

# Digital Image Correlation and Color Cameras

A. Baldi<sup>1</sup> 

Received: 25 April 2017 / Accepted: 2 October 2017 / Published online: 13 October 2017  
© Society for Experimental Mechanics 2017

**Abstract** Digital Image Correlation (DIC) is a well-known experimental technique. It works by constructing a (surjective) mapping of pixel intensity from reference to target image, where the mapping parameters are identified using a Least Squares approach. Because it makes use of the luminance component of the image, Digital Image Correlation is usually implemented by assuming monochrome cameras. In this work, we will discuss its implementation when color cameras are used, focusing on pitfalls and potential advantages of this solution. Since most cameras implement color acquisition using a Color Filter Array (CFA), much of the article will focus on this technology. However, we will not limit ourselves to this aspect and will show that Three-CCD cameras can provide significant advantages over both CFA and monochrome cameras.

**Keywords** Digital image correlation · Color filter array · Bayer pattern · Color cameras · Demosaicing algorithm

## Introduction

Digital Image Correlation (DIC) is a full-field experimental technique widely used to estimate the displacements (strains) of target objects [1–3]. It works by correlating two images, acquired before and after the event of interest, under the assumption that the image intensity of each point

does not depend on motion. Although this assumption is theoretically quite difficult to satisfy exactly—it requires a homogeneous and isotropic illumination [4]—the standard experimental conditions are near enough to these requirements to allow the technique to work. To obtain field data, the image is partitioned into smaller parts (subsets) and a local mapping from the reference to the target image is assumed, i.e.  $f(x_i, y_i) = g(x_i + u, y_i + v)$ , where

$$\begin{aligned} u &= p_0 + p_1\xi + p_2\eta + [p_3\xi^2 + p_4\xi\eta + p_5\eta^2] \\ v &= q_0 + q_1\xi + q_2\eta + [q_3\eta^2 + q_4\xi\eta + q_5\xi^2] \end{aligned} \quad (1)$$

whereas  $\xi$  and  $\eta$  are the local coordinates parallel to the  $x$  and  $y$  axis ( $\xi = x - x_0$ ,  $\eta = y - y_0$ ) and  $(x_0, y_0)$  is the origin of the local reference system.

The  $p_i$  and  $q_i$  parameters control the mapping functions (equation (1)) and are usually computed by minimizing over the area of the current subset a suitable error function [5], e.g.

$$C_{ZNCC} = \frac{\sum \bar{f}_i \bar{g}_i}{\sqrt{\sum \bar{f}_i^2 \sum \bar{g}_i^2}} \quad (2)$$

or

$$C_{PSSDab} = \sum (af_i + b - g_i)^2 \quad (3)$$

where  $f_i$  and  $g_i$  are respectively the intensity of pixel  $i$  in the reference and target images,  $\bar{f}_i = f_i - \bar{f}$ ,  $\bar{g}_i = g_i - \bar{g}$  and  $\bar{f}$  and  $\bar{g}$  are the mean values of  $f$  and  $g$  over the corresponding subsets. The coefficients  $a$  and  $b$  appearing in the  $C_{PSSDab}$  error criterion account for offset and scale change of the target system intensity.

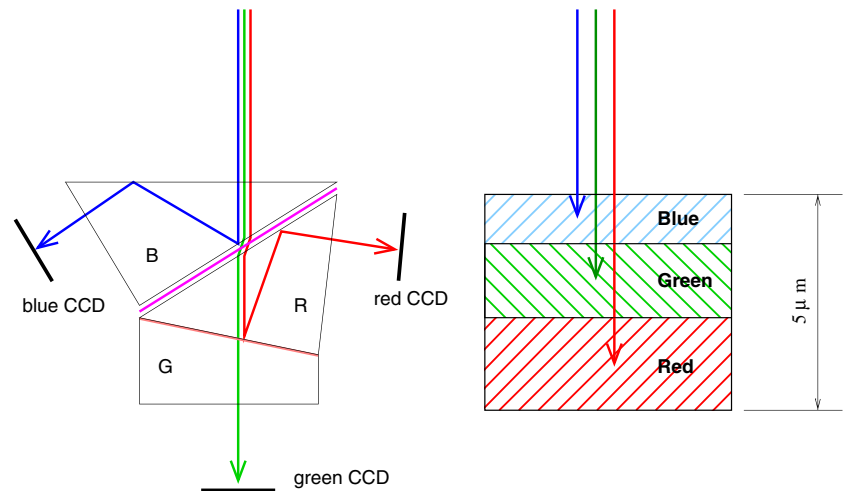
To obtain a solution—the set of  $p_i$  and  $q_i$  minimizing the error functional—either  $f(x, y)$  or  $g(x, y, u, v)$  is expanded in Taylor's series truncated to the first order, then a solution system is computed by setting to zero the derivatives

✉ A. Baldi  
antonio.baldi@dimcm.unica.it

<sup>1</sup> Dipartimento di Ingegneria Meccanica, Chimica e dei Materiali, Università degli Studi di Cagliari, Via Marengo, 2, 09123 Cagliari, Italy



**Fig. 1** True-color cameras can be produced using either a Phillips prism to separate the three color components (left), or a multi-layer sensor (right), where each layer is sensitive to a different subrange of the color spectrum. No interpolation is performed in either case, so by assuming the same size of the sensor and the same pixel dimensions, three times the number of uncorrelated data is available



of the error function with respect to the control parameters  $p_i$ ,  $q_i$  (and eventually  $a$  and  $b$ ). Although the resulting system appears to be linear, its solution does not correspond to the sought set of parameters because the numerical value of the computed derivatives depends on the point of evaluation, i.e. it depends on the solution. Thus, the above-described procedure has to be iterated up to convergence.

The solution algorithm sketched above implies that a unique intensity pattern exists for each location, thus the surface of the specimen has to be textured either naturally or artificially (by spraying random speckles on the surface); moreover, the algorithm requires comparison of the value of the intensity at location  $g(x_i + u, y_i + v)$  with  $f(x_i, y_i)$  by means of equation (3) (or equation (2)) for each step of the iteration and for all points  $i$  belonging to the current subset; thus, considering that both  $u, v \in \mathbb{R}$ ,  $N$  interpolations are required for each step, where  $N$  is the number of pixels belonging to the current subset.

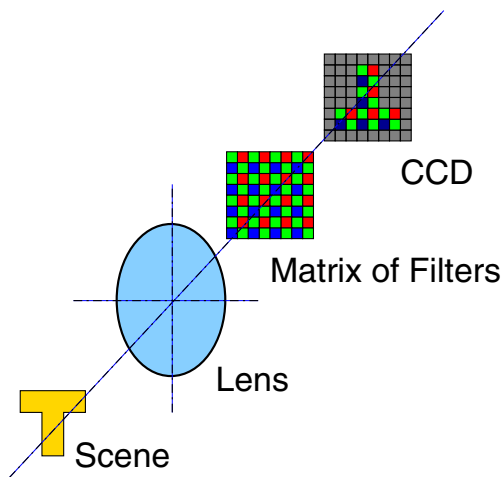
Intensity interpolation is a critical point in DIC and several works have been devoted to this point [6–10]. Indeed, the theoretically exact interpolating function is the *sinc*, defined as  $\text{sinc}(x) = \sin(\pi x)/(\pi x)$ , because its Fourier transform is unitary up to the Nyquist frequency and exactly zero for frequencies above it. However, the sinc has an infinite support and the convergence of the series is very slow, thus making its use impractical.<sup>1</sup> The standard solution to this problem is polynomial interpolation, but its use introduces a systematic error in the DIC-estimated displacements: even when no noise is present, the fractional part of the displacements shows a sinusoidal-like error whose

amplitude depends on the type of polynomial. The error is obviously null for integer displacements (no interpolation is required at these locations) and, by symmetry, at  $dx = 0.5$  pixel.

The solution algorithm sketched above naturally fits gray color images. However, monochrome cameras are mainly used for scientific applications. On the contrary, owing to the wide diffusion of digital cameras, a large selection of consumer digital cameras is available. Thanks to the huge market (when compared to the scientific area), the number of pixels and the dynamic range of these cameras are significantly better than same-price scientific instruments; moreover, a larger set of optics is usually available, thereby allowing much more flexibility. Thus, it makes sense to try to use them. But, apart from a few notable exceptions—e.g. the Leica M Monochrom—all consumer cameras are equipped with a color sensor.

The first color picture was taken in 1861 by Thomas Sutton, following an idea suggested by Maxwell in 1855 [11]: using a rotating disk painted with different ratios of red, green and blue, Maxwell had shown that all colors can be obtained as a combination of three components; thus, Sutton performed three monochrome acquisitions, each using a different color filter; the color image was obtained by superimposing their projections. The technique was quite rough and Maxwell commented on the inadequacy of the result, nevertheless, apart from the technical improvements, we are still using the same approach. Current color cameras work either in full-color or interpolating mode. The former (known as true-color or three-CCD cameras) acquire all color components for each pixel (Fig. 1), whereas the latter (known as Color Filter Array or Bayer cameras) acquire only one of them (Fig. 2). True color cameras divide the spectrum into three components using either a specially designed prism [12, 13] and three CCDs or a specialized sensor where each pixel consists of three layers, each sensitive to a different

<sup>1</sup>Truncation or windowing of the theoretical function allows generation of a more workable interpolant. However, both operations imply an energy leakage in the frequency plane and the theoretical properties of the sinc are completely lost.



**Fig. 2** A CFA (Color Filter Array) color camera acquires a single color component for each pixel. To this end, a color filter is installed in front of each pixel of the CCD. The missing color components are extrapolated using data from neighbor pixels

wavelength range.<sup>2</sup> Both approaches are difficult to implement, because of alignment issues in the former or the non-standard CCD architecture in the latter, thus only a few camera models adopt these techniques.

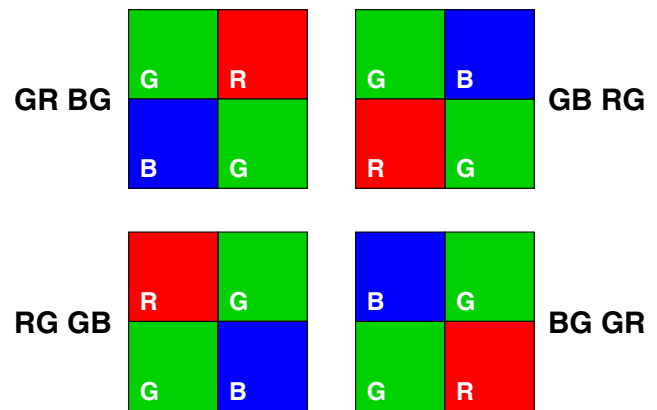
Most color cameras adopt the CFA (Color Filter Array) approach. They work under the assumption that the color fields are continuous and mostly smooth; thus, instead of acquiring all three color components at each location, it suffices to sample them on a regular grid. At each pixel location a single color component is acquired, whereas the missing ones are estimated by interpolating data from neighbor pixels. The implementation of this idea is relatively simple: it requires only the installation of a matrix of color filters in front of the CCD. Filters are organized in a regular pattern, known as the Bayer pattern [14], which ensures that reliable color data are available at each location (Fig. 3).

