Cost effective multiframe demosaicking for noise reduction

This item was submitted to Loughborough University’s Institutional Repository by the/an author.


Additional Information:

- This is a conference paper [© IEEE]. It is also available at: http://ieeexplore.ieee.org/ Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Metadata Record: https://dspace.lboro.ac.uk/2134/6169

Version: Published

Publisher: © IEEE

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
COST EFFECTIVE MULTIFRAME DEMOSAICKING FOR NOISE REDUCTION

K. Gorokhovskiy, J.A. Flint, S. Datta, and N. Glushnev

Department of Electronic and Electrical Engineering, Loughborough University, Loughborough, LE11 3TU, UK
kgorokhovsky@apical-imaging.com, J.A.Flint@bboro.ac.uk, S.Datta@bboro.ac.uk;

Apical Limited, Loughborough Technology Centre, Epinal Way, Loughborough, LE11 3GE, UK
Nikolay.Glushnev@microsoft.com

ABSTRACT
A simple but effective multiframe demosaicking method is proposed. It is compared to a multiframe noise reduction of similar complexity. The comparison was based on computer-based simulation of a shaking camera. MSE, PSNR and NCD errors measurements were taken. Further ways of enhancing the algorithm without significant increase in complexity are proposed. The described multiframe demosaicking algorithm is suitable for mass production devices such as mobile phones of digital cameras. Its primary goal is to replace more expensive mechanical motion compensation systems.

Index Terms — multiframe noise reduction, demosaicing, demosaicking, temporal noise reduction, multiframe demosaicking

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 noise. Current cameras perform on their physical limits and photon noise is dominant. On physical level, this type of noise can be reduced by increasing number of photons detected by each cell on a sensor. Usually, the solutions are: increasing of optical efficiency of a lens system or increasing exposure times. Improving optical efficiency is expensive as the complexity of the lens grows disproportionally relative to its quality, not to mention that camera often needs to be small. 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 on their own. The combination of these two methods only recently received a proper attention [1]. However, there is still lack of simple but effective methods which can be implemented in existing devices.

In this work the proposed method of multiframe demosaicking is compared to combination of simple frame-based demosaicking (see Figure 1) and multiframe noise reduction. The comparison is carried out using computer-based simulation of a series of shots which are shifted and rotated, then mosaicized. After that, Poisson noise is added to simulate the photon noise of a photo sensor.

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) The algorithm should operate on the fly, or in other words, user should receive the result just after the shot (no time-consuming post processing 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 wide variety of super-resolution algorithms ([2]—[5]). The requirements for memory and computational power do

Fig. 1. Bayer pattern. There are variations of it but main idea remains - two green samples per one red and blue. This pattern is chosen primarily because green colour represents luminosity.
Fig. 2. Global motion estimation using block matching. We can assume
that if rotation is small motion blocks are just shifting without rotation
not allow an increase in the resolution except by reducing
noise.

2. DESCRIPTION OF METHODS

There were three independent modules implemented:
• Global motion estimation
• Simple multiframe noise reduction based on
  "affine transform and add" method
• Multiframe demosaicking based on "affine
  transform and weighted add" method

Using the same motion estimation for both methods it is
possible to exclude it from the comparison. The description
of used global motion estimation is available below but its
optimization is a separate subject and is not described here.

2.1. Global Motion Estimation

As the global motion estimation was not an essential part of
the comparison the simplest exhaustive search was taken as
a basis. It was assumed that global motion of the frame can
be described as affine transform with relatively small
coefficients so that for small area of image it is just shift in
two dimensions. Formulas (1) and (2) define this condition:

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix}
= \begin{bmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{21} & a_{22} & a_{23} \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

(1)

Where

\[ a_{21} \ll 1, a_{12} \ll 1 \]

(2)

Then 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 or
robust fitting [6]. Figure 2 illustrates this.

Exhaustive search block matching algorithm was taken
from [7]. It was modified to introduce a penalty term for big
motion vectors. Having two vectors with equal cost the
shortest will be selected.

2.2. Noise reduction

There were the steps taken for noise reduction:

1. Demosaic every given frame with bilinear or
   more sophisticated demosaicking method.
2. Find global motions between key frame (in our
case simply first frame from the sequence) and
the rest of frames using demosaicked images.
3. Apply backward affine transform to match with
   key frame.
4. Take average colours from matched frames for
each pixel.

The following methods were also used instead of bilinear
demosaicking

1. Variable Number of Gradients [9]
2. High quality linear interpolation [10]

It is important to note that the above described scheme is
memory efficient as images can be processed one by one, or
in other words, amount of memory required does not depend
on number of frames in one sequence.

2.3. Multiframe demosaicking

The multiframe demosaicking consists of the following
steps:

1. Initial demosaicking. Bilinear demosaicking is
   used here
2. Find the global motion between frames based on
   the reconstructed images from the first step
3. Apply a backward transform to the Bayer pattern
   and Bayer pattern mask

Fusing Bayer frames and masks using an averaging method
is described below.

Steps 1 and 2 are the same as for noise reduction, but step
3 requires a more detailed explanation.

