



HAL
open science

An FPGA-based pipeline for micropolarizer array imaging

Pierre-Jean Lapray, Luc Gendre, Alban Foulonneau, Laurent Bigue

► **To cite this version:**

Pierre-Jean Lapray, Luc Gendre, Alban Foulonneau, Laurent Bigue. An FPGA-based pipeline for micropolarizer array imaging. *International Journal of Circuit Theory and Applications*, 2018, Computational Image Sensors and Smart Camera Hardware, 46 (9), pp.1675-1689. 10.1002/cta.2477 . hal-03500188

HAL Id: hal-03500188

<https://hal.science/hal-03500188>

Submitted on 21 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An FPGA-based pipeline for Micro-Polarizer Array imaging

Pierre-Jean Lapray, Luc Gendre, Alban Foulonneau, and Laurent Bigué

IRIMAS institute, 12 rue des frères lumière, 68093 Cedex Mulhouse, France

Abstract

The enhancement of current camera performances, in terms of framerate, image resolution and pixel width, has direct consequences on the amount of resources needed to process video data. Stokes imaging permits to estimate polarization of light and create multiple polarization descriptors of the scene. Therefore, such video cameras need fast processing for critical applications like overseeing, defect detection or surface characterization. An FPGA hardware implementation of Stokes processing is presented here that embeds dedicated pipeline for micropolarizer array sensors. An optimized fixed-point pipeline is used to compute polarimetric images, i.e. Stokes vector, degree of polarization and angle of polarization. Simulation and experimental studies are done. The hardware design contains parallel processing, low latency and low power and could meet actual real-time and embeddable requirements for smart camera systems.

Keywords: Stokes imaging, micro-polarizer array, hardware implementation, FPGA

1. Introduction

Analyzing the polarization of the light coming directly from a source or scattered by an object, using an efficient polarimeter instrument, has become of great interest. Due to their nature, polarimeters provide information that are not available with conventional imaging systems. It is used for example in astrophysics [1, 2, 3], remote sensing [4], interferometry [5], biomedical applications [6, 7, 8], or nanostructures and metamaterials characterization [9, 10]. Their benefits are growing bigger as the technology allows faster, more detailed, and more precise measurements [11].

Polarization of light is linked to the wave-propagation vector of the electromagnetic waves. Stokes theory [12] is a method for describing polarization properties of light. In this formalism, the polarization is totally described by a four-components vector, called Stokes vector and commonly denoted $S = [s_0 \ s_1 \ s_2 \ s_3]^T$.

Stokes imaging is done by using one imaging sensor (or several sensors, depending on the technology) and several optical elements, like linear polarizers, wave plates or retarders,

*Corresponding author

Email address: pierre-jean.lapray@uha.fr (Pierre-Jean Lapray, Luc Gendre, Alban Foulonneau, and Laurent Bigué)

15 prisms, liquid crystals, etc. Each pixel of the imaging system needs to be processed in
 16 order to bring out, finally, the four components of the Stokes vector. Linear polarimeter
 17 is the class of device that is designed to measure only the first three polarization Stokes
 18 parameters: s_0 , s_1 , and s_2 . These parameters are stored in full resolution images, and are
 19 used to calculate other useful descriptors like degree of linear polarization (*DOLP*) or angles
 20 of linear polarization (*AOLP*).

21 There are different imaging device architectures that allow the polarization to be ana-
 22 lyzed, each of which has its own drawbacks and advantages. A review of recent acquisition
 23 systems for polarimetric imaging is done in Table 1. The same diversity of instruments
 24 exists for multispectral acquisition systems [13]. There are two main methods to acquire
 25 multi-channel polarimetric images: the scanning technique and the snapshot technique. The
 26 scanning technique implies that multiple polarimetric information are acquired successively
 27 in time. Snapshot could give multiple polarization states at the same time and allows for
 28 video acquisition and direct processing/visualization. Nowadays, the snapshot imaging in-
 29 struments have become more and more exploited, especially with the Micro-Polarizer Array
 30 (MPA) device (e.g. the PolarCam from 4D technology [14]), due to its compactness. Polari-
 31 metric imaging using MPA recently gains in maturity to become out-of-the-lab instruments.

