IPOL Journal · Image Processing On Line

HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH

Self-similarity Driven Demosaicking

Antoni Buades, Bartomeu Coll, Jean-Michel Morel, Catalina Sbert

article demo archive

published • 2011-06-01

reference • Antoni Buades, Bartomeu Coll, Jean-Michel Morel, and Catalina Sbert, *Self-similarity Driven Demosaicking*, Image Processing On Line, 1 (2011). http://dx.doi.org/10.5201/ipol.2011.bcms-ssdd

→ BibTeX

Communicated by Guillermo Sapiro
Demo edited by Pascal Getreuer

- Antoni Buades toni.buades@uib.es, MAP5, Université Paris Descartes
- Bartomeu Coll tomeu.coll@uib.es, TAMI, Universitat de les Illes Balears
- Jean-Michel Morel morel@cmla.ens-cachan.fr, CMLA, ENS Cachan
- Catalina Sbert catalina.sbert@uib.es, TAMI, Universitat de les Illes Balears

Content

Overview

Digital cameras record only one color component per pixel, red, green or blue. Demosaicking is the process by which one can infer a whole color matrix from such a matrix of values, thus interpolating the two missing color values per pixel. The selected color configuration of the sensor usually follows the Bayer color filter array, CFA. Out of a group of four pixels, two are green (in quincunx), one is red and one is blue. The Self Similarity driven Demosaicking (SSD) treats the Bayer CFA configuration, but the proposed algorithm is easily adapted to other CFA configurations.

The basic observation used in demosaicking algorithms is that in the Bayer CFA configuration, a higher resolution is obtained in the interpolated green channel. For this reason, most methods first interpolate the green channel and then use it to drive the interpolation of the red and blue channels. The second key observation is that at the same time the high frequencies of the sub-sampled red and blue channels (second derivatives) can be used to interpolate the green channel. The first method proposing the use of these observations was the Hamilton-Adams algorithm [1].

Adams-Hamilton algorithm

The green channel is horizontally or vertically interpolated at each pixel depending on the green first derivatives and the red and blue second derivatives as follows.

🟥	🟩	🟥 1	🟩	🟥
🟩	🟦	🟩 2	🟦	🟩
🟥 3	🟩 4	🟥 5	🟩 6	🟥 7
🟩	🟦	🟩 8	🟦	🟩
🟥	🟩	🟥 9	🟩	🟥

1. Calculate horizontal gradient
   $\Delta H = |G4 - G6| + |R5 - R3 + R5 - R7|$
2. Calculate vertical gradient
   $\Delta V = |G2 - G8| + |R5 - R1 + R5 - R9|$
3. If $\Delta H > \Delta V$,
   $G5 = (G2 + G8)/2 + (R5 - R1 + R5 - R9)/4$
   Else if $\Delta H < \Delta V$,
   $G5 = (G4 + G6)/2 + (R5 - R3 + R5 - R7)/4$
   Else
   $G5 = (G2 + G8 + G4 + G6)/4 + (R5 - R1 + R5 - R9 + R5 - R3 + R5 - R7)/8$

The red and blue channels are interpolated taking into account the already filled green channel. A bilinear interpolation is applied to the differences R-G and B-G as originally proposed by Cok [2].

The use of the high frequencies of the sub-sampled red and blue channels (second derivatives) has been incorporated by all advanced algorithms. We will use the Hamilton-Adams algorithm as initialization.

The SSD demosaicking method is divided in two steps. In the first step the image geometry is reconstructed by using its non-local self-similarity. In the second step the colors are updated by transferring information from the green to the red and blue channels. The second step is called chrominance regularity. It subsumes the observation, underlying most algorithms, that the high frequencies of the three channels are very similar. The first step instead performs a non-local estimate taking advantage of all image self-similarities. This step is crucial for the retrieval of fine geometric structure close to the Nyquist frequency.

References

1. J. Hamilton Jr and J. Adams Jr, “Adaptive color plan interpolation in single sensor color electronic camera,” 1997, US Patent 5,629,734.
2. D. Cok, “Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal,” 1987, US Patent 4,642,678.
3. A. Buades, B. Coll, J.M. Morel, C. Sbert "Self similarity driven demosaicking", IEEE Transactions on Image Processing, Vol. 18(6), pp:1192-1202, 2009. DOI:10.1109/TIP.2009.2017171
4. A. Buades, T. Coll and J.M. Morel, Image data processing method by reducing image noise, and camera integrating means for implementing said method, EP Patent 1,749,278 (Feb. 7, 2007).