The advantages are significant: a single sensor suffices to acquire a color image, the electronics is simpler and the required transfer rate of the data bus is lower. Finally, there is no space limitation for the lens.<sup>3</sup> On the contrary, the acquired data cannot be used “as is” and a post-processing step (known as the demosaicing process) is required to reconstruct the three continuous color planes starting from the raw data.

Each pixel of a color image stores a set of color components (usually the three additive primary colors [15], Red,

<sup>2</sup>In the following, we will call both types of cameras *three-CCD cameras* because both configurations require three sensors (a layer can be viewed as a separate CCD).

<sup>3</sup>The color-separating prism has to be installed between the lens and the CCDs. This constitutes a serious limit for both the optics and the sensor if standard lenses are to be used.



**Fig. 3** A Bayer pattern consists of four pixels organized as a  $2 \times 2$  cell. Each cell contains two green, one red and one blue pixel. The figure shows the four possible pixel orderings, named by the initial of the color of the filter (top-bottom, left-right). Cells repeat themselves both vertically and horizontally, so that for each color there is at most only one missing pixel in the vertical, horizontal and diagonal directions

Green and Blue (RGB), but sometimes Cyan, Magenta, Yellow and Black (CMYK) or luminance and two chrominance components ( $Y C_b C_r$ ). Focusing on the RGB encoding, a color image can be viewed as a stack of monochrome images (color planes), each related to a specific wavelength range. To process them using DIC, only a few options are possible:

- use only a single plane, discarding all the remaining data;
- compute the luminance from the color components and use it as a “standard” monochrome image;
- extend the error functional to process all the data.

The first option is very simple to implement because it requires only a simple preprocessing of the images, but does not use the available data efficiently: indeed, if a three-CCD camera is used for acquisition, we are discarding  $\frac{2}{3}$  of its information content; on the contrary, if the image is acquired using a CFA camera, the data of all color planes are incomplete and an interpolation has to be performed (more on this in the following sections).

The second option is only marginally better: it reduces noise to some extent on three-CCD cameras, but still does not use data efficiently; on the contrary, if a CFA camera is used for the acquisition, two of the three color components involved in the computation of the luminance are not statistically independent but result from an interpolation, thus giving unsatisfactory results.

If we bear in mind that a color image consists of a stack of three images, the latter option is relatively straightforward to implement; it suffices to compute the error functional plane by plane, incrementally summing the contribution of each

of them to the total error [16, 17]. Using this approach, the  $C_{PSSDab}$  functional becomes

$$C_{PSSDab} = \frac{\sum (af_i^R + b - g_i^R)^2}{\sum (f_i^R)^2} + \frac{\sum (af_i^G + b - g_i^G)^2}{\sum (f_i^G)^2} + \frac{\sum (af_i^B + bg_i^B)^2}{\sum (f_i^B)^2} = \sum_k \frac{\sum_i (af_i^k + b - g_i^k)^2}{\sum_i (f_i^k)^2} \quad (4)$$

where  $k$  extends over all the color planes and we inserted an (optional) normalizing factor. Note that although the modifications to the code are marginal, an efficient use of equation (4) may require using colored speckles: indeed, if a black and white object is imaged using a color camera, the red, green and blue planes store the same information. Because the data are correlated, no improvement has to be expected from the use of a larger number of samples; employment of equation (4) simply results in a waste of processing time.

This work is organized as follows: because we need to know the expected result to estimate errors, the next section is devoted to the description of a speckle image generator and focuses on the emulation of the behavior of CFA and three-CCD cameras. The following section will concentrate on the analysis of the performance of DIC when color images are used, starting from the true RGB picture to move to CFA images. Much of this section will be devoted to the analysis of various demosaicing options and will show that it is possible to obtain results significantly worse than, or similar to, a monochrome camera, depending on the algorithm. Later, we will show some experimental results of a commercial CFA camera (a Nikon D700). The last section summarizes the work and sketches some conclusions.

## Generating Synthetic Color Images

To compare the accuracy of the various approaches to color-DIC, the expected results have to be known. Thus, we opted for starting our analysis by using a numerical image generator.

Synthesizing speckle images is not a simple task because of the aforementioned effect of interpolation on DIC-estimated displacements. This subject has been discussed extensively in the technical literature and three different approaches have been proposed: interpolation [6, 18, 19], super-sampling [8] and known texture functions [20–23].

Our implementation follows the third approach [9]: the speckle field is described as the sum of several bell-shaped functions sprayed over the surface of the image:

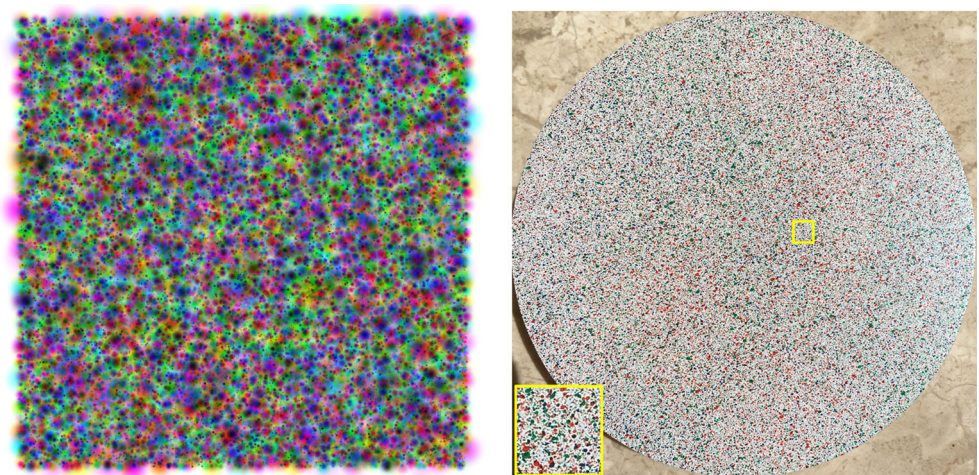
$$b(r) = \begin{cases} s [1 - (r/\rho)^2]^3 & r \leq \rho \\ 0 & \text{elsewhere} \end{cases}$$

where  $r$  is the radius from the center of the bell,  $s$  is the scale factor and  $\rho$  the radius at which the function nullifies. All of them are randomly generated, but known. Integration is performed numerically by super-sampling each pixel (i.e. instead of estimating the inverse mapping, we subdivide the surface of each pixel into sub-areas and cumulate their contributions to the intensity of the mapped pixels in the target image).

The tool sketched above generates monochromatic images, but it can easily be adapted to RGB image generation: it suffices to generate three different speckle fields (to have uncorrelated data between the various channels) taking care to deform all of them using the same displacement function. The three images thus obtained can easily be combined (e.g. using the public domain software *imagej*) and become the color channels of a RGB image (see Fig. 4).

It is worth noting that the spectral sensitivity of a CCD across the visible region is not uniform: it is usually

**Fig. 4** Color speckle. Left: synthetic image; right: real image. Note that the speckle size has been significantly enlarged in the numerically generated image to facilitate visualization. Due to the overlapping of the speckle fields, mixed colors appear. In the inset of the real image is a magnified view of the speckle field



maximum in the green region, somewhat lower in the blue range and significantly smaller in the red area. As an example, the scale factors suggested by *dcrw*<sup>4</sup> for a Nikon D700 camera are 2.06, 0.93 and 1.11 (for the red, green and blue components respectively). This means that to obtain a daylight white the red and blue signals have to be multiplied by 2.21 and 1.19 respectively, with respect to the green one. Keeping in mind that a CCD uses a single Digital-Analogue Converter for all pixels, this implies that the illumination-independent component of noise in the red channel is more than twice as large as in the green one. Moreover, the above coefficients account only for the camera; actually the spectral content of the image impinging the sensor also depends on the spectrum of the light source. Thus, the real values of the scaling factors are always uncertain. During image generation we assumed a unitary scale factor for the green component and 0.55 and 0.8 respectively for the red to green and the blue to green ratios.

## Using Color Images with DIC

Digital Image Correlation can be combined with color images in several ways [24–26]. Various processing approaches are possible depending on the combination of two options:

- the sample can be painted either in the “standard” way, i.e. using black speckles on a white background or using three independent speckle fields (red/green/blue, as shown in Fig. 4);
- the camera can be either a three-CCD (or similar) or a CFA one;

Moreover, given an image acquired with a CFA-camera, there are several different ways to recover continuous fields.

In any case, to assess the performance of the various solutions we need a test case. Because all the data sets available in the DIC Challenge website [27] are monochrome, we opted to use our in-house-developed synthetic image generator and a simple parabolic displacement function:  $u(x, y) = \alpha x^2$  where  $\alpha$  moves from 0 to  $100 \cdot 10^{-6}$  in five steps (from first to last image), the image size is  $1320 \times 1170$  pixel (rows  $\times$  columns) and the origin of the displacement field is located at (30, 30).

The proposed displacement field ensures linearly increasing strains ( $\varepsilon_{xx} = 2\alpha x$ ), thus,  $\max(\varepsilon_{xx}) = 0.2 \frac{\text{m}}{\text{m}}$  for  $x = 1000$ ), so the identification should be progressively more difficult with growing  $x$ .

<sup>4</sup>*dcrw* is an open-source computer program able to read numerous raw image formats. The stated objective of the author of the code is to be able to decode any raw image from any digital camera on any computer running any operating system.

## Monochrome Cameras

Monochrome DIC is the obvious reference in assessing the performance of the various algorithms. To this end, we used the green component of the RGB speckle images as monochrome data. We analyzed an area of  $1240 \times 1000$  pixels (bottom left corner: (30,30), top right corner: (1030,1270), i.e. from  $x = 0$  to  $x = 1000$  in the displacement space) using a subset-based code. The active area was sampled on a regular grid using a step of 2 and 31 pixels in the  $x$  and  $y$  directions respectively. The relatively large step in the  $y$  direction ensures that measurements within the same column are uncorrelated, thus allowing a proper statistical treatment, whereas the large oversampling in the  $x$  direction is for display purposes only. Figure 5 top shows the errors  $\Delta_u$  between DIC-estimated and theoretical displacements as a function of the  $x$  coordinate of the center of the subset. The plotted values correspond to the averaged errors along each column, whereas each curve is related to a different “loading” step (i.e. to an increased value of the  $\alpha$  parameter). The location of the first twenty points where the theoretical displacement becomes an integer is shown for each curve (i.e. we solved for  $x$  the equation  $\alpha x^2 = i$ , with  $i = 1, \dots, 20$  and we put a mark on the related curve). Looking at the figure, it is easy to recognize the effect of shape-function-under-matching bias [28] (the offset from zero). In the same way, the interpolation-induced bias is easily identifiable. Note that owing to the parabolic behavior, the distance between successive points becomes progressively smaller. The amplitude of the oscillation thus decreases and eventually disappears, depending on the number of periods inside a subset (see also Appendix).