32
33
Table 1: Summary of the acquisition methods for passive Stokes imaging.

Method	Recent Work	Full	Compact
Scan (division-of-time)			
Rotatable Retarder & Fixed Polarizer (RRFP)	[15]	✓	✗
One Liquid-Crystal Variable Retarder & fixed linear polarizer (LCVR)	[16, 17, 18, 19]	✓	✓
Two Liquid-Crystal Variable Retarders & fixed linear polarizer (LCVRs)	[20, 21, 22, 23]	✓	[Depend]
Liquid-crystal variable retarder	[17, 18, 19]	✓	✓
Acousto-Optic Tunable Filter (AOTF)	[24]	✓	✗
Snapshot			
Division-of-Amplitude (DoAmP)	[25]	✓	✗
Division-of-Aperture (DoAP)	[26]	✓	✓
Division-of-Focal-Plane and Micro-Polarizer Array (DoFP & MPA)	[27, 28, 29, 30, 31]	[Depend]	✓
Canonical Refraction (CR) / Biaxial Crystal (BC)	[32, 33]	✓	✗
Channeled Imaging Polarimeters (CIP)	[34, 35]	✓	✗

34 The industry is demanding more and more requirements about efficient image process-
 35 ing, low-power and low-cost camera architecture. On this, we can add the emergence of
 36 embedded systems dedicated to applications such as video protection, medical imaging or
 37 driving assistance. This gives operators the ability to make decision faster. Regarding the
 38 enhancements for 20 years in terms of image sensor resolution (e.g. actual 8K format), fram-
 39 erate or dynamic range, the snapshot technique seems adapted but could contain relatively
 40 high throughput of data to process. To reduce the volume of data to be transmitted by
 41 restricting only the information that the user deems relevant, some cameras have the possi-
 42 bility to do image processing in real-time. We deduce that there is a need to have an efficient

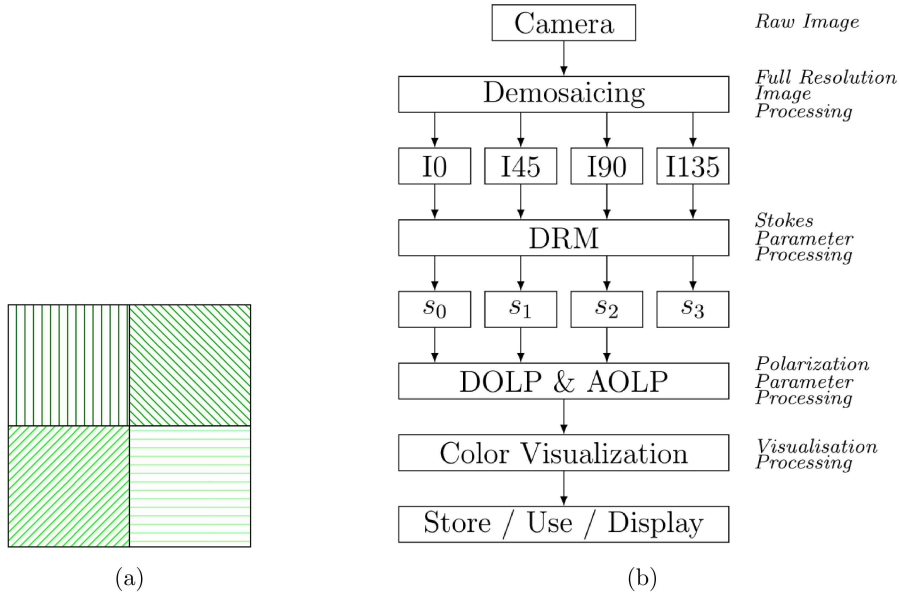


Figure 1: (a) The super pixel spatial arrangement of the MPA considered in this work. The pattern is uniformly repeated over all of the photosensitive cells. (b) Global architecture pipeline. It includes four processing steps.

43 polarimetric imaging pipeline, as it was done for other imaging techniques in the past few
 44 decades, e.g. [36, 37]. We have not found complete and comprehensive works dealing with
 45 Stokes imaging on FPGA; here is the subject of this article.