Let us assume that value of one colour in pixel with
coordinates \((x_1, y_1)\) is

\[ s_1 = s(x_1, y_1) \]
It is possible to describe a probability density function for a pixel with coordinates \((x_1, y_1)\) with known neighbour in point \((x_2, y_2)\) as follows

\[
P(s_1|s_2, \rho(s_1, s_2)) = P(s_2|s_1, \rho(s_1, s_2)) \tag{3}
\]

Where \(\rho(s_1, s_2)\) is the distance between two coordinates of pixels. It is also assumed that both values \(s_1\) and \(s_2\) are equally likely. Equation (3) is a simplified Bayes rule. For \(N\) neighbouring pixels we have

\[
P(s) = \frac{1}{c} \sum_{i=1}^{N} P(s_i) \cdot P(s|s_i) = \frac{1}{c} \sum_{i=1}^{N} P(s_i) \cdot f(\rho(s, s_i)) \tag{4}
\]

Where \(P(s_i)\) is a distribution of individual colour and \(c\) is a normalization coefficient. Form of \(P(s|s_i) = f(\rho_i)\) can be estimated from experimental data for a given set of images. The distribution should look close to Gaussian with centre in \(\rho = 0\). Mathematical expectation of \(s\) equals:

\[
<s> = \frac{1}{c} \sum_{i=1}^{N} s_i \cdot f(\rho_i) \tag{5}
\]

Where normalization coefficient:

\[
c = \sum_{i=1}^{N} f(\rho_i) \tag{6}
\]

In practice the described formula can be calculated using cumulative value and weight masks for every colour plane.

As we are working with discrete images to use formula (5) we only need to know the cumulative sum of all the values and the number of values dropped into a given location. An example is shown in the Fig. 3. For the given distribution of pixels, Equation (5) will give us value of 3. The distance from the four neighbouring pixels is the same and hence \(P(s|s_i)\) are equal and can be cancelled.

### 3. Comparison of Methods

The methods were compared using raw images generated from "Kodak Image Set". Images were downsampled to reduce simulation time. The aim was to simulate the image sequence from the real camera. Simulation was introduced into the comparison because with real raw images it is impossible to get precise colours for every pixel.

The following assumptions are used when comparing the two methods:

1. Overall exposure time of a set of shots is short (less than 1/4 second)
2. More than 16 frames to capture (each less than 1/64 second)
3. There is only rotation and shift taking place (no scaling)
4. Rotation is no more than 5 degrees between any two individual images in a set.
5. Motion shift is no more than 100 pixels between any two images in a set. This restriction is applied only for faster simulation.

For error measurement MSE, PSNR, NCD formulae were taken. MSE and PSNR are

\[
\sigma_{MSE} = \frac{1}{W \cdot H \cdot C} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{c=1}^{3} \left[ O(x, y, c) - R(x, y, c) \right] \tag{7}
\]

\[
\sigma_{PSNR} = 10 \cdot \log_{10} \left( \frac{1}{\sigma_{MSE}} \right) \tag{8}
\]

Here we assume that colour values are within the range \([0, 1]\).

#### Table I

<table>
<thead>
<tr>
<th>Number of frames in a set</th>
<th>Temporal Noise Reduction</th>
<th>Multiframe Denoising</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSE</td>
<td>PSNR</td>
</tr>
<tr>
<td>5</td>
<td>0.00408</td>
<td>29.32</td>
</tr>
<tr>
<td>10</td>
<td>0.00573</td>
<td>29.45</td>
</tr>
<tr>
<td>15</td>
<td>0.00585</td>
<td>29.40</td>
</tr>
<tr>
<td>20</td>
<td>0.001677</td>
<td>29.48</td>
</tr>
<tr>
<td>25</td>
<td>0.001340</td>
<td>29.57</td>
</tr>
<tr>
<td>30</td>
<td>0.001360</td>
<td>29.54</td>
</tr>
<tr>
<td>35</td>
<td>0.001358</td>
<td>29.53</td>
</tr>
<tr>
<td>40</td>
<td>0.001341</td>
<td>29.38</td>
</tr>
</tbody>
</table>

Fig. 4. Dependence of normalized color difference from number of images available for warping. As can be seen multiframe denoising becomes more effective with number of images higher than 35.
Fig. 5. From left to right: (a) Original image, (b) noisy image from the distribution of 32, (c) result of temporal noise reduction on 32 images, (d) result of multiframe noise reduction on 32 images.

NCD stands for Normalised Colour Difference and is used to quantify the perceptual colour difference and is defined as follows:

$$\sigma_{NCD} = \frac{\sum_{x,y} \sqrt{(L_x - L_y)^2 + (U_x - U_y)^2 + (V_x - V_y)^2}}{\sum_{x,y} L_x^2 + U_x^2 + V_x^2}$$

(9)

Where $L$, $U$, $V$ are lightness and chrominance components of result and original images in CIELUV colour space. They are converted from sRGB colour space through CIE XYZ colour space. The conversion sequence was the following:

$$sRGB \rightarrow \text{linear RGB} \rightarrow \text{CIE XYZ} \rightarrow \text{CIE LUV}$$

(10)

4. RESULTS

The simulation was performed for different number of images in a set varying from 5 to 40 with a step size of 5. The results are shown in the Table I. As can also be seen from the graph in Fig. 4 the multiframe demosaicking becomes more effective compared to temporal noise reduction as the number of images in one set increases. Actual image samples from processing are shown in Fig. 5.

It is important to note that the proposed method is not based on the specific structure of classic Bayer filter layout and can be easily adapted for alternative filter patterns such as described in [12].

5. CONCLUSION

The proposed method of multiframe demosaicking showed clear advantage over temporal noise reduction on large image sets (more than 35 images). It is also simple to implement in hardware of modern digital camera or 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 spatial filtering kernel for uniform surfaces and temporal filtering for edges.

REFERENCES