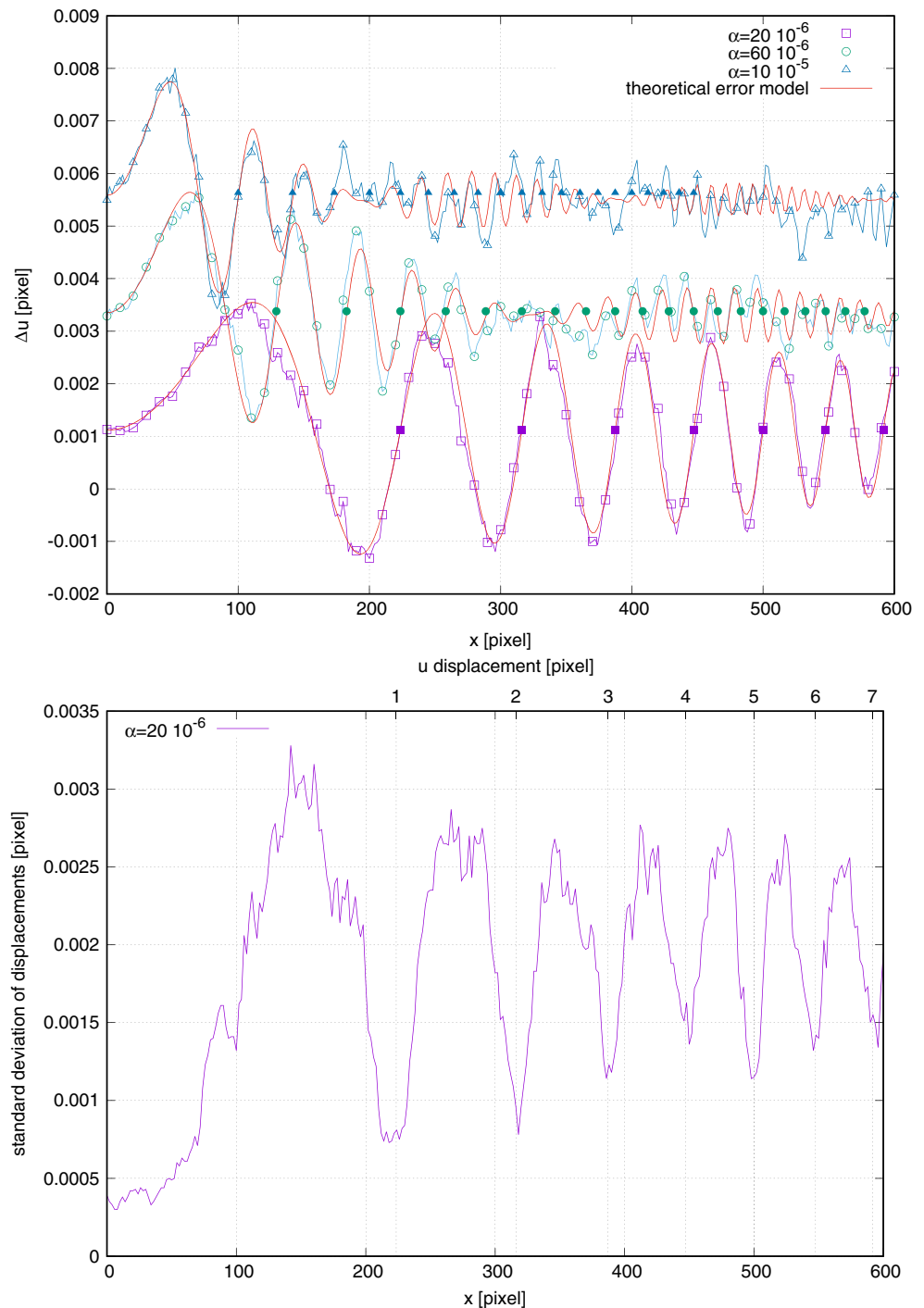
Figure 5 bottom shows the standard deviation of the horizontal displacements of each column of the grid of subsets related to the first “loading” step ( $\alpha = 20 \cdot 10^{-6}$ ). The sequence of bell-like oscillations is easily correlated to the polynomial bias (standard deviation shows a minimum whenever the displacement assumes an integer value, see top tics). Because the displacement error is mainly controlled by the order of the shape functions used for the local description of the displacement field as well as by the polynomial bias, we expect no significant variation in the different tests; on the contrary, the standard deviation changes significantly: Fig. 5 bottom constitutes the reference value for all the successive analyzes.

## Three-CCD Cameras

Three-CCD cameras acquire three times the number of independent data with respect to the same-size (rows by columns) monochrome cameras. It is thus obvious to expect significantly better results [29], provided that a consistent experimental procedure is adopted.



**Fig. 5** DIC error with respect to imposed displacement. Monochrome images. Top: displacement errors; the three graphs are related to  $\alpha = 20 \cdot 10^{-6}$ ,  $60 \cdot 10^{-6}$  and  $100 \cdot 10^{-6}$ . Each curve shows, apart from displacement error, the theoretical error (see Appendix) and, as solid dots, the location of the first 20 integer displacement points (i.e. the  $x$  where displacement becomes 1, 2, 3, ..., 20). Bottom: standard deviation of displacements ( $\alpha = 20 \cdot 10^{-6}$ ). Note that the horizontal range has been truncated at  $x = 600$

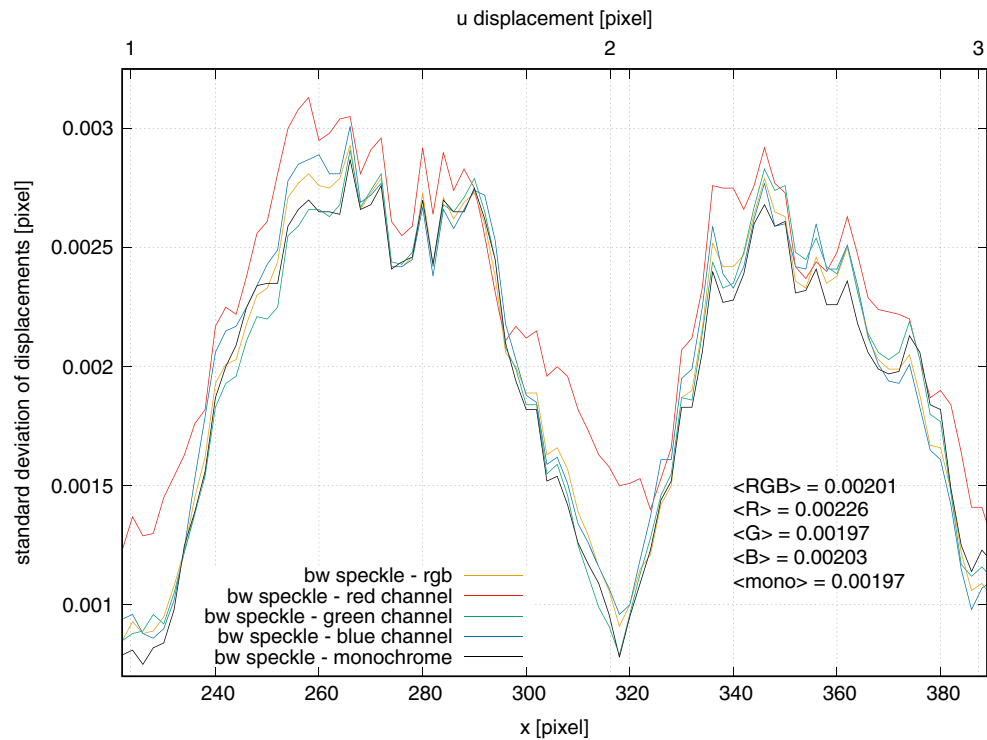


The use of a monochromatic speckle field eliminates all the advantages of using a color camera because the camera acquires essentially the same data on each channel.

Figure 6 shows the standard deviation of displacements resulting from the DIC analysis of a black and white speckle field imaged with a three-CCD camera (See Algorithm 1 for a detailed description of the generation procedure). Different processing approaches were considered, but no significant variation of results was observed, apart from

small differences related to the noise content of the channels. Thus, using only the red component of the image (the noisiest) gives the largest standard deviation; the blue channel is the second channel in terms of noise and consequently it shows the second mean value of standard deviation. The green channel gives almost the same performance as a monochrome camera (gain is unitary). Finally, the use of a color DIC code to process all the components of the image provides no advantage (results are actually a bit worse than

**Fig. 6** Standard deviation of displacements. Three-CCD camera, black and white speckle field. Values reported between < and > are the mean value of standard deviation of displacements; in particular, <mono> refers to a monochrome camera; <R> uses only the red channel of a color camera; <G> ditto, green channel; <B> ditto, blue channel; <RGB> refers to full RGB processing



a monochrome camera because of the noisier red and blue channels), and requires almost three times the processing power of a monochrome analysis.

**Algorithm 1** Generation of color components  $\mathbf{c}_i$  of a monochromatic speckle field (three-CCD camera)

```

1 generate a monochromatic reference speckle field  $\mathbf{r}$ ;
2 foreach color channel  $i$  do
3   generate a noise field  $\mathbf{n}_i$ ;
4   scale the noise field by the sensitivity factor  $s_i$  of
   channel  $i$ , i.e.  $\mathbf{n}_i \leftarrow s_i \mathbf{n}_i$ ;
5    $\mathbf{c}_i \leftarrow \mathbf{r} + \mathbf{n}_i$ ;
6   clamp( $\mathbf{c}_i$ );
7 end

```

On the contrary, Fig. 7 shows the results of the analysis using a RGB speckle field. As in the previous case, displacement errors are almost the same as in the monochrome case (Fig. 7 top) but the mean value of standard deviation of displacements moves from  $197 \cdot 10^{-5}$  to  $123 \cdot 10^{-5}$  (Fig. 7 bottom), i.e. almost exactly the expected (theoretical) improvement ( $197/\sqrt{3} = 113$ ).