46 The paper is organized as follows; in Section 2, we start by proposing a Stokes imaging
 47 pipeline dedicated to MPA, that will be embedded in a smart camera. Then, we present the
 48 hardware design of the pipeline in Section 3. Finally, we analyze the efficiency of the solution
 49 by a complete implementation of the pipeline in an FPGA in Section 4 before concluding in
 50 Section 5.

51 2. Stokes imaging pipeline

52 The MPA design that we are considering in the present paper corresponds to the pattern
 53 presented in Figure 1(a). It is composed of pixel size linear polarizers oriented at 0° , 45° ,
 54 90° , 135° , superimposed on a camera sensor chip. Therefore, each pixel measures only one
 55 of the four different intensities, called polarization states, depending on the orientation of the
 56 polarizer in front of the considered pixel. The polarization states are named hereafter I_0 , I_{45} ,
 57 I_{90} , I_{135} . With this setup, a single image acquisition gives a mosaiced image providing partial
 58 spatial information on each of the polarization states simultaneously. A few computation
 59 steps are needed to estimate the incoming polarization at full picture resolution from such an
 60 image. We propose here a pipeline dedicated to MPA. Although we consider a precise MPA
 61 architecture, the whole pipeline can still be applied on other MPA architectures with the
 62 only change of the data reduction matrix (**DRM**) described below, such as MPA that would
 63 allow the circular polarization component to be estimated in the future. This pipeline will

64 then be adapted in an efficient hardware design in Section 3 using VHDL (VHSIC Hardware
 65 Description Language). The pipeline is summarized as a block diagram on Figure 1(b),
 66 which is composed of the following elements:

- 67 • A demosaicing block, composed of an interpolation method to retrieve the full spatial
 68 resolution of the intensity data,
- 69 • A reduction matrix processing, that outputs the Stokes vector parameters in parallel,
- 70 • *DOLP* (Degree Of Linear Polarization) and *AOLP* (Angle Of Linear Polarization)
 71 modules for recovering polarimetric descriptors,
- 72 • A visualization processing block that outputs useful qualitative information, taking
 73 into account the human visual system.

74 Stokes imaging is based on irradiance measurements. So it intrinsically includes all issues
 75 that arise from the standard imaging radiometry domain. If we do not correct for fixed pat-
 76 tern noise (i.e. dark noise and photo response non-uniformity), similar noise consequences
 77 as conventional radiometric imaging could occur. But some recent sensors often have em-
 78 bedded noise corrections within the chip to prevent these effects. Additionally, if no proper
 79 polarimetric calibration is done for the data reduction matrix, variations on transmission
 80 and extinction ratio of the polarimetric elements are not taken into account. Thus the po-
 81 larization descriptors could be miscalculated. Complete calibration of micro-polarizer array
 82 cameras can be found in the literature [38], along with the impact of noise in polarimetric
 83 applications [39]. In the whole pipeline, we assume that images from the MPA camera are
 84 calibrated and do not need pre-processing (i.e. radiometric calibration, linearization, dark
 85 correction, flat-field, etc.).

86 2.1. Estimation from measurements

87 In the current paper, the Stokes vector \mathbf{S} is used to represent the polarization of the
 88 light [12]. There are other possible representations [40] that will not be discussed here.

$$\mathbf{S} = [s_0 \ s_1 \ s_2 \ s_3]^T \quad (1)$$

89 with s_0 the total light intensity, s_1 the intensity difference through a 0° and 90° polarizers,
 90 s_2 the intensity difference through a 45° and -45° polarizers, and s_3 referring to left or right
 91 handedness of the polarized light.

When the light is coming from a source or a surface to a polarimeter, the vector \mathbf{I} that
 represents measured intensities by the sensor can be described as follows:

$$\mathbf{I} = \mathbf{M} \cdot \mathbf{S} \quad (2)$$

where \mathbf{M} is the measurement matrix, defined during system calibration. A Data Reduction
 Matrix (DRM) [41] can be defined for reconstruction of the input signal \mathbf{S} such as:

$$\hat{\mathbf{S}} = \mathbf{DRM} \cdot \mathbf{I} \quad \text{with} \quad \mathbf{DRM} = \mathbf{M}^+ \quad (3)$$

92 where \mathbf{M}^+ is the pseudo-inverse of the measurement matrix.

