Cost-effective multiframe demosaicking based on bilateral filtering

K. Gorokhovskiy

Aerocosmos Scientif c Center for Aerospace Monitoring, Moscow, Russia
gorokhovskiy@gmail.com

Abstract

A simple but effective multiframe demosaicking method is proposed. Its primary goal is to replace more expensive mechanical motion compensation systems. Therefore, it is designed to be easily implemented in hardware for consumer devices. The described multiframe demosaicking algorithm is suitable for mass production devices such as mobile phones or digital cameras. It is compared to a multiframe noise reduction of similar complexity. The comparison is based on computer-based simulation of a camera being (unintentionally) shaken by a human operator. The following error measurements were taken: Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR) and Normalized Color Difference (NCD) errors measurements were taken.

Keywords: *demosaicking, noise reduction, bilateral filtering, multiframe processing*

1. INTRODUCTION

Digital cameras and so-called camera-phones are now widely spread. Although, image quality from them has improved drastically in recent years, still, it is not comparable to human vision capabilities especially under low light conditions. One of the main problems is sensor noise.

Current cameras perform at their physical limits and photon noise is dominant. On a physical level, this type of noise can be reduced by increasing the number of photons detected by each cell on a sensor. Usually, the solutions are: increasing the optical efficiency of a lens system or increasing exposure times.

Improving optical efficiency is expensive as the complexity of the lens grows disproportionately relative to its quality, not to mention that the camera often needs to be small in its application (e.g. a camera-phone). Longer exposures, in turn, produce motion blur which can be compensated mechanically or electronically.

Taking into account the generally falling cost of electronic components electronic motion compensation becomes more and more attractive in terms of quality per unit cost.

Both frame-based demosaicking and multiframe noise reduction are well developed areas in their own right. The combination of these two methods only recently received a proper attention [1]. However, there is still a lack of simple but effective methods which can be implemented in existing devices.

In this paper a new method of multiframe demosaicking is proposed and compared to combination of simple frame-based demosaicking and multi-frame noise reduction. The comparison is carried out using computer-based simulation of a series of shots which are shifted and rotated, then mosaicked. After that, Poisson noise is added to simulate the photon noise of a photo sensor. This algorithm is an extension of the work described in [2].

2. ALGORITHM REQUIREMENTS

The original prerequisite for the proposed algorithm is that it can be put into a camera image processing pipeline without a significant increase in cost. This leads to the following requirements:

(a) The method should not consume too much memory (not more than 4 image frames) even if the technique involves merging many more frames.

(b) It should be real-time or, in other words, the user should receive the result just after the shot (no time-consuming post processing is allowed).

It is clear from the requirements that algorithm should be stream based and data should be accumulated and processed "on the fly".

Having many images of the same scene it is possible to use a wide variety of super-resolution algorithms. However, the requirements for memory and computational power restrict application of those methods inside digital still cameras and mobile phones.

3. TEMPORAL BILATERAL DEMOSAICKING

Having the classical bilateral filtering equation for image $f(\mathbf{x})$ as in [3]:

$$h(\mathbf{x}) = k_d^{-1} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(\xi) c(\xi - \mathbf{x}) s(f(\xi) - f(\mathbf{x})) d\xi \qquad (1)$$

and normalization coefficient k_d :

$$k_d = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} c(\xi - \mathbf{x}) s(f(\xi) - f(\mathbf{x})) d\xi \qquad (2)$$

where $c(\xi - \mathbf{x})$ is the *geometric* closeness between the neighborhood centre $\mathbf{x}$ and a nearby point ξ, $s(f(\xi) - f(\mathbf{x}))$ measures the *photometric* similarity between the pixel at the neighborhood centre $\mathbf{x}$ and that of a nearby point ξ.