### CFA Cameras

The output of low cost CFA cameras is usually a jpeg image, but it cannot be used for DIC analysis because of the low-pass filtering used by jpeg encoding and because

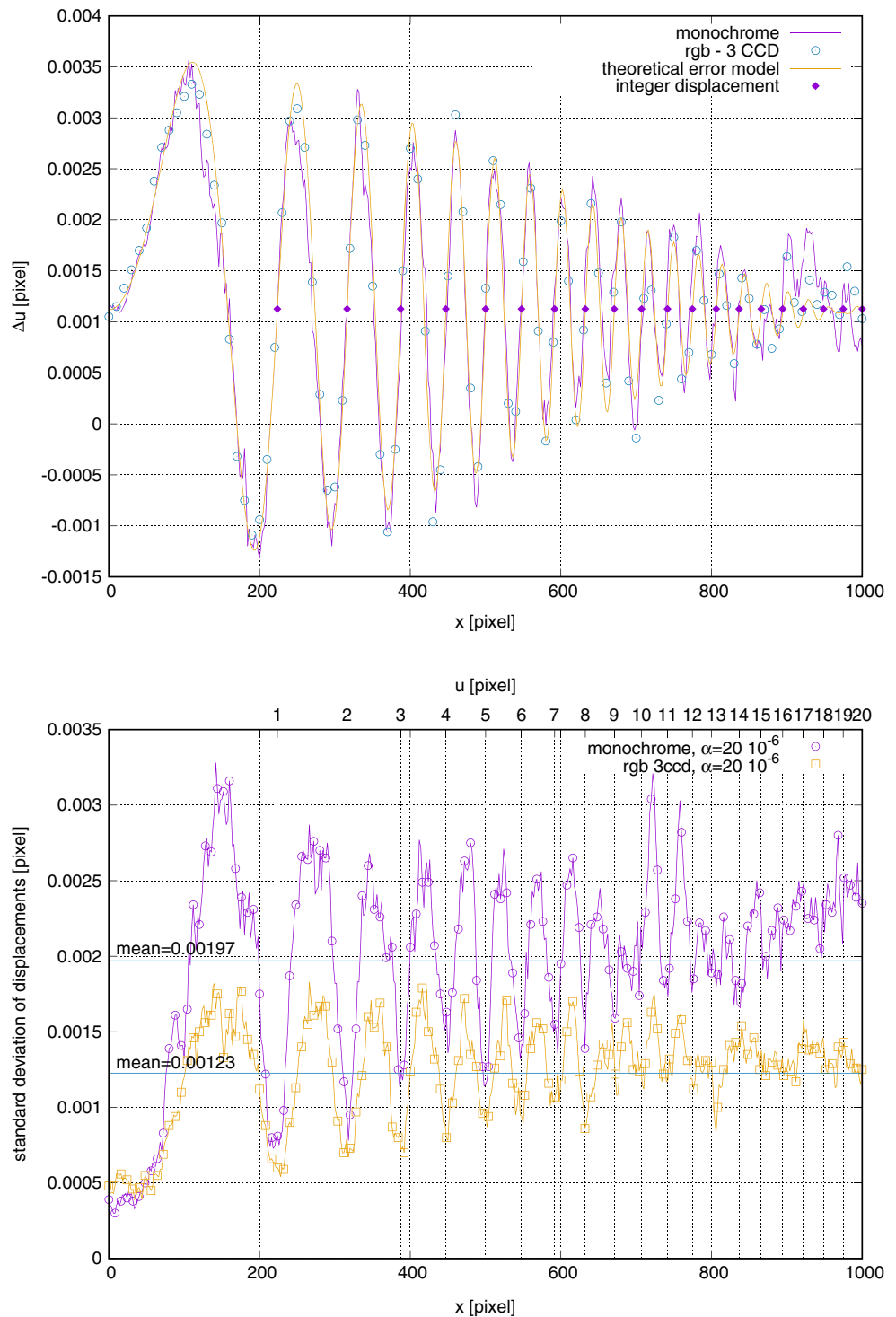
of the (usually unknown) demosaicing algorithm. Indeed, several demosaicing algorithms have been proposed [30, 31], but none of them is suitable for DIC because their objective is the improvement of the visual apparency of the reconstructed image, mainly focusing on the treatment of aliasing effects near the border of objects and on the accuracy of color reconstruction. Although they differ under many aspects, a common framework is easily recognizable: all of them perform some form of interpolation, working either in the image or frequency space, usually followed by post-processing steps (usually nonlinear). Moreover, it is common practice to process the chrominance and luminance components separately (not necessarily using the same algorithm). This significantly affects the results of DIC analysis, which appear substantially distorted.

For this reason, the standard processing pipeline of CFA images cannot be used; however, mid and high-level camera models give direct access to the acquired data, normally using a proprietary file format,<sup>5</sup> but there are several commercial and public domain programs able to extract the relevant information for successive post processing.

It is to be noted that a CFA camera acquires one input for each pixel, thus, from a statistical viewpoint, it is impossible

<sup>5</sup>Almost all of them use a TIFF framework with proprietary fields. File extension changes, depending on the camera maker, e.g. .CRW (Canon), .3FR (Hasselblad), .NEF (Nikon), .ORF (Olympus), .X3F (Sigma).

**Fig. 7** Three-CCD cameras, color speckle ( $\alpha = 20 \cdot 10^{-6}$ ). Top: displacement errors with respect to imposed displacement. Bottom: Standard deviation of displacements

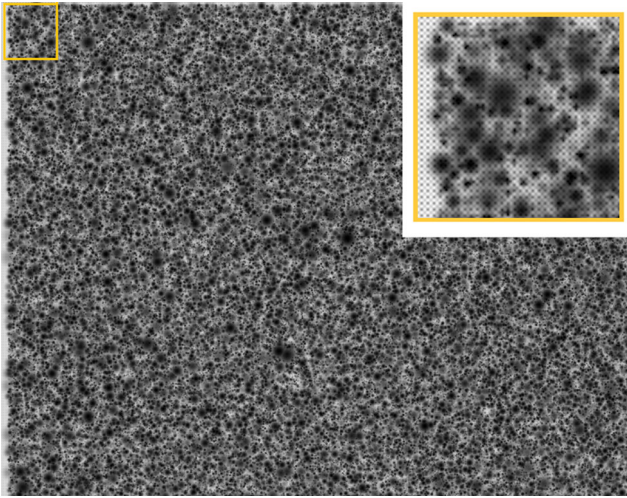


to obtain results better than those of a monochrome camera. In the following, various processing options will be analyzed and compared, with the explicit objective of approaching the performance of monochrome DIC as closely as possible. Thus, only monochrome speckle fields will be taken into account.

#### *Bare CFA image*

At first glance, a raw CFA image of a monochrome speckle field looks quite “normal”. However, on magnifying the image, the Bayer pattern clearly becomes visible (Fig. 8). One could object that this is not a problem, because adding





**Fig. 8** A black and white speckle field acquired using a CFA camera. In the inset, a magnification of the highlighted rectangle (the top left corner). Note the chessboard pattern due to the Color Filter Array

a secondary speckle field should not significantly modify the system; but this is not the case: the Color Filter Array does not move with the imaged speckle field, thus, if a pixel, initially under a red filter, moves under a green one, its (apparent) intensity will be approximately doubled. This means that the evaluation of the error function (e.g. equation (3)) for a rigid body translation of an odd, integer, number of pixels will not be 0, but a very large number, because no pixel matches the expected value. To have a null error, displacement has to be even.

To support this statement, Fig. 9 top shows the displacement errors observed in the DIC output when processing a set of Raw images. Apart from the amplitude of the bias (scaled by a factor of 100 with respect to the monochrome case) the period of oscillation becomes two pixels, as shown by the marks flagging the location corresponding to integer displacements.<sup>6</sup>

Figure 9 bottom, related to the standard deviation of displacements, confirms the two-pixel period. Note that although a significant increment of the peak standard deviation is observed, the scale factor is smaller than that observed for displacements.

#### CFA: demosaicing (interpolation)

Recently, some authors proposed to interpolate the color channels to reconstruct continuous color fields [32]. Since CFA cameras acquire a single signal per pixel, the interpolated values are then used to estimate luminance as the

weighted mean of the color channels (more on this later). The standard approach uses a bilinear interpolation, but, looking at Fig. 10-left it is apparent that a simpler approach is possible. Taking into account the red channel, it is obvious that the red component at pixel  $E$  can be computed as the averaged value of pixels  $A$  and  $I$ . The same can be done in the horizontal direction (i.e.  $r_B = (r_A + r_C)/2$ , where  $r_k$  is the red component at point  $k$ ); finally, the somewhat more complex point  $F$  requires computation of the mean of the four corner pixels ( $r_F = (r_A + r_C + r_I + r_K)/4$ ).

The same procedure can be used for the blue channel, whereas interpolation of the green component requires only computation of the pixel at the center of the cross  $B-J-E-G$  (i.e.  $g_F = (g_B + g_E + g_G + g_J)/4$ ).

The “minimalistic” approach sketched above cannot be generalized. To use higher order polynomials, a somewhat different procedure is required (Fig. 10-right): taking into account that interpolation algorithms use the value at the center of the pixel, the true dimension of the active area does not matter and for each channel we can assume that pixels are twice as big (still considering the red channel, we are assuming that pixels are not the solid red squares but the dashed ones); thus, point  $a$  is located at  $(0.5, 0.5)$ , point  $b$  at  $(0.5, 1)$  and point  $c$  at  $(0, 0.5)$ . Moving the reference system to the successive quadruple, three more points can be computed and so on.

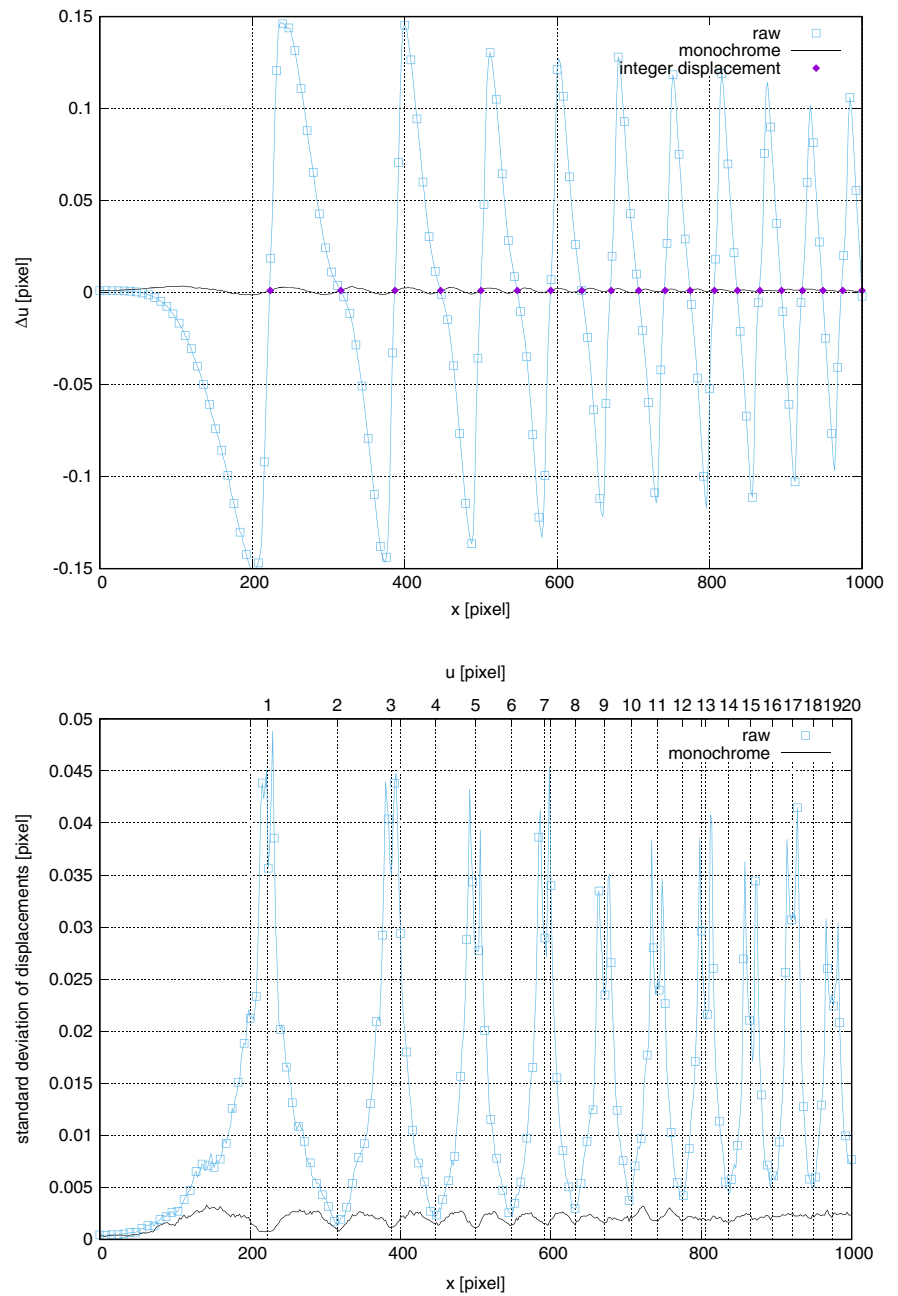
From a practical viewpoint, the proposed algorithm simply requires sampling each channel every two pixels both in the vertical and horizontal directions (thus generating four data matrices half rows by half columns in size); the interpolation can then be performed using the same library functions used by the DIC code for each data plane.