93 Using Eq. (3), the Stokes vector can be recovered from a set of at least four intensities.
94 Using only linear polarizers in the optical setup will not allow the s_3 component to be
95 estimated [42]¹. We are precisely in that case with the polarimeter system we are considering
96 in this paper, since the MPA is composed of only linear polarizers. Even though the system
97 provides four different polarization states, only the three first Stokes vector elements s_0 , s_1 ,
98 s_2 can be computed. For the rest of the paper, we will only consider polarization descriptors
99 that can be computed from these three elements.

100 2.2. Descriptor computation

101 From the Stokes vector parameters s_0 , s_1 , s_2 , the following quantities can be computed,
102 that help understanding the nature of the polarization.

103
104 The Degree Of Linear Polarization (*DOLP*) represents the amount of linear polarization
105 in the light beam. It takes values between **zero** for non polarized light and **one** for totally
106 polarized light, intermediate values referring to partial polarization.

$$DOLP = \frac{\sqrt{s_1^2 + s_2^2}}{s_0} \quad (4)$$

107 The azimuthal angle of linear polarization (*AOLP*) is also computed from the Stokes
108 vector. It represents the angular orientation of the main axis of the polarization with respect
109 to the chosen angular reference used for system calibration :

$$AOLP = \frac{1}{2} \arctan \left(\frac{s_2}{s_1} \right) \quad (5)$$

110 2.3. Visualization application

111 An interesting application that could be done when performing Stokes imaging is the
112 color visualization of data. It is an application in the sense that the visualization is a
113 direct interpretation of light polarization by the user. It is well known that some insects
114 and animals can have the polarization vision capacities. Bio-inspired techniques to map
115 the polarization signature into a color representation has been widely studied [43, 44]. In
116 this work, we implemented the Tyo *et al.* method [45], that is probably the most common
117 method from the state-of-art. It is based on the HSV (Hue, Saturation, Value) color data
118 fusion that map polarization features to the HSV space as follows:

$$AOLP \rightarrow H \quad DOLP \rightarrow S \quad s_0 \rightarrow V \quad (6)$$

119 Hue is associated with the angle of polarization; the connection between hue and AOLP is
120 the circularity behavior of data. Example of this mapping will be shown in the next section.
121 The main drawback is that a pixel could sense light properties with both low irradiance and
122 high polarization state, but this specificity can't really be represented along this technique,
123 because s_0 is mapped to the image pixel intensity. It is corrected in a recent work [46].

¹In most imaging applications, the circular polarization magnitude is very low.

124 *2.4. Demosaicing*

125 In case of a snapshot camera using MPA with a mosaiced pattern of filters [14], each
 126 pixel has a different instantaneous field of view (IFOV)². In other words, a single pixel only
 127 senses a fraction of the total polarization states, so the other missing polarization states have
 128 to be interpolated. If we compute Stokes parameters without using a spatial interpolation
 129 method among channels, it causes severe artifacts such as zipping or aliasing (especially
 130 when viewing *DOLP*), and makes computer vision algorithms to fail. Due to the regularity
 131 of an MPA filter pattern, it is easy to define convolution kernels applied to each polarization
 132 channel separately. It is well known that bilinear interpolation could avoid a lot of IFOV
 133 problems [47]. Moreover, this is known to be efficient and computationally simple, and thus
 134 could be implemented in real-time. More evolved demosaicing algorithms that are designed
 135 for Color Filter Array (CFA) could not be used directly, because polarimetric imaging does
 136 not have significant correlation among channels when capturing a randomly polarized scene.
 137 We propose to evaluate five kernels and build a choice for the final implementation.

