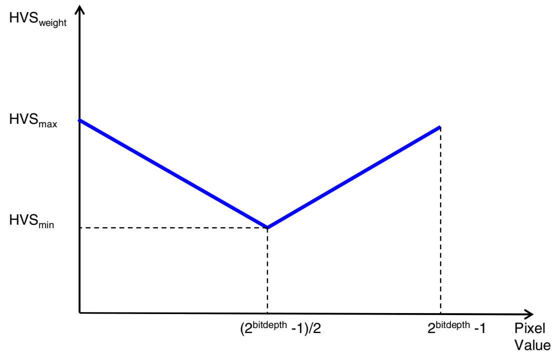


Fig. 2. HVS curve used in the proposed approach

The *HVS* coefficient computed by this block will be used by the Texture Degree Analyzer that outputs a degree of texture taking also into account the above considerations (Section 3.4). As stated in Section 2, in order to make some simplifying assumptions, we use the same *HVS* curve for all CFA colour channels taking as input the pixel intensities directly from the sensor.

3.3 Filter Masks

The proposed filter uses different filter masks for green and red/blue pixels to match the particular arrangement of pixels in the CFA array. The size of the filter mask depends on the resolution of the imager: at higher resolution a small processing window might be unable to capture significant details. For our processing purposes a 5x5 window size provided a good trade-off between hardware cost and image quality. Typical Bayer processing windows are illustrated in Fig. 3.

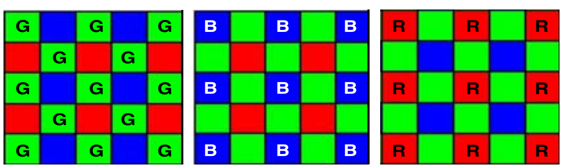


Fig. 3. Filter Masks for Bayer Pattern Data

3.4 Texture Degree Analyzer

The texture analyzer block computes a reference value T_d that is representative of the local texture degree. This reference value approaches 1 as the local area becomes increasingly flat and decreases as the texture degree increases (Fig. 4). The computed coefficient is used to regulate the filter strength so that high values of T_d correspond to flat image areas in which the filter strength can be increased.

Depending on the color of the pixel under processing, either green or red/blue, two different texture analyzers are used. The red/blue filter power is increased by slightly modifying the texture analyzer making it less sensitive to small pixel differences (Fig. 5).

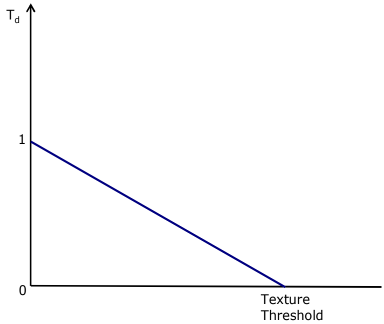


Fig. 4. Green Texture Analyzer

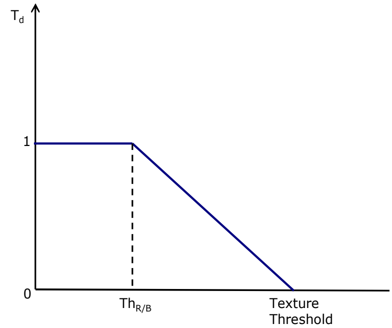


Fig. 5. Red/Blue texture analyzer

The texture analyzer block output depends on a combination of the maximum difference between the central pixel and the neighborhood D_{max} and *TextureThreshold*, a value that is obtained by combining information from the *HVS* response and noise level, as described below (2). The green and red/blue texture analyzers are defined as follows:

$$T_d(green) = \begin{cases} 1 & D_{Max} = 0 \\ -\dfrac{D_{\max}}{TextureThreshold} + 1 & 0 < D_{Max} \leq TextureThreshold \\ 0 & D_{Max} > TextureThreshold \end{cases} \tag{1}$$

$$T_d(red/blue) = \begin{cases} 1 & D_{\max} \leq Th_{R/B} \\ -\dfrac{(D_{\max} - Th_{R/B})}{(TextureThreshold - Th_{R/B})} + 1 & Th_{R/B} < D_{\max} \leq TextureThreshold \\ 0 & D_{\max} > TextureThreshold \end{cases}$$

hence:

- if $T_d = 1$ the area is assumed to be completely flat;
- if $0 < T_d < 1$ the area contains a variable amount of texture;
- if $T_d = 0$, the area is considered to be highly textured.