Some points require attention:

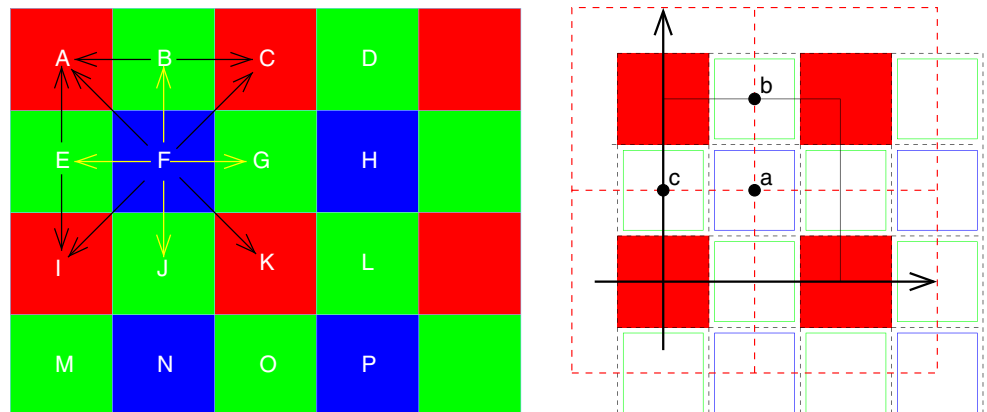
- interpolation is performed on a wider spatial support (twice as large): we thus expect the polynomial bias to have a period of two pixels instead of one;
- processing the green channel is somewhat different from that of the red or blue: comparing the sketch related to the red channel in Fig. 10-right with the general pattern shown on its left, it is apparent that computing interpolation at point  $a$  is not required because there is already a green pixel at that location (if we consider the quadruple  $B, D, J$  and  $L$ , point  $a$  coincides with pixel  $G$ );
- there are two sets of green pixels, the latter shifted one pixel right and one pixel down with respect to the former. As an example, we can consider either the square  $BDJL$  (and successive) or the square  $EGMO$  (and successive). Members of one set never appear in computations related to the other so that, the number of data points involved in the computation of each channel is always the same, i.e. the unbiased computation of

<sup>6</sup>It is to be noted that two bias mechanisms contribute to the total error: the above-described CFA-induced bias, with a 2 pixels period, and the “standard” polynomial bias, as is apparent from Fig. 9 top e.g. the 200–400 pixel range.

**Fig. 9** Displacement error with respect to imposed displacement (top) and standard deviation of displacements (bottom). Raw CFA image with no post-processing ( $\alpha = 20 \cdot 10^{-6}$ ). The solid dots flag the location of integer displacement. Note the period of the bias



**Fig. 10** Bayer Pattern. Left: minimalistic interpolation; right: generalized interpolation

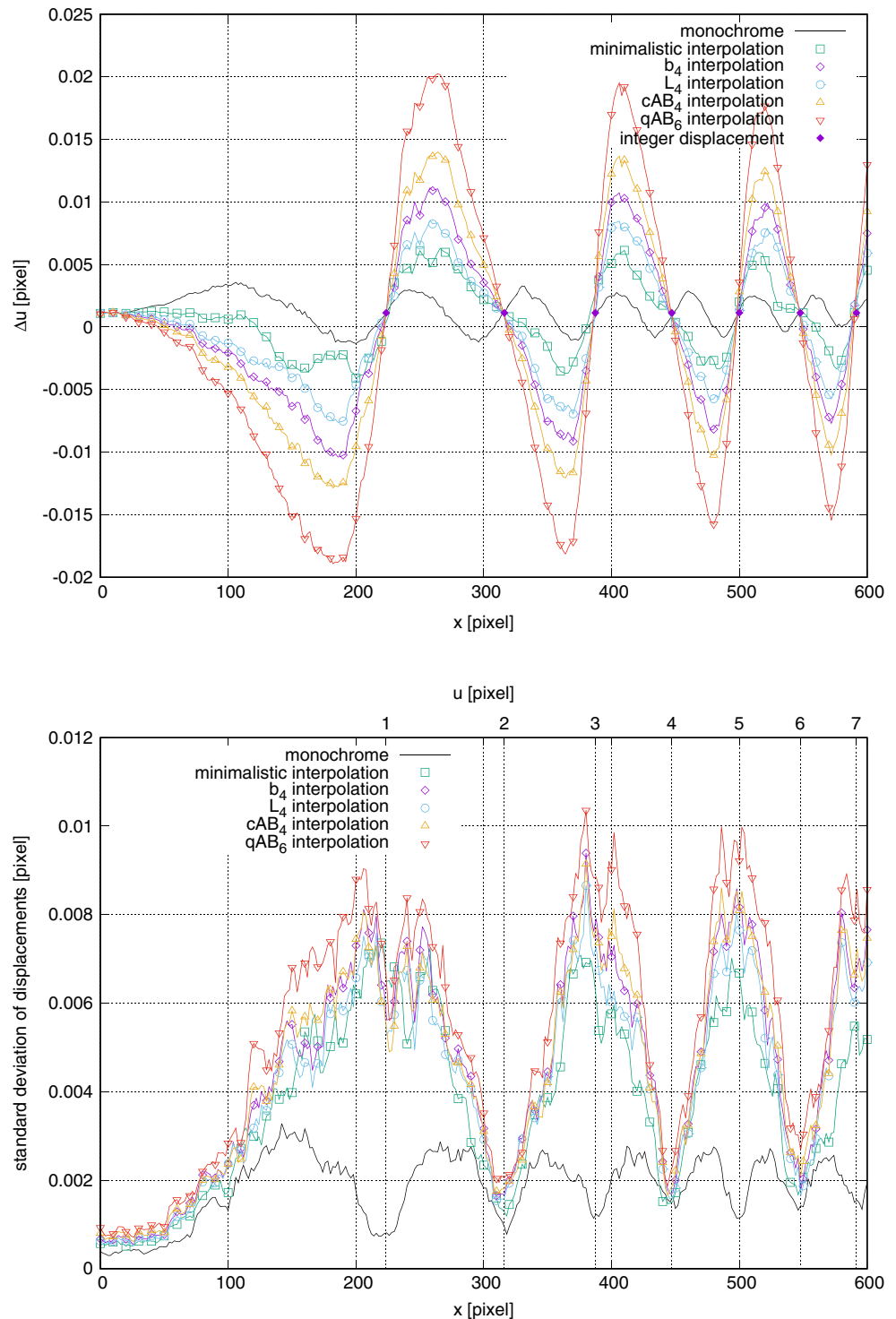


luminance requires the use of the same weight for all four components;

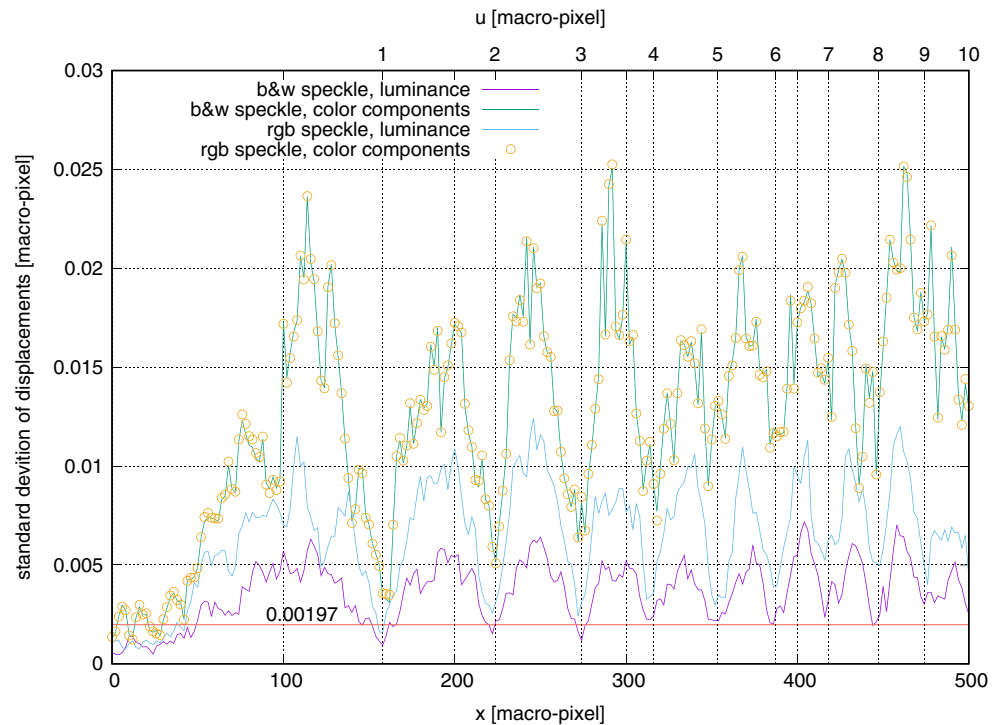
- because of the two green sets, either interpolation at  $b$  or  $c$  is unnecessary: assuming that both sets use the same schema, the missing point is computed by the other one.

Figure 11 top shows the displacement errors related to various interpolating / approximating functions. Results of monochromatic analysis are also shown as reference. Observing the image, we have confirmation of the points discussed above, but some unexpected facts also appear. The period of the bias is doubled as expected, but the amplitude

**Fig. 11** Interpolated CFA image. Top: displacement errors, bottom: standard deviation of displacements. Key:  $b_4$ : bilinear interpolation,  $L_4$ : cubic Lagrange interpolation,  $cAB_4$ : cubic b-spline approximant;  $qAB_6$ : quintic b-spline approximant



**Fig. 12** Standard deviation of displacements. Binned CFA image. The various curves correspond to different processing algorithms. All results refer to macro-pixels, i.e.  $2 \times 2$  cells. The horizontal line is the mean value of standard deviation for monochromatic images



of the errors follow a path that is the reverse of the expected. It is well known that the best interpolating functions for DIC are the bi-cubic and bi-quintic b-spline approximants (respectively  $cAB_4$  and  $qAB_6$ ). Thus, we used them to interpolate the CFA images, together with the bi-cubic Lagrange polynomial, the bilinear interpolation and the minimalistic approach. We expected very good performance from the former and progressively worse performance from the others (note that the DIC code always uses the  $cAB_4$  interpolant during computation). Actually, the results are completely reversed: the use of the minimalistic interpolation for CFA demosaicing induces smaller amplitude errors in DIC computation than does the Lagrange interpolation, which by itself is better than cubic and quintic b-spline approximants.