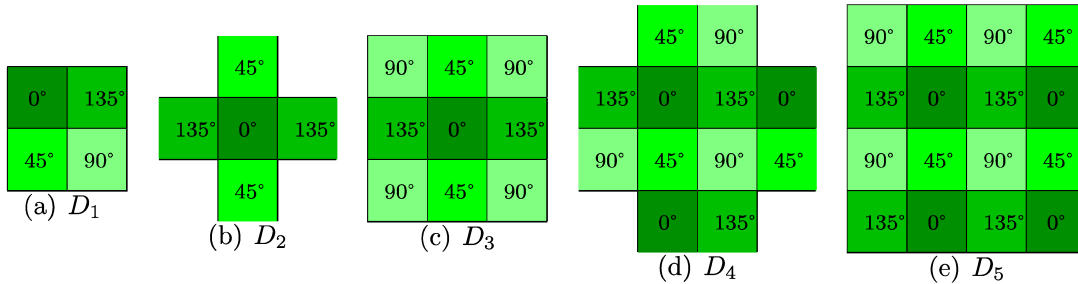


Figure 2: Visualization of the **five** demosaicing kernels D_{1-5} used across the evaluation. It refers to the neighborhood used for interpolation. Each pixel records only I_0 , I_{45} , I_{90} or I_{135} light polarization states.

138 *2.4.1. Demosaicing method evaluation*

139 Here, we are interested in evaluating the five demosaicing kernels and their influence
 140 on the resulting image quality. These methods are described in a recent work by Ratliff *et*
 141 *al.* [47]. Kernels can be visualized in Figure 2. In this past evaluation study [47], only
 142 IFOV artifacts were measured using purely simulated data, and modulation/intermodulation
 143 transfer function as evaluation metrics. To select which methods we should use for any
 144 application, we made an evaluation using more quality metrics. We argue that a more
 145 comprehensive assessment using a larger number of metrics is missing, and that the use of
 146 objective and subjective metrics is useful for selecting a demosaicing algorithm. Indeed,
 147 the key of our evaluation is to use well-known and benchmarked metrics that have been
 148 already used for CFA imaging [48], excepted for perceptual color difference metrics, that
 149 is not applicable in our case. We propose to use these four indicators: PSNR (peak signal
 150 to noise ratio), SSIM [49] (Structural SIMilarity), RMSE (root mean squared error) and

²This step could be by-passed in case of having a polarimeter with already full resolution polarization images at its output (using a division-of-aperture polarimeter for example).

151 correlation [50] metrics. PSNR has a clear physical meaning and is commonly used in
 152 computer science for compression and reconstruction evaluation in digital image processing.
 153 Higher score means better image quality. SSIM has a better perceptual matching, where
 154 best image quality is achieved by a score near to one. It is typically a modified MSE metric
 155 where errors are penalized according to their visibility in the image. Perceptual quality
 156 is not straightforward to measure at all, but to our knowledge SSIM tends to be a well
 157 benchmarked method. RMSE defines the square root of average square deviation between
 158 the original and reconstructed image. The cross-correlation criterion (between 0 and 1) gives
 159 similar quality results independently if an offset exists among intensities, where better score
 160 means higher reconstruction quality. These metrics are fully described in [48].

161 According to the application target, some of these metrics could be preferred to select
 162 proper algorithm independently for its signal to noise ratio, its structural similarity or its
 163 better correlation results.

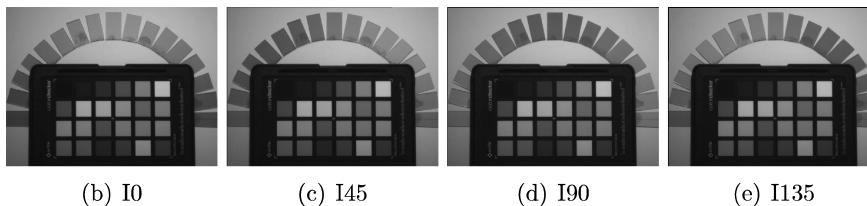
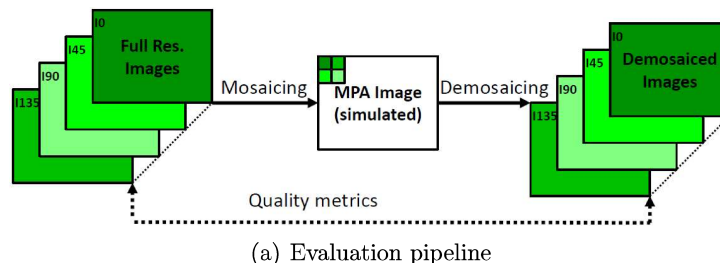


Figure 3: (a) Pipeline for the evaluation of interpolation kernels. (b), (c), (d), (e) Full resolution images used for the demosaicing evaluation. Images were captured using a gray-level sensor and linear polarization filter. The scene is composed of a hand-made polarization chart with pieces of linear polarizers arranged in half circle (polarization axis in the lengthiness of the pieces), and a X-Rite Passport color checker (with patches that are relatively highly diffuse, thus unpolarized).