Algorithm description

Initial interpolation by Adams-Hamilton algorithm

The CFA image is interpolated by using the Adams-Hamilton algorithm, getting a full color image u_0.

The self similarity driven step (NL_h)

The missing color values at a pixel *p* are interpolated by averaging original CFA values of the same channel with a similar color neighborhood in the initialization image $u_0 = (r_0, g_0, b_0)$. In other terms, color inter-correlation is enforced by computing the distance in the three-channel image.

Denote by Ω_u the set of pixels where the channel $u \in \{r, g, b\}$ was originally available in the CFA. The averaging process for a pixel *p* reads,

$$NL_h u(p) = \frac{1}{C_u(p)} \sum_{q \in \Omega_u} w(p,q)\ u(q), \quad w(p,q) = e^{-\frac{1}{h^2} \sum_{t \in W} \|u_0(p+t) - u_0(q+t)\|^2}$$

with $p \notin \Omega_u$ where $C_u(p)$ is the normalization factor, equal to the sum of the weights and *W* indicates a square window of fixed size centered at the origin. The weight *w(p, q)* depends on the Euclidean distance of the two color windows centered at *p* and *q*, $u_0(p+W)$ *and* $u_0(q+W)$. Thus, pixels with a small patch Euclidean distance will have a larger weight in the average.

The parameter *h* controls the decay of the exponential kernel. When *h* is small only pixels having a very similar patch around are taken into account into the average. When using a larger value of *h*, pixels with a more different neighborhood can take part in it. When *h* tends to infinity, the weighted average tends to a pure arithmetic mean of all pixels of the same color.

In order to gradually correct the erroneous structures and artifacts of the initial image, a coarse to fine strategy is applied. This process refines at each step the similarity search by reducing the value of *h* from *16* to *4* and *1*. As it is commented in [3], these values have been experimentally fixed using the database Kodak and IMAX. By testing only in the Kodak database, results are improved by beginning with *h=32* but results get worse on the IMAX database.

Chromatic regularization (CR)

In contrast with all mentioned algorithms the similarity driven algorithm does not involve at this point any assumption on the regularity of the chromatic components of the image. Since the similarity driven super-resolution step fills in all color channels, the chromatic regularity assumption can be applied to the chrominance *(U, V)* instead of the differences red - green and blue - green. Recall that

$$Y = a_r r + a_g g + a_b b, \quad U = r - Y, \quad V = b - Y$$

with $a_r = 0.299$, $a_g = 0.587$ and $a_b = 0.114$. Notice that the green dominates in this linear combination. This coordinate system aims at separating the geometric information contained in Y from the chromatic information *U*, *V*, and gives more importance to the green channel.

The chromatic regularity step (CR) decomposes the color image into the YUV components. A median of the coordinates U and V in a 3 × 3 neighborhood is applied obtaining U0 and V0. This median is applied to all the pixels independently of the CFA mask.

The RGB components are obtained from $Y\ U_0 V_0$, and the original values of the CFA are put back at pixels where they are known.

Final algorithm

The final method alternates each iteration (with decreasing *h*) of the self-similarity driven algorithm NL_h and the chromatic regularity step CR(u). This final strategy is summarized in the following pseudo-code which takes as input the values of the CFA mask, I,

```
u0 <- Initial_Interpolation(I);
for h in {16,4,1} do {
    u <- NL_h(u0);
    u <- CR(u); u0 <- u;
}
Output <- u;
```

Parameters

For computational purposes the search for similar pixels of the self similarity step is restricted to a search window of 15×15 around each pixel. Even if many similar pixels can be found far away, there is usually enough information kept by taking a large search window.

The window comparison is performed by a flat Euclidean distance on a 3×3 neighborhood. This window has shown to be robust (27 values) and small enough to take care of details and fine structure.

The exponential function used for weight computation in the similarity driven step is replace by a look up table reducing the overall time of computation in a non negligible factor.

On Line Demo: Try It!

An on-line demo of this algorithm is available.

The demo permits to simulate the Bayer CFA and apply the demosaick it by using the Self Similarity Driven Demosaicking algorithm.

Source Code

A C implementation is provided. It should compile on any system since it's only strict ANSI C.

Some of the files use algorithms possibly linked to the cited patents [1,2,4]. These files are made available for the exclusive aim of serving as scientific tool to verify the soundness and completeness of the algorithm description. Compilation, execution and redistribution of these files may violate exclusive patents rights in certain countries. The situation being different for every country and changing over time, it is your responsibility to determine which patent rights restrictions apply to you before you compile, use, modify, or redistribute these files.

The rest of files are distributed under GPL license.