The standard deviation of displacements (Fig. 11 bottom) confirms these findings: the period of the bell-like curves is two pixels and peak values are significantly larger than the monochromatic-related one. Even though differences are not so large, the minimalistic approach still gives the smaller standard deviation while the worst function is the  $qAB_6$ .

#### CFA: binning

An alternative approach to demosaicing is image binning.<sup>7</sup> The idea is quite simple: whatever the color ordering, a Bayer cell (a  $2 \times 2$  group of pixels) contains two green,

one red and one blue pixel. Thus, the mean of a cell does not depend on the relative sensitivity of the CCD to color components.

From this viewpoint, a Bayer cell is a (large) color sensor, thus, four post-processing paths are possible: the speckle can be either monochromatic or colored; moreover, it is possible either to compute the mean of the “color” components (the luminance) or to process the various color planes using color-DIC (i.e. the color error functional (equation (4))). Note that differently from three-CCD cameras, where all acquired data refer to the same point, in this case the sampling points are different so the color components are not the same even when a black and white speckle field is used.<sup>8</sup>

Figure 12 shows results (standard deviation of displacements) of four DIC analyzes performed using the described approaches. To have a fair comparison, i.e. to involve in the computation the same number of data points as in the monochromatic case, a smaller subset size was used (half width by half height). Moreover, considering the binning operation, we doubled the mean and standard deviation of speckle size during image generation. Using the average of the color components is much faster and gives a

<sup>7</sup>F.Hild. Personal communication.

<sup>8</sup>The resulting images have four color components: red, green, blue and green. Depending on the imaging software, the fourth color component may be completely ignored, used as the alpha channel (the transparency), or the software may refuse to visualize the image. In any case, the error functional (equation (4)) is not limited in any way to three colors.



significantly smaller standard deviation with respect to the use of color-DIC. Note that using a monochromatic or a colored speckle field does not affect results when using the extended formulation (4).

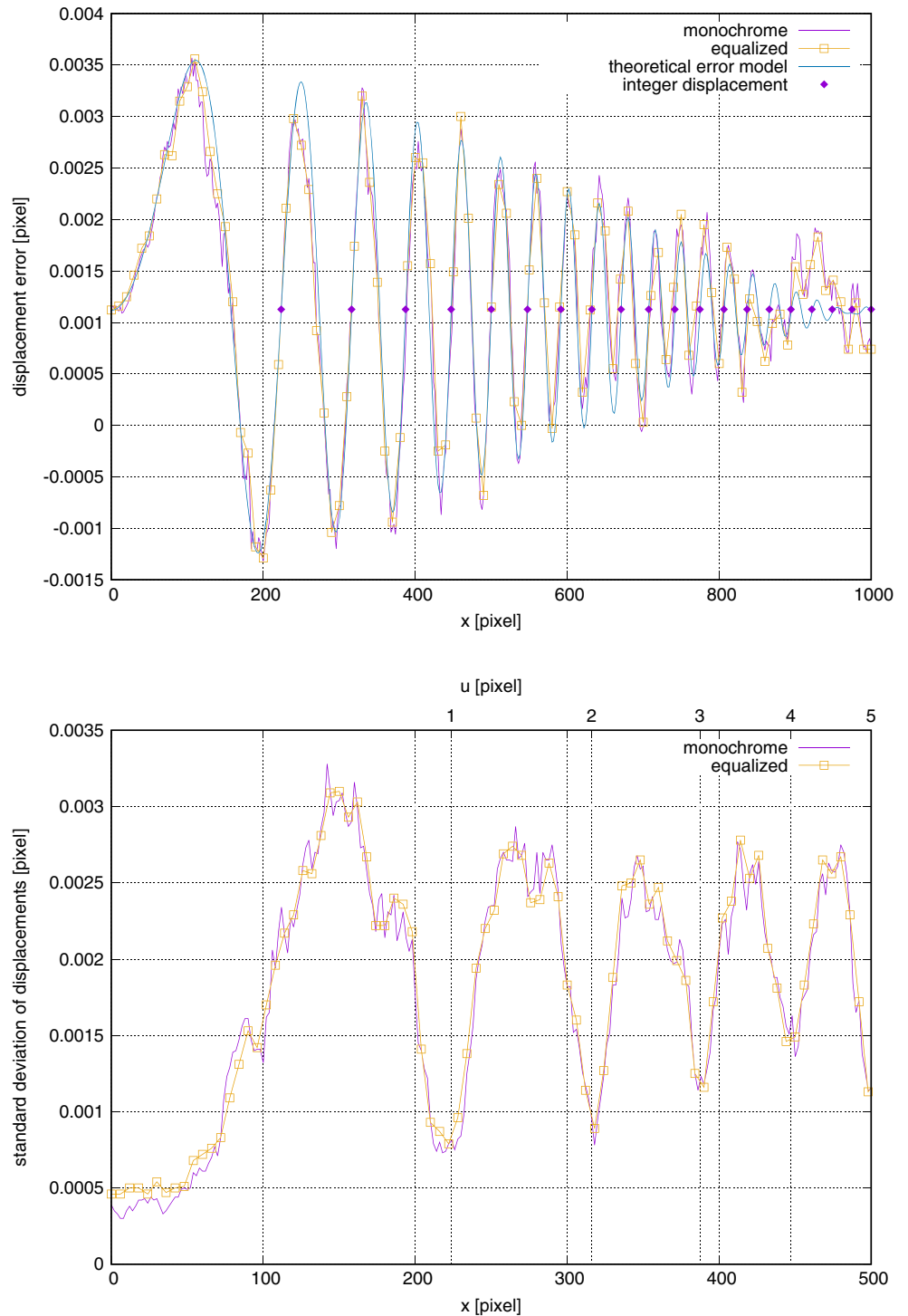
Finally, it is to be noted that the analysis of results has to be performed with care. Reported values refer to the macro-pixels ( $2 \times 2$  pixel clusters) so that, all the

displacement-related values were doubled when performing the comparison.

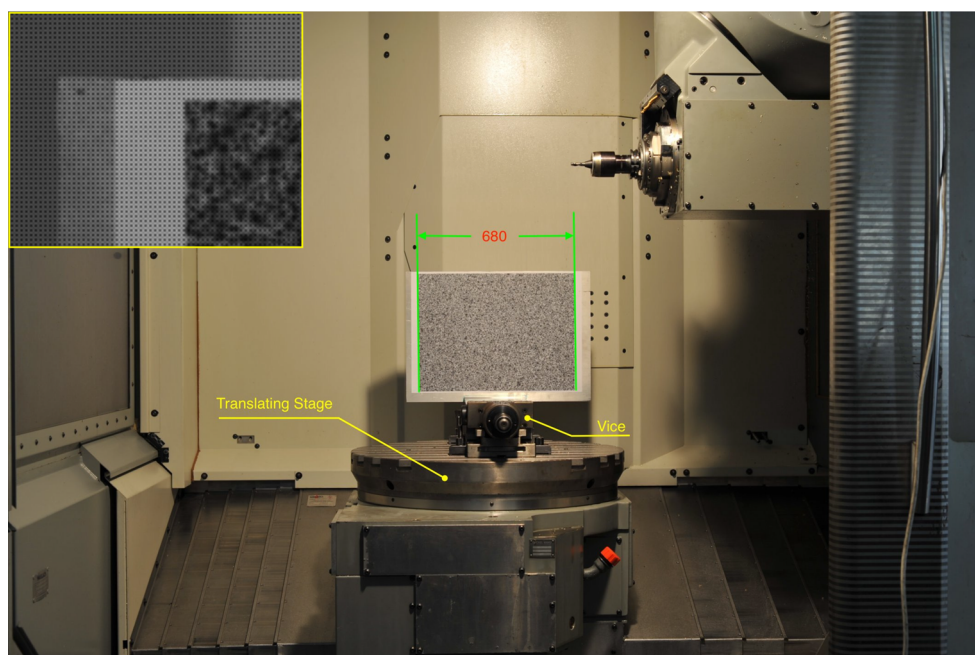
*The egg of Columbus: intensity equalization*

In the previous sections we showed that a CFA image cannot be used as is; however, neither interpolation nor image

**Fig. 13** CFA: intensity equalization. Top: displacement errors; bottom: standard deviation of displacements (note the reduced range of the  $x$  axis). Results are almost exactly the same as with a monochrome camera



**Fig. 14** Experimental setup. The printed speckle image was glued on an aluminum alloy plate. The latter was installed on the horizontal translating axis of the CNC using a high-precision vice. The width of the printed area was 384 mm, corresponding to 680 pixels when imaged, thus, physical size of pixel was 565  $\mu\text{m}$ . In the inset: a magnified view of the raw image (top left corner of the specimen). Note the chessboard pattern due to the CFA



binning appears to solve the problem. Nevertheless, a simple solution exists: let us assume that a black and white speckle field exists on the surface of interest. When the field is imaged using a CFA camera, the averaged value of each channel should be the same, not considering the spectral sensitivity, because the random field is uniformly sampled. Actually, the computed values will differ because of the (generally unknown) sensitivity factors (which depend on camera hardware (potentially known) and on illumination (unknown)).

However, as we know they must be the same, we have only to

1. compute the average  $\bar{r}$ ,  $\bar{g}$  and  $\bar{b}$  of the CFA image (obviously sampling only pixels acquiring the color of interest);
2. scale the red and blue channels respectively by  $\bar{g}/\bar{r}$  and  $\bar{g}/\bar{b}$  (we used the green component as reference because there are twice as many green pixels as there are red or blue ones in the image).

With this simple operation, the chessboard pattern of CFA images (Fig. 8) completely disappears. Note that we performed no interpolation, thus we did not correlate neighbor pixels nor did we filter out signal components. Figure 13 confirms our statement: both the displacement errors and the standard deviation of displacement are almost the same as in the monochromatic case. Obviously, an exact matching is not possible because the noise content of the red and blue pixels is higher than the corresponding ones in a monochromatic image.

## Experimental Validation

To validate the above findings, we performed a simple experimental campaign. We printed one of the speckle images on an A3 paper sheet and glued it on a 15 mm thick aluminum alloy plate. The plate was installed on the horizontal translating stage of a five-axis, numerically controlled milling machine.<sup>9</sup> Aiming to visualize DIC errors, we imaged the specimen using a 50 mm fixed lens:<sup>10</sup> the camera, a Nikon D700, was installed about 3 m from the plate. Owing to the experimental configuration, a large fraction of the active area<sup>11</sup> of the sensor was lost; however, the area imaged in one pixel resulted 564  $\mu\text{m} \times 564 \mu\text{m}$  in size, thus making it easy to perform sub-pixel shifts. Indeed, the horizontal axis of the CNC is equipped with a Heidenhain linear encoder; whose stated resolution is 1  $\mu\text{m}$ , thus ensuring small positioning errors during image acquisition.