The texture threshold for the current pixel, belonging to Bayer channel c ($c=R,G,B$), is computed by adding the noise level estimation to the *HVS* response (2):

$$TextureThreshold_c(k) = HVS_{weight}(k) + NL_c(k-1) \tag{2}$$

where NL_c denotes the noise level estimation on the previous pixel of the same Bayer color channel c(see Section 3.4) and HVS_{weight} (Fig. 2) can be interpreted as a *jnd* (*just noticeable difference*); hence an area is no longer flat if the D_{max} value exceeds the *jnd* plus the local noise level *NL*.

The green texture analyzer (Fig. 4) uses a stronger rule for detecting flat areas, whereas the red/blue texture analyzer (Fig. 5) detects more flat areas being less sensitive to small pixel differences below the $Th_{R/B}$ threshold.

3.5 Noise Level Estimator

In order to adapt the filter strength to the local characteristics of the image, a noise level estimation is required. The proposed noise estimation solution is pixel based and is implemented taking into account the previous estimation to calculate the current one. The noise estimation equation is designed so that:

i) if the local area is completely flat ($T_d = 1$) , then the noise level is set to D_{max};
ii) if the local area is highly textured ($T_d = 0$), the noise estimation is kept equal to the previous region (i.e., pixel);
iii) otherwise a new value is estimated.

Each color channel has its own noise characteristics hence noise levels are tracked separately for each color channel. The noise level for each channel c (c=R,G,B) is estimated according to the following formulas:

$$NL_c(k) = T_d(k) * D_{\max}(k) + [1 - T_d(k)] * NL_c(k-1) \tag{3}$$

where $T_d(k)$ represents the texture degree at the current pixel and $NL_c(k\text{-}1)$ is the previous noise level estimation, evaluated considering pixel of the same colour, already processed. These equations satisfy requirements i), ii) and iii).

3.6 Weighting Coefficients

The final step of the filtering process consists in determining the weighting coefficients W_i to be assigned to the neighboring pixels of the filter mask. The absolute differences D_i between the central pixel and its neighborhood must be analyzed in combination with the local information (noise level, texture degree and pixel intensities) for estimating the degree of similarity between pixel pairs (Fig. 6).

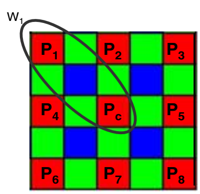


Fig. 6. Wi coefficients weight the similarity degree between the Pc and its neighborhood

As stated in Section 2, if the central pixel P_c belongs to a textured area, then only small pixel differences must be filtered. The lower degree of filtering in textured areas allows maintaining the local sharpness, removing only pixel differences that are not perceived by the *HVS*.

Let:

– P_c be the central pixel of the working window;
– P_i, i = 0,…,7, be the neighborhood pixels;
– D_i = abs(P_c -P_i), i=0,…,7 the set of absolute differences between the central pixel and its neighborhood;

In order to obtain the W_i coefficients, each absolute difference D_i must be compared against two thresholds Th_{low} and Th_{high} that determine if, in relation to the local information, the i-th difference D_i is:

- small enough to be heavily filtered,
- big enough to remain untouched,
- an intermediate value to be properly filtered.

To determine which of the above cases is valid for the current local area, the local texture degree is the key parameter to analyze. It is important to remember at this point that, by construction, the texture degree coefficient (T_d) incorporates the concepts of dark/bright and noise level; hence, its value is crucial to determine the similarity thresholds to be used for determining the W_i coefficients. In particular, the similarity thresholds are computed according to the following rules:

1. if the local area is flat both thresholds (Th_{low} , Th_{high}) are set to D_{max}, which means that all neighborhood pixels whose difference from the central pixel is less than D_{max} have maximum weight.
2. if the local area is fully textured then Th_{low} is set to D_{min} and Th_{high} is set to the average point between D_{min} and D_{max}, meaning that only pixels whose difference from the central pixel is very small have the maximum weight.
3. if the local area has a medium degree of texture T_d ($0 < T_d < 1$), the situation is as depicted in Fig. 7, where the similarity weight progressively decreases as the i-th difference increases.