For the task of multiframe demosaicking it is possible to introduce an additional pixel weight coefficient responsible for trustworthiness of a pixel $w(\xi)$. In the situation when several frames are merged together some pixels may contain more pixels of a particular color. The greater the number of values in a given pixel position, the better the accuracy. Thus equations (1) and (2) become:

$$h(\mathbf{x}) = k_d^{-1} \sum_{t=1}^{T} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{f}(\xi) w(\xi) c(\xi - \mathbf{x}) s(f(\xi) - f(\mathbf{x})) d\xi \qquad (3)$$

and

$$k_d = \sum_{t=1}^{T} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} w(\xi) c(\xi - \mathbf{x}) s(f(\xi) - f(\mathbf{x})) d\xi \qquad (4)$$

where $\hat{f}(\xi)$ are the resultant mean values of the colors in the given locations, t is the index of the frame in a sequence, T is total number of frames available for fusion . By converting the equation in discrete space we have

$$h(x,y)=k_d^{-1}\sum_{t=1}^{T}\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}\hat{f}(i,j)w(i,j)c(i-x,j-y)s(f(i,j)-f(x,y)) \tag{5}$$

and

$$k_d=\sum_{t=1}^{T}\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}w(i,j)c(i-x,j-y)s(f(i,j)-f(x,y)) \tag{6}$$

Usually the *photometric* similarity function is defined as:

$$s(f(i,j)-f(x,y))=\exp\left\{-\left(\frac{\delta(f(i,j)-f(x,y))}{2\sigma}\right)^2\right\} \tag{7}$$

where

$$\delta(f(i,j)-f(x,y))=\|f(i,j)-f(x,y)\| \tag{8}$$

Usually, $\|f(i,j)-f(x,y)\|$ is selected as a suitable measure of distance between the two color values. In the scalar case, this may be simply the absolute difference of the pixel values or, since the photon noise increases with intensity, an intensity dependant version of it. It is possible to determine what the *photometric* similarity function should be in case of Poisson noise.

In order to simplify the formulae and minimize the amount of computations the image samples can be converted to a space where the noise has a normal distribution (i.e. Gaussian). Also, it is assumed for simplicity that image under consideration has only one channel. The equations below can be easily extended for multi-channel images. The probability function relating to the difference of two image samples which are close in space is:

$$p(a-b)=\frac{1}{\sqrt{2\pi(\sigma_a^2+\sigma_b^2)}}\exp\left\{-\frac{((a-b)-(\mu_a-\mu_b))^2}{2(\sigma_a^2+\sigma_b^2)}\right\} \tag{9}$$

where σ^2 is variance of a random variable and μ is the mean or its expected value. The probability that these two samples have the same value is

$$p(a-b)\big|_{\mu_a=\mu_b}=\frac{1}{\sqrt{2\pi(\sigma_a^2+\sigma_b^2)}}\exp\left\{-\frac{(a-b)^2}{2(\sigma_a^2+\sigma_b^2)}\right\} \tag{10}$$

It can also be demonstrated that the standard deviation of both variables are the same

$$\sigma_a=\sigma_b=\sigma \tag{11}$$

Thus

$$p=\frac{1}{\sqrt{4\pi\sigma^2}}\exp\left\{-\frac{(a-b)^2}{4\sigma^2}\right\} \tag{12}$$

which is similar to equation (7). For the experiment the following conversion to the space with a normal distribution was used

$$f_n(x,y)=\sqrt{f_p(x,y)} \tag{13}$$

where f_p are the samples with a Poisson distribution and f_n are the samples with a normal distribution.

In order to reduce the amount of computations used a simplified spatial penalty function was used:

$$c(i,j)=\begin{cases}1, & -N\le i\le N,-M\le j\le M\\ 0, & otherwise\end{cases} \tag{14}$$

The multiframe bilateral demosaicking algorithm was applied on a 5×5 neighborhood with $N=M=2$.

The described equations can be optimized in order to minimize memory usage on a computational device. Equations (5) and (6) allow accumulation of the intermediate results frame buy frame.

By storing numerator and denominator k_d of equation (5) as two separate frames in memory it is possible accumulate data frame by frame. When no more frames are expected in a sequence the final result of computation can be achieved by dividing accumulated numerator

$$\sum_{t=1}^{T}\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}\hat{f}(i,j)w(i,j)c(i-x,j-y)s(f(i,j)-f(x,y))$$

by denominator k_d from equation (6).

4. GLOBAL MOTION ESTIMATION ON MOSAICKED IMAGES

As the global motion estimation was not an essential part of the comparison and the simplest exhaustive search was taken as a basis.

It is important to stress out that there was no novelty introduced for global motion estimation in this paper. Motion estimation was not a goal of this research. Any state of the art research results on global motion estimation could be used here. Therefore the comparison of accuracy of used global motion estimation (ME) algorithm was outside of the scope of this paper.