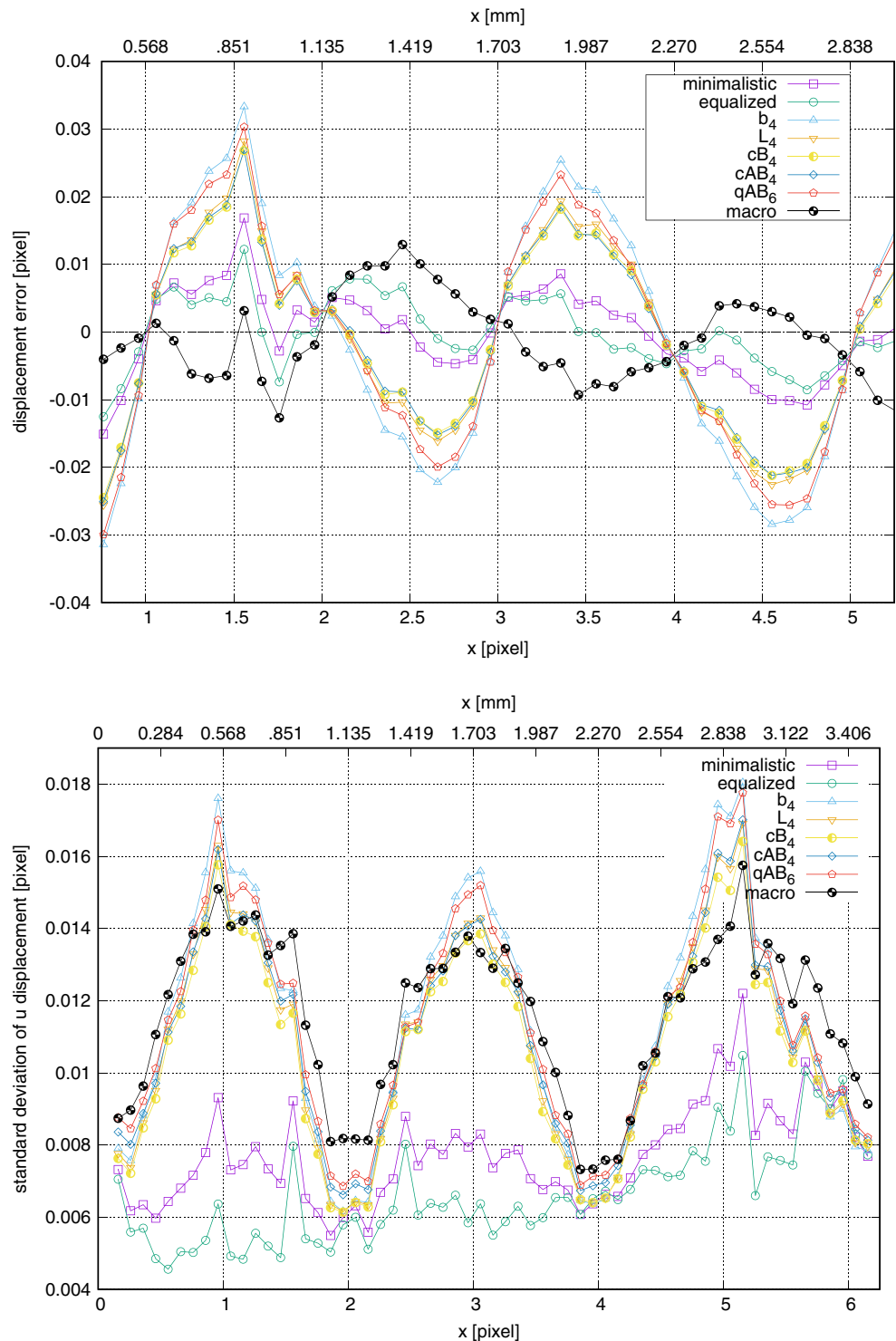
We acquired 62 images, each (apart from the first and second) translated 0.057 mm from the previous one. The camera was set in raw mode and the images were saved as lossless-compressed 14 bit .NEF files. To avoid camera motion due to button pushing and to reduce the influence of vibrations following image acquisition, we used the time-lapse feature of the camera, with a 5 s interval between acquisitions. The full set of images was successively converted to 16 bit TIFF format using *dcraw* in full

<sup>9</sup>Deckel Maho DMU 60 P hi-dyn.

<sup>10</sup>Nikkor AF-S 50 mm f/1.8G.

<sup>11</sup>4256  $\times$  2832 pixel.

**Fig. 15** Experimental images. Top: displacement errors with respect to nominal values using various demosaicing approaches. Bottom: Standard deviation of  $u$  displacements



document mode—no white balance, no gamma compensation, no interpolation—and cropped to preserve the area of interest only.<sup>12</sup> The resulting images were relatively small

<sup>12</sup>Cropping raw images requires some care because the width and height of the cropped area have to be even; in the same way, the  $x$  and  $y$  coordinates of the origin of the active rectangle should be even, otherwise the ordering of the color filters changes.

( $680 \times 506$  pixel), looked quite dark (due to the omission of the gamma correction step) and they obviously showed the characteristic chessboard pattern (Fig. 14: D700 uses RG-GB ordering) thus, they cannot be used directly, unless large errors are considered acceptable (experimental behavior is exactly the same as in the simulation shown in “Bare CFA image”, including the characteristic CFA bias).

**Table 1** Summary of processing options and results

Camera	Speckle	Preprocessing	DIC	Result	Notes
3CCD	BW	luminance	mono	ok	underused
	BW	–	color	ok	slow ( $3 \times t$ )
	RGB	luminance	mono	ok	underused
	RGB	–	color	<b>best</b>	( $3 \times t$ ), best accuracy
CFA	RGB	interp. + lum.	mono	bad	
		interpolation	color	bad	slow ( $3 \times t$ )
		binning + lum.	mono	bad	use BW speckle
		binning	color	bad	( $3 \times t$ ), use BW speckle
CFA	BW	–	mono	<b>worse</b>	raw: $10 \times$ err.
		interp. + lum.	mono	ok	mediocre
		min. + lum.	mono	ok	robust, fast
		binning + lum.	mono	ok	best binning
		binning	color	bad	slow, 2 <sup>nd</sup> binning
		equalized	mono	ok	best CFA

Reference configuration for both execution time and accuracy is the monochrome analysis

Starting from the above-described set of raw images, we generated eight different sets of DIC-friendly images, related respectively to minimalistic interpolation, bilinear interpolation, cubic Lagrange interpolation, cubic B-Spline interpolation, cubic B-Spline approximation, quintic B-Spline approximation, image binning and image equalization.

Figure 15 shows the  $u$  displacement errors (top) and the standard deviation of  $u$  displacements (bottom) resulting from the comparison of DIC-estimated displacements with respect to the expected values for all the image sets. The results substantially confirm what was observed in the simulations: cubic or higher-order interpolation kernels are detrimental both in terms of displacement errors and standard deviation of displacements.

In the interpolation sub-class, the best algorithm is the minimalistic one, probably because of the smaller spatial span; image binning is competitive in terms of mean displacement error, but shows a substantially larger standard deviation of displacements with respect to the best algorithms. Image equalization gives the best results: its displacement errors are not significantly better than the minimalistic or binning methods (Fig. 15 top) but its standard deviation is by far the lower one (Fig. 15 bottom). To be noted is that these results were obtained even though the intensity equalization was not perfect, as clearly flagged by the two-pixel period of the bias.

One more point needs to be observed: using numerically generated images, bilinear interpolation (code  $b_4$ ) appears to be better than cubic and quintic kernels (Fig. 11). This is not confirmed by the experimental results (Fig. 15), where the  $b_4$  kernel gives the worst performance.

## Discussion and Conclusions

In this work, the use of color cameras for DIC is discussed in this work. The combination of three-CCD or CFA cameras, the preprocessing step and the selection of DIC engine allow several processing options, summarized in Table 1. The same table also provides some hints on usage and expected performance based on both numerical and experimental results.

Starting from CFA cameras, it is shown that their use requires some care: indeed, employing either the raw camera data or “standard” image processing (i.e. the software supplied with the camera) gives unreliable results. In particular, the former solution induces a peculiar oscillating bias, with a two-pixel period and very large amplitude.

Various approaches were tested to solve this problem. In particular, a general scheme allowing for use of a standard interpolating function was developed. Different interpolating/approximating kernels were tested; results were somewhat unexpected since polynomial functions, well known for their reliability and accuracy in the DIC field (e.g. bicubic and biquintic B-Spline approximant), do not provide significant advantages. As a general pattern, lower order functions appear to perform better than higher order ones; between the same degree functions, interpolant polynomials overcome approximant ones. The controlling parameter appears to be the locality of the function: indeed, the minimalistic (linear) interpolant discussed in the article was the top performing algorithm and involved as few pixels as possible in the computations, whereas the standard bilinear function ( $b_4$ ) gave poorer results. The pattern sketched above results from the analysis of synthetic images, but is





substantially confirmed by the experimental test we performed. The only notable exception was the aforementioned  $b_4$  polynomial, whose performance worsened significantly (worse than the cubic B-Spline interpolant,  $cB_4$ ).

An alternative to interpolation is image binning. Because the color filters are organized as a repeating  $2 \times 2$  structure, each Bayer cell contains the same sensors and possesses identical sensitivity both to the luminance and chrominance components. Thus, a  $2 \times 2$  block of pixels can be viewed as a four-color<sup>13</sup> macro-sensor (we call them macro-pixels), allowing for either color or monochrome processing. Our tests show that computing the mean of each quadruple (i.e. using the result of standard image binning of the raw data) is faster and more accurate than processing the color components separately. DIC errors and standard deviations of displacements are comparable to what can be obtained using a monochrome camera, but displacements are (obviously) halved. Thus, when doubled, both displacement errors and standard deviation worsen significantly; in particular, the latter becomes similar to interpolation.

A significant step forward can be obtained using image equalization: under the assumption that we are sampling a random field uniformly, the mean intensity of the various color components should be the same. This is obviously not the case because of the different sensitivity of the CCD to the spectral components. However, this suggests a simple processing algorithm: by scaling the red and blue pixels respectively by the green-to-red and green-to-blue ratios the image becomes “almost” monochromatic (apart from a somewhat higher noise component in the red and blue pixels). Results of simulations confirm the above hypothesis and both displacement error and standard deviation of displacements behave as monochromatic images (i.e., they show a one-pixel period). Experimental findings are not so successful, because the period of the bias is two pixels, but both displacement errors and standard deviation are by far the best of the tested algorithms.

Moving to three-CCD cameras (or similar), in the previous sections it was shown that their use is not problematic and can even be advantageous: indeed, in the worst case (black and white speckle combined with color-DIC processing) results are the same as with monochrome cameras, while they can be significantly better if a fully color-enabled processing pipeline is used (i.e. RGB speckle combined with RGB processing). However, they are not “consumer” cameras and most of their advantages can be obtained using a larger-sensor (i.e. with three times the number of pixels or more) lower-cost CFA camera.