Once the similarity thresholds have been fixed, it is possible to finally determine the filter weights by comparing the D_i differences against them (Fig. 7).

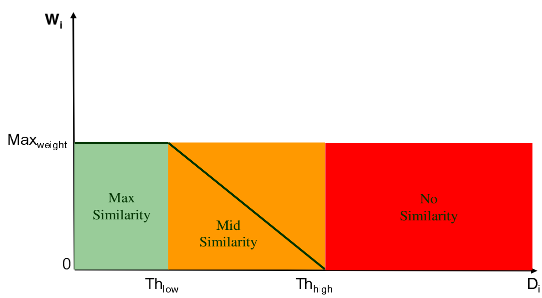


Fig. 7. Weights assignment. The i-th weight denotes the degree of similarity between the central pixel in the filter mask and the i-th pixel in the neighborhood.

3.7 Final Weighted Average

Let $W_0,\ldots,W_N$ (N: number of neighborhood pixels) be the set of weights computed for the each neighboring element of the central pixel P_c. The final filtered value P_f is obtained by a weighted average as follows:

$$P_f = \frac{1}{N}\sum_{i=0}^{N}\left[W_i P_i + (1 - W_i)P_c\right] \tag{4}$$

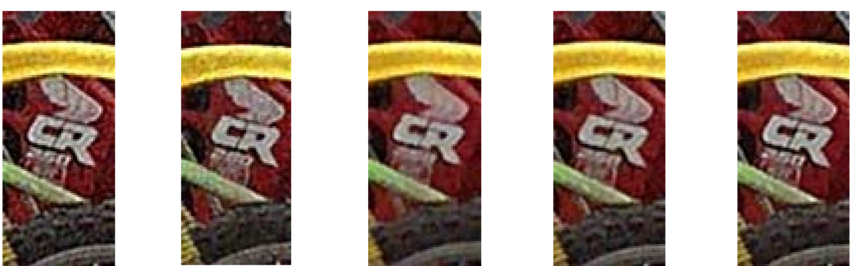
4 Experimental Results

In order to assess the visual quality of the proposed method, we have compared it with the *SUSAN* (*Smallest Univalue Segment Assimilating Nucleus*) [14] and multi-stage median filters [9] classical noise reduction algorithm. This choice is motivated by considering the comparable complexity of these solutions. Though more complex recent methods for denoising image data achieve very good results, they are not yet suitable for real-time implementation.

The test noisy image in Fig. 8 was obtained adding noise with mean standard deviation σ=10. Fig. 9 (b),(c),(d) show filtered results respectively with SUSAN, Multi-stage median-1, Multistage median-3 and proposed technique of the cropped and zoomed detail of Fig. 8, showed in Fig. 9(a). To perform the test, all the input images were bayerized before processing.

Fig. 10 shows how the proposed method performs well in terms of PSNR compared to the other algorithms used in the test over the 24 Standard Kodak Images

Fig. 8. Noisy image (PSNR 32.8 dB)



(a) Cropped and zoomed noisy image (PSNR 32.8.1 dB)

(b) SUSAN (PSNR 32.5 dB)

(c) Multistage median -1 filter. (PSNR 32.9 dB)

(d) Multistage median -3 filter. (PSNR 29.8 dB)

(e) Proposed method. (PSNR 33.8 dB)

Fig. 9. (a) Cropped and zoomed noisy image in Fig.8. (b) SUSAN. (c) Multistage median-1 filter. (c) Multistage median-3 filter. (e) Proposed method.