The same global motion estimation coefficients were used for both compared methods and therefore the *relative* accuracy of demosaicking methods should be unaffected by the accuracy of the global motion estimation. However, the algorithm for motion estimation is explained below for the reproducibility of the results.

It was assumed that global motion of the frame can be described as an affine transform with a relatively small number of coefficients so that for small area of image it can simply be defined as a shift in two dimensions (see Figure 1).

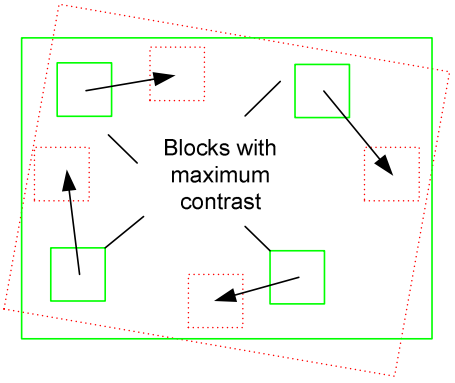

Figure 1: Global motion estimation using block matching. The proposed model assumes that if rotation is small (0° – 2°) it can be neglected for motion blocks (32x32 pixels). Only shifts are taken into account.

Equation (15) and condition (16) define this:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (15)$$

where

$$|a_{21}| << 1, |a_{12}| << 1 \quad (16)$$

A limited number of blocks with maximal contrast were selected. The target is to find shifts in these blocks and to calculate the global motion using linear regression [4] or robust fitting. Figure 1 illustrates this.

The exhaustive search block matching algorithm was adapted from [5]. It was modified to introduce a penalty term for large motion vectors. In cases where two vectors exist with an equal cost the shortest will be selected.

The precision of motion estimation can be optimized further with a priori knowledge that the transform contains only rotation and shift. This condition can be described as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & a_{13} \\ -\sin\alpha & \cos\alpha & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (17)$$

As the rotation is small, the $\cos\alpha$ component can be replaced by 1 . Let c be:

$$c = \sin\alpha \quad (18)$$

Hence:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & c & a_{13} \\ -c & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (19)$$

The number of transform coefficients is thereby reduced from 6 to 3 making linear regression methods more effective.

The difference between classical motion estimation techniques and the proposed approach is that it is required to work with mosaicked images. The mosaicked images can be demosaicked before motion estimation but this is not the most precise or most computationally efficient way. The proposed method uses mosaicked (RAW) images for motion estimation. It will be shown that it is possible to obtain pixel precise motion estimate vectors using mosaicked data.

The basic operation in motion estimation is a measure of similarity between two regions of images. In our case the mask of existence of the given color in a given position is available, which simplifies the task. Let us assume that the penalty is the absolute difference between the two colors at a given pixel. Then it is possible to say that having no particular color in the mask should not add a penalty. This can be formalized as follows

$$p(x_1, y_1, x_2, y_2) = \\ = \sum_{c=1}^{C} |s_1(x_1, y_1, c) m_2(x_2, y_2, c) - s_2(x_2, y_2, c) m_1(x_1, y_1, c)| \quad (20)$$

where C is number of colors used in a sensor (usually 3 or 4), $p(x_1, y_1, x_2, y_2)$ is the penalty term for pixels in locations x_1, y_1 and x_2, y_2 respectively. Also, s_1 denotes a sample of the first image, m_1 is a mask value for the first image, s_2 and m_2 are defined similarly for the second image.

For some offsets there will be situations when all pixels between two block of image are unmatched according to color masks. In such conditions the penalty will be zero no matter what are the contents of the image.

In order to avoid such conditions the values of pixels are blurred spatially (separately by color planes) together with corresponding color masks.

The experiments carried out by the author show that Gaussian filtering with a small 3×3 kernel of one of the images (including mask) improves the accuracy of the motion estimation for Bayer pattern.

Then, the penalty or difference measure $P(x_1, y_1, x_2, y_2)$ for the block of pixels with dimensions N and M will be

$$P(x_1, y_1, x_2, y_2) = \sum_{i=1}^{N} \sum_{j=1}^{M} p(x_1 + i, y_1 + j, x_2 + i, y_2 + j) \quad (21)$$

Square blocks of pixels with dimensions $M = N = 32$ were used for in this research.