A final warning on intensity equalization: the proposed procedure may produce incorrect results in the case of over-exposed acquisition: due to their higher sensitivity, the green pixels saturate long before the red and blue ones. Thus  $\bar{g}$  will be limited by saturation whereas  $\bar{r}$  and  $\bar{b}$  will not. The resulting correction will obviously be erroneous.

**Acknowledgements** The author wishes to thank Mr. Gianluca Marongiu and Mr. Daniele Lai for their support during data acquisition.

## Appendix: Theoretical Model of DIC Displacement Error

Linear shape functions are unable to describe the parabolic displacement function used in this work. Thus we expect an undermatching error. Considering that the displacement field is monodimensional, it is relatively simple to compute the best linear interpolant: the least squares error can be written as

$$\epsilon = \int_{\xi-s}^{\xi+s} (u - N)^2 dx = \int_{\xi-s}^{\xi+s} [\alpha x^2 - (a + bx)]^2 dx \quad (5)$$

where  $u = \alpha x^2$  is the assumed displacement function,  $N = a + bx$  is the interpolating function used by DIC for the local approximation,  $\xi$  is the center of the subset and  $s$  is half its size. Computing the integral, equation (5) becomes,

$$\begin{aligned} \epsilon = & a^2 + b^2 \left( \frac{s^2}{3} + \xi^2 \right) - 2b\alpha\xi(s^2 + \xi^2) \\ & + \frac{1}{5}\alpha^2(s^4 + 10s^2\xi^2 + 5\xi^4) \\ & - \frac{2}{3}a \left[ \alpha(s^2 + 3\xi^2) - 3b\xi \right] \end{aligned} \quad (6)$$

Minimizing equation (6) with respect to the parameters  $a$  and  $b$ , we obtain two equations:

$$\begin{aligned} \frac{\partial \epsilon}{\partial a} = 0 & \rightarrow 3[a + \xi(b - \alpha\xi)] = s^2\alpha \\ \frac{\partial \epsilon}{\partial b} = 0 & \rightarrow b(s^2 + 3\xi^2) = 3\xi[\alpha(s^2 + \xi^2) - a] \end{aligned} \quad (7)$$

Solving equation (7) for  $a$  and  $b$ , we obtain the parameters of the DIC shape function:

$$a = \frac{1}{3}(s^2\alpha - 3\alpha\xi^2) \quad b = 2\alpha\xi$$

i.e.  $b$  is  $\partial u / \partial x$  at  $\xi$  and  $a$  is the inverse of the displacement plus a constant offset ( $\alpha s^2 / 3$ ). Thus, the DIC result will be biased; to give an example, when  $\alpha = 20 \cdot 10^{-6}$  ( $60 \cdot 10^{-6}$ ,  $100 \cdot 10^{-6}$ ) and  $s = 13$  pixel, the expected offset at  $\xi = 0$  is  $a = 0.0011$  pixel (0.0034, 0.0056), i.e. exactly the values observed in Fig. 5.

Apart from the undermatching error, DIC is also affected by polynomial bias, i.e. a sinusoidal-like term, which is null

<sup>13</sup>The second green pixel is sometimes replaced by a gray pixel (i.e. there is no color filter in front of it) and is used as a “luminance” meter.



for integer displacement and maximum for fractional displacements equal to  $\frac{1}{4}$  and  $\frac{3}{4}$ . Assuming the error can be described by a sinusoidal signal,  $b_p = \gamma \sin[2\pi u(x)]$  with  $\gamma$  the amplitude of the oscillation,<sup>14</sup> its contribution  $\eta$  to DIC results can be estimated by integrating  $b_p$  over the subset length

$$\eta = \frac{1}{2s} \int_{\xi-s}^{\xi+s} b_p dx = \gamma \frac{S[2\sqrt{\alpha}(\xi+s)] - S[2\sqrt{\alpha}(\xi-s)]}{4s\sqrt{\alpha}} \quad (8)$$

where  $S(x)$  is the Fresnel integral defined as

$$S(x) = \int_0^x \sin\left(\frac{\pi t^2}{2}\right) dt$$

Summarizing, the expected result of DIC code at  $\xi$  is

$$u_{th} = a + b\xi + \eta = \frac{1}{3}(s^2\alpha - 3\alpha\xi^2) + 2\alpha\xi^2 + \gamma \frac{S[2\sqrt{\alpha}(\xi+s)] - S[2\sqrt{\alpha}(\xi-s)]}{4s\sqrt{\alpha}}$$

thus, the expected error is

$$e_{th} = \frac{\alpha s^2}{3} + \eta = \frac{\alpha s^2}{3} + \gamma \frac{S[2\sqrt{\alpha}(\xi+s)] - S[2\sqrt{\alpha}(\xi-s)]}{4s\sqrt{\alpha}}$$

which exactly describes the observed DIC behavior, providing the subset size is significantly smaller than the polynomial bias period.

## References

- Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: Proceedings of imaging understanding workshop, vol 130, pp 121–130
- Horn BKP (1986) Robot vision. MIT Press
- Sutton MA, Orteu J-J, Schreier H (2009) Image correlation for shape, motion and deformation measurements: basic concepts, theory and applications. Springer, NY, USA. ISBN 978-0-387-78746-6. <https://doi.org/10.1007/978-0-387-78747-3>
- Del Bimbo A, Nesi P, Sanz JLC (1995) Analysis of optical flow constraints, vol 4. ISSN 1057-7149
- Pan B, Xie H, Wang Z (2010) Equivalence of digital image correlation criteria for pattern matching. Appl Opt 49(28):5501–5509
- Schreier HW, Braasch JR, Sutton MA (2000) Systematic errors in digital image correlation caused by intensity interpolation. Opt Eng 39:2915–2921
- Robinson D, Milanfar P (2003) Bias-minimizing filters for motion estimation. In: Conference record of the thirty-seventh asilomar conference on signals, systems and computers, 2003, vol 1
- Reu PL (2011) Experimental and numerical methods for exact subpixel shifting. Exp Mech 51(4):443–452
- Baldi A, Bertolino F (2013) A posteriori compensation of the systematic error due to polynomial interpolation in digital image correlation. Opt Eng 52:101913–1–101913–13. <https://doi.org/10.1117/1.OE.52.10.101913>
- Baldi A, Bertolino F (2015) Experimental analysis of the errors due to polynomial interpolation in digital image correlation. Strain 51(3):248–263. <https://doi.org/10.1111/str.12137>
- Longair MS (2008) Maxwell and the science of colour. Phil Trans R Soc A 366:1685–1696. <https://doi.org/10.1098/rsta.2007.2178>
- Stoffels J, Bluekens AAJ, Peters PJM Color splitting prism assembly, April 1978. US patent 4084180
- Slagle TM, Lyon RF, Ruda MC, Stuhlinger TW Color separation prism with adjustable path lengths, December 11 2001. US patent 6,330,113
- Bayer BE Color imaging array, July 1976. US patent 3,971,065
- Young T (1802) The bakerian lecture: on the theory of light and colours. Philos Trans R Soc Lond 92:12–48. <https://doi.org/10.1098/rstl.1802.0004>
- Dinh NV, Hassan GM, Dyskin AV, MacNish C (2015) Digital image correlation for small strain measurement in deformable solids and geomechanical structures. In: IEEE international conference on image processing (ICIP), 2015. IEEE, pp 3324–3328
- Hang D, Hassan GM, MacNish C, Dyskin A (2016) Characteristics of color digital image correlation for deformation measurement in geomechanical structures. In: 2016 international conference on digital image computing: techniques and applications (DICTA), pp 1–8. <https://doi.org/10.1109/DICTA.2016.7797018>
- Koljonen J, Alander JT (2008) Deformation image generation for testing a strain measurement algorithm. Opt Eng 47(10):107202–107202
- Lava P, Cooreman S, Coppieters S, De Strycker M, Debruyne D (2009) Assessment of measuring errors in dic using deformation fields generated by plastic fea. Opt Lasers Eng 47(7):747–753
- Wattrisse B, Chrysochoos A, Muracciole JM, Nemoz-Gaillard M (2001) Analysis of strain localization during tensile tests by digital image correlation. Exp Mech 41(1):29–39
- Zhou P, Goodson KE (2001) Subpixel displacement and deformation gradient measurement using digital image/speckle correlation (DISC). Opt Eng 40:1613–1620
- Baldi A (2006) Comparative analysis of super-resolution algorithms for digital holography. In: Slangen P, Cerruti C (eds) Speckles, from grains to flowers, vol 6341 of proceedings of SPIE, pp 634114–1–634114–6
- Orteu JJ, Garcia D, Robert L, Bugarin F (2006) A speckle texture image generator. In: Proceedings of SPIE, vol 6341, pp 63410H–1–63410H–6
- Yu L, Pan B (2017) Full-frame, high-speed 3d shape and deformation measurements using stereo-digital image correlation and a single color high-speed camera. Opt Lasers Eng 95:17–25
- Yu L, Pan B (2017) Color stereo-digital image correlation method using a single 3ccd color camera. Exp Mech 57(4):649–657
- Gubbels W, Schajer GS (2015) Three-dimensional digital image correlation using a single color-camera. In: Jin H, Sciammarella C, Yoshida S, Lamberti L (eds) Advancement of optical methods in experimental mechanics, vol 3 of Conference proceedings of the society for experimental mechanics. Springer. [https://doi.org/10.1007/978-3-319-06986-9\\_36](https://doi.org/10.1007/978-3-319-06986-9_36)

<sup>14</sup>Computing  $\gamma$  is quite involved and requires knowing the blending functions of the interpolating kernel. The full process is described in Appendix 2 of [10]. The procedure requires the wavelength of the signal (assumed to be monochrome). To this end, we used the mean size of the speckle (5 pixel), thus obtaining  $\gamma = 0.00248$  pixel.



27. Reu P (2013) DIC Challenge. <http://sem.org/dic-challenge/>
28. Schreier HW, Sutton MA (2002) Systematic errors in digital image correlation due to undermatched subset shape functions. *Exp Mech* 42(3):303–310. <https://doi.org/10.1007/BF02410987>. ISSN 0014-4851
29. Yoneyama S, Morimoto Y (2003) Accurate displacement measurement by correlation of colored random patterns. *JSME International Journal, Series A: Solid Mechanics and Material Engineering* 46(2):178–184
30. Li X, Gunturk B, Zhang L (2008) Image demosaicing: a systematic survey. In: *Electronic imaging 2008*. International Society for Optics and Photonics, pp 68221J–68221J
31. Zapryanov G, Nikolova I (2008) Comparative study of demosaicing algorithms for bayer and pseudo-random bayer color filter arrays. In: *International scientific conference on computer science*
32. Forsey A, Gungor S (2016) Demosaicing images from colour cameras for digital image correlation. *Opt Lasers Eng* 86:20–28. <https://doi.org/10.1016/j.optlaseng.2016.05.006>. ISSN 0143-8166