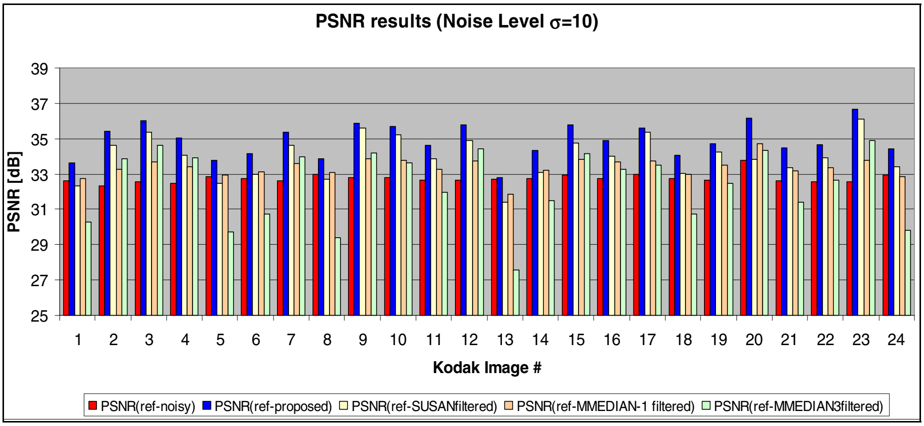


Fig. 10. PSNR of the Standard Kodak Images test set with standard deviation

5 Conclusions and Future Works

A spatial adaptive denoising algorithm has been presented; the method exploits characteristics of the human visual system and sensor noise statistics in order to achieve pleasant results in terms of perceived image quality. The noise level and texture degree are computed to adapt the filter behaviour to the local characteristics of the image. The algorithm is suitable for real time processing of images acquired in CFA format since it requires simple operations and divisions that can also be implemented via lookup tables.

Future works include the extension of the processing masks along with the study and integration of other *HVS* characteristics.

References

1. Amer, A., Dubois, E.: Fast and reliable structure-oriented video noise estimation. IEEE Transaction on Circuits System Video Technology 15(1) (2005)
2. Battiato, S., Mancuso, M.: An Introduction to the Digital Still Camera Technology. ST Journal of System Research, Special Issue on Image Processing for Digital Still Camera 2, 2–9 (2001)
3. Barcelos, C.A.Z., Boaventura, M., Silva, E.C.: A Well-Balanced Flow Equation for Noise Removal and Edge Detection. IEEE Transactions on Image Processing 12(7), 751–763 (2003)
4. Bayer, B.E.: Color Imaging Array, US. Patent No. 3, 971, 965 (1976)
5. Chou, C.-H., Li, Y.-C.: A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. IEEE Transactions on Circuits and Systems for Video Technology 5(6), 467–476 (1995)
6. Foi, A., Trimeche, M., Katkovnik, V., Egiazarian, K.: Practical Poissonian-Gaussian Noise Modeling and Fitting for Single-Image Raw-Data. IEEE Transactions on Image Processing 17(10), 1737–1754 (2008)

7. Gonzales, R., Woods, R.: Digital Image Processing, 3rd edn. Prentice Hall, Englewood Cliffs (2007)
8. Hontsch, I., Karam, L.J.: Locally adaptive perceptual image coding. IEEE Transactions on Image Processing 9(9), 1472–1483 (2000)
9. Kalevo, O., Rantanen, H.: Noise Reduction Techniques for Bayer-Matrix Images. In: Proceedings of SPIE Electronic Imaging, Sensors, Cameras, and Applications for Digital Photography III 2002, San Jose, CA, USA, vol. 4669 (2002)
10. Kim, Y.-H., Lee, J.: Image feature and noise detection based on statistical hypothesis tests and their applications in noise reduction. IEEE Transactions on Consumer Electronics 51(4), 1367–1378 (2005)
11. Lian, N., Chang, L., Tan, Y.-P.: Improved color filter array demosaicking by accurate luminance estimation. In: IEEE International Conference on Image Processing, vol. 1, pp. 41–44 (2005)
12. Portilla, J., Strela, V., Wainwright, M.J., Simoncelli, E.P.: Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain. IEEE Transactions on Image Processing 12(11), 1338–1351 (2003)
13. Scharcanski, J., Jung, C.R., Clarke, R.T.: Adaptive Image Denoising Using Scale and Space Consistency. IEEE Transactions on Image Processing 11(9), 1092–1101 (2002)
14. Smith, S.M., Brady, J.M.: SUSAN - A New Approach to Low Level Image Processing. International Journal of Computer Vision 23(1), 45–78 (1997)
15. Wandell, B.: Foundations of Vision, Sinauer Associates (1995)