The resultant motion estimation algorithm used in this research can be described as following steps:

1. Split the first frame in the sequence into blocks 32 by 32
2. Select 50% of these blocks with maximal contrast
3. Let us assign the index k for each block having $k \in [1 \ldots K]$, where K is total number of selected blocks
4. Store *integer* coordinates of the centers of the blocks as (x_k, y_k)
5. For each block with coordinates of the (x_k, y_k) find the corresponding block on a given frame (different from first one) with *integer* coordinates (x'_k, y'_k) which minimizes $P(x_k, y_k, x'_k, y'_k)$
6. Using multivariate linear regression algorithm [4] on initial and resultant sets of coordinates (x_k, y_k) and (x'_k, y'_k) find coefficients c, a_{13}, and a_{23} for this frame

5. COMPARRISON OF THE RESULTS USING SIMULATION OF NOISE AND SHAKE

There methods were compared using raw images generated from "Kodak Image Set" [6]. Images were downscaled in order to reduce simulation time. The aim was to reproduce the image sequence from the real camera. Using real image sequence it is difficult to evaluate the accuracy of described methods as it is impossible to get the original reference image. By contrast when using a simulation the reference image is known in advance.

The following assumptions were used for the simulation process:

1. Overall exposure time of a sequence of shots is less than ¼ second
2. There is only rotation and shift of the image taking place (no scaling or second order distortions)
3. Rotation is no more than 5 degrees between any two images in a sequence

(a) (b) (c) (d)

Figure 2: From left to right: (a) original image, (b) one of the noisy images form the sequence of 9 images, (c) result of temporal variable number of gradients demosaicking using 9 images, (d) result of proposed method using 9 images.

For error measurements MSE, PSNR, NCD formulae were used. MSE and PSNR are:

$$MSE = \frac{1}{W \cdot H \cdot C} \sum_{y=1}^{W} \sum_{x=1}^{H} \sum_{c=1}^{3} \left\| O(x,y,c) - R(x,y,c) \right\|^2 \qquad (22)$$

$$PSNR = 10 \cdot \log_{10}\left(\frac{1}{MSE}\right) \qquad (23)$$

It is assumed that color values are within the range $[0,1]$.

NCD stands for Normalized Color Difference. It was used previously to quantify the perceptual color difference and is defined as follows:

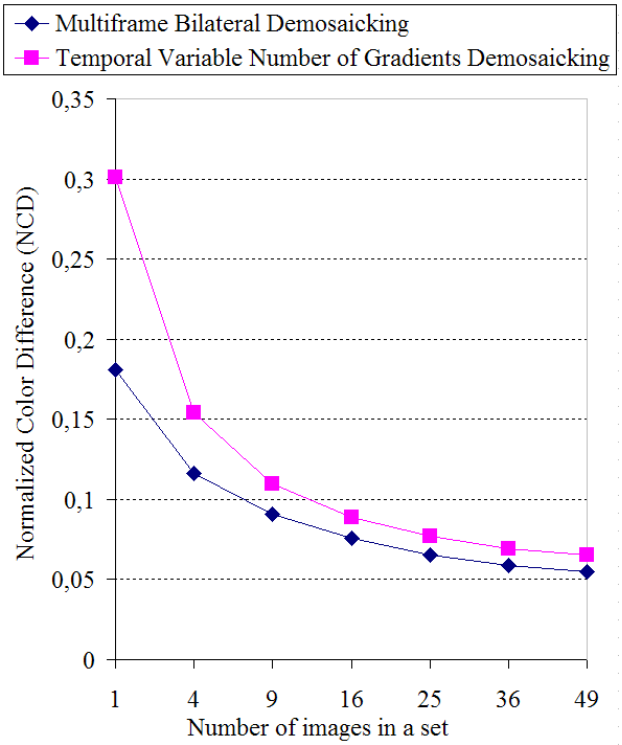

Figure 3: Dependence of normalized color difference from number of images available for fusing.

$$NCD = \frac{\sum_{x,y} \sqrt{(L_o - L_r)^2 + (u_o - u_r)^2 + (v_o - v_r)^2}}{\sum_{x,y} \sqrt{L_o^{\,2} + u_o^{\,2} + v_o^{\,2}}} \qquad (24)$$

where L_r, u_r, v_r are lightness and chrominance components of the result image in CIELUV color space [7] at the pixel's location (x,y), L_o, u_o, v_o are the same color components that were in the original image.