- source code : tar/gz
- source code documentation : tar/gz online

Basic compilation and usage instructions are included in the `README.txt` file. This code requires `libtiff` and is limited to 8bit RGB TIFF image files.

The same code is used for the online demo

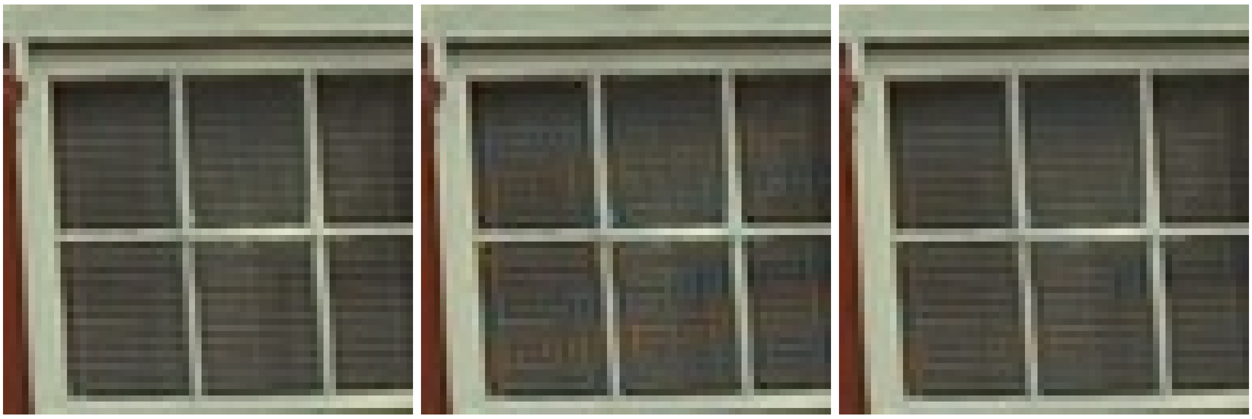
The same code is used for the online demo.

Examples

The examples below illustrate several aspects of the demosaicking algorithm. We display the original image, the Hamilton-Adams algorithm applied to the simulated CFA and the SSD demosaicking algorithm. We remind that the Adams-Hamilton is used as initialization for the SSD algorithm.

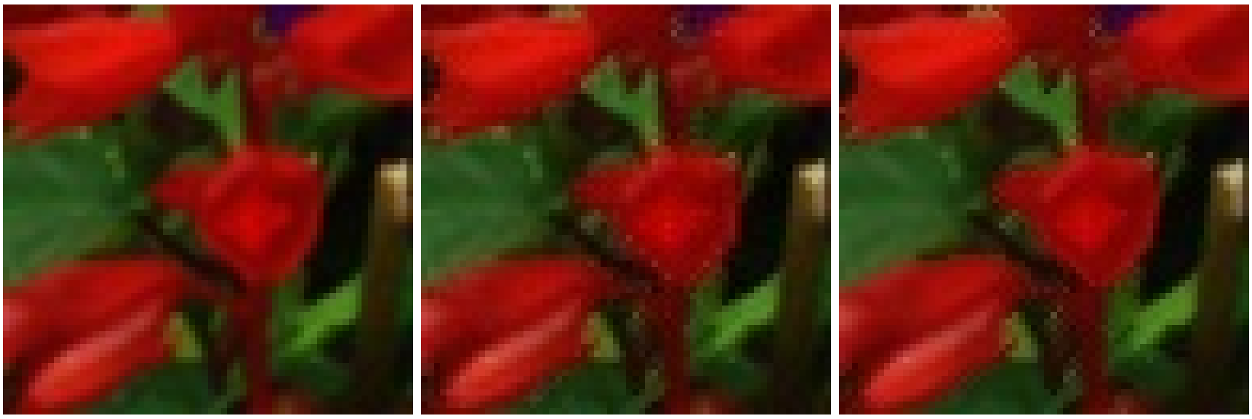
Structures at a critical frequency are difficult to interpolate. The SSD algorithm is able to correctly reconstruct the patterns by involving both the self similarity and the chromatic regularity.

original *Hamilton-Adams* *SSD*



original *Hamilton-Adams* *SSD*

When in an image zone, the channels do not have exactly the same high frequencies, artifacts can appear. The non local regularization permits to attenuate this erroneous pattern.



original *Hamilton-Adams* *SSD*

feeds & twitter • sitemap • contact • privacy policy • ISSN: 2105-1232 • DOI: 10.5201/ipol
supported by CMLA, ENS Cachan • DMI, Universitat de les Illes Balears • Fing, Universidad de la República • Gandi.net