The simulation was performed for different number of frames in a sequence varying from 1 to 49. The results are shown in Table I. The Multiframe Bilateral Demosaicking shows best results for MSE, PSNR and NCD measures.

Variable Number of Gradients Demosaicking [8] is one of the best non-iterative algorithms described in scientific publications. The operational neighborhood for both these methods is 5×5 pixels. However, the computational efficiency of Multiframe Temporal Demosaicking is better than for Temporal Variable Number of Gradients Demosaicking.

Reference, noisy, and processed images are shown in Figure 2.

TABLE I: ACCURACY OF THE RESULTS OF TEMPORAL VARIABLE NUMBER OF GRADIENTS DEMOSAICKING COMPARED TO MULTIFRAME BILATERAL DEMOSAICKING ON THE KODAK IMAGE SET USING MSE, PSNR, AND NCD ERROR MEASUREMENTS

Number of images in a sequence	Temporal Variable Number of Gradients			Multiframe Bilateral Demosaicking		
	MSE	PSNR	NCD	MSE	PSNR	NCD
1	0.00966	20.22	0.301	0.00414	24.19	0.181
4	0.00262	25.90	0.154	0.00236	26.57	0.116
9	0.00151	28.38	0.110	0.00151	28.50	0.091
16	0.00111	29.80	0.089	0.00108	29.97	0.076
25	0.00091	30.76	0.077	0.00079	31.29	0.065
36	0.00083	31.27	0.069	0.00066	32.12	0.059
49	0.00077	31.67	0.065	0.00057	32.81	0.055

As can be seen from Table 2 and also from Figure 2, multiframe demosaicking is more effective than temporal noise reduction for a small number of frames. As the number of frames increases these two methods show comparable performance. On the other hand, multiframe temporal demosaicking is at least marginally better for each case in the simulation.

As can also be seen from the graph in Figure 3 both methods become more effective as the number of frames in the set increases.

It is important to note that the proposed method of Multiframe Bilateral Demosaicking is not based on the specific structure of a classic Bayer filter layout and can be easily adapted for alternative filter patterns.

6. CONCLUSION

The proposed method of multiframe demosaicking has shown an advantage over temporal noise reduction on sequences with number of frames varying from 1 to 49. It is also simple to implement in the hardware of modern digital camera or a mobile phone. To get better results with a small number of images in a set, multiframe demosaicking can be improved in an adaptive way such as a spatial filtering kernel for uniform surfaces and temporal filtering for edges. It should be mentioned that one of the disadvantages of the proposed method is absence of local motion estimation. The method can be significantly improved by detecting the areas of local motion between the frames. Thus, by matching the moved areas it would be possible to reduce the noise without introducing motion blur.

7. REFERENCES

[1] S. Farsiu, M. Elad, and P. Milanfar, "Multiframe Demosaicking and Super-Resolution of Colour Images", *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 141-159, Jan. 2006.

[2] K. Gorokhovskiy; J.A. Flint, S. Datta, and N. Glushnev, "Cost Effective Multiframe Demosaicking for Noise Reduction," *15th International Conference on Digital Signal Processing*, Cardiff, UK, pp. 407–410, July 2007.

[3] C. Tomasi, R. Manduchi, "Bilateral Filtering for Gray and Colour Images," *Proceedings of the 1998 IEEE International Conference on Computer Vision*, Bombay, India, 1988

[4] P.W. Holland, R.E. Welsch, "Robust Regression Using Iteratively Reweighted Least-Squares," *Communications in Statistics: Theory and Methods*, A6, pp. 813 827, 1977.

[5] A. Barjatya, "Matching Algorithms for Motion Estimation," 2004, http://www.mathworks.com/ matlabcentral/fileexchange/loadFile.do?objectId=8761, last accessed December 2007.

[6] 24 scanned images, "Eastman Kodak © photographic color image database", 1993

[7] M.D. Fairchild, "Color Appearance Models," *Wiley–IS&T series in imaging science and technology*, Chichester, West Sussex, England: J. Wiley, pp. 194–201, 2005.

[8] C E. Chang, C. Shiufun, and D. Pan, "Color filter array recovery using a threshold-based variable number of gradients," *Proceedings of SPIE*, vol. 3650, pp. 36–43, 1999.