164 About the methodology: Figure 3(a) presents the pipeline used for evaluation. A set
 165 of images acquired with a gray-level camera was first taken. A linear polarizer in front of
 166 the camera is rotated to 0°, 45°, 90° and 135° using a motion controlled instrument (the
 167 Agilis™Conex-AG-PR100P piezo rotation mount from Newport). The resolution of images
 168 is 1024 × 768 pixels. A tungsten lamp is used for the illuminant. It is assumed that placing
 169 a filter in front of a camera in different positions could cause optical image translation. The
 170 four images are registered using a simple correlation-based registration from the state-of-
 171 art [51].

172 An MPA image could be represented by a mosaiced image with sampled polarization
 173 component. One polarization state is sensed by spatial pixel location. For the simulation,

174 the four full resolution images are combined to simulate an MPA image. The spatial ar-
 175 rangement selected is that of the commercial MPA camera from 4D technology [14]. When
 176 mosaiced image is generated, we apply the five demosaicing kernels D_1 to D_5 . So, we re-
 177 cover 5×4 spatially interpolated images corresponding to the five kernels for each of the
 178 four polarization states. After that, images are compared with the full resolution images
 179 (ground truth) by applying the selected metrics. To be more consistent, we also apply these
 180 verification to all parameters and descriptor images described in Section 2, namely on s_0 ,
 181 s_1 , s_2 , *DOLP*, *AOLP* and HSV (Hue Saturation Value) visualization of polarization.

182 2.4.2. Demosaicing method analysis

183 Visualization of the results are summed up in Figure 4. For an exhaustive visualization
 184 of the results, all image resulting from all methods are shown in the appendix in Figure A.1.
 185 By looking at the reconstructed intensity image s_0 in Figure A.1(a), we can see that D_4 and
 186 D_5 images look blurry, whereas D_{1-3} preserve edges. It could be simply explained by the
 187 fact that the kernels used are larger (4×4 pixels), and that pixel values are estimated using
 188 largest neighborhood. The HSV color visualization in Figures 4(n) to 4(r) is also interesting
 189 because we can see by zooming that all methods feature some color artifacts and chromatic
 190 aberrations that could also appear in CFA images. About D_2 , and by looking at the cross
 191 at the center of the color checker, we can distinguish a lot of zipper effects [52].

192 By looking more particularly at the *DOLP* images in Figures 4(i) to 4(m), we see that
 193 the zipper effect is very pronounced for kernel D_1 and D_2 and is the least marked for kernel
 194 D_4 and D_5 . Hence we verify the fact that D_4 gives the best results concerning the removing
 195 of IFOV artifacts according to [47], even in *AOLP*. Kernel D_5 is not giving the best results
 196 because it intrinsically contains a symmetric structure in the kernel (see Figure 2), whereas
 197 D_4 breaks this symmetry by removing the corner pixel factors in the filter processing.

198 The quantitative evaluation results are presented in Table 2. We find that all *AOLP*
 199 images have very bad scores. This is due to the fact that the arc tangent operation is a
 200 circular operation, which can lead to very different values in the case where an angle is
 201 calculated in the part of the image where *DOLP* is very small (see Figures 4(d) and 4(e)).
 202 Globally, the different metrics seem to be correlated; all the results clearly show that D_3 is
 203 the best interpolation method for most images tested and most metrics. Thus we selected
 204 it to be implemented in our design.

205 In applications such as computer vision (e.g. semantic segmentation, image dehazing,
 206 image denoising, etc.), it is important to preserve perfect edge information, thus we will
 207 prefer the method which gives less artifacts. Moreover, applications with natural scenes
 208 containing a lot of moving objects would prefer to use D_4 , because the effects of IFOV
 209 artifacts are often more pronounced in these conditions. In other applications that need
 210 accurate measurements like in machine vision or computer graphics (metallic object defect
 211 detection, diffuse/specular separation, rendering, etc.), we would prefer D_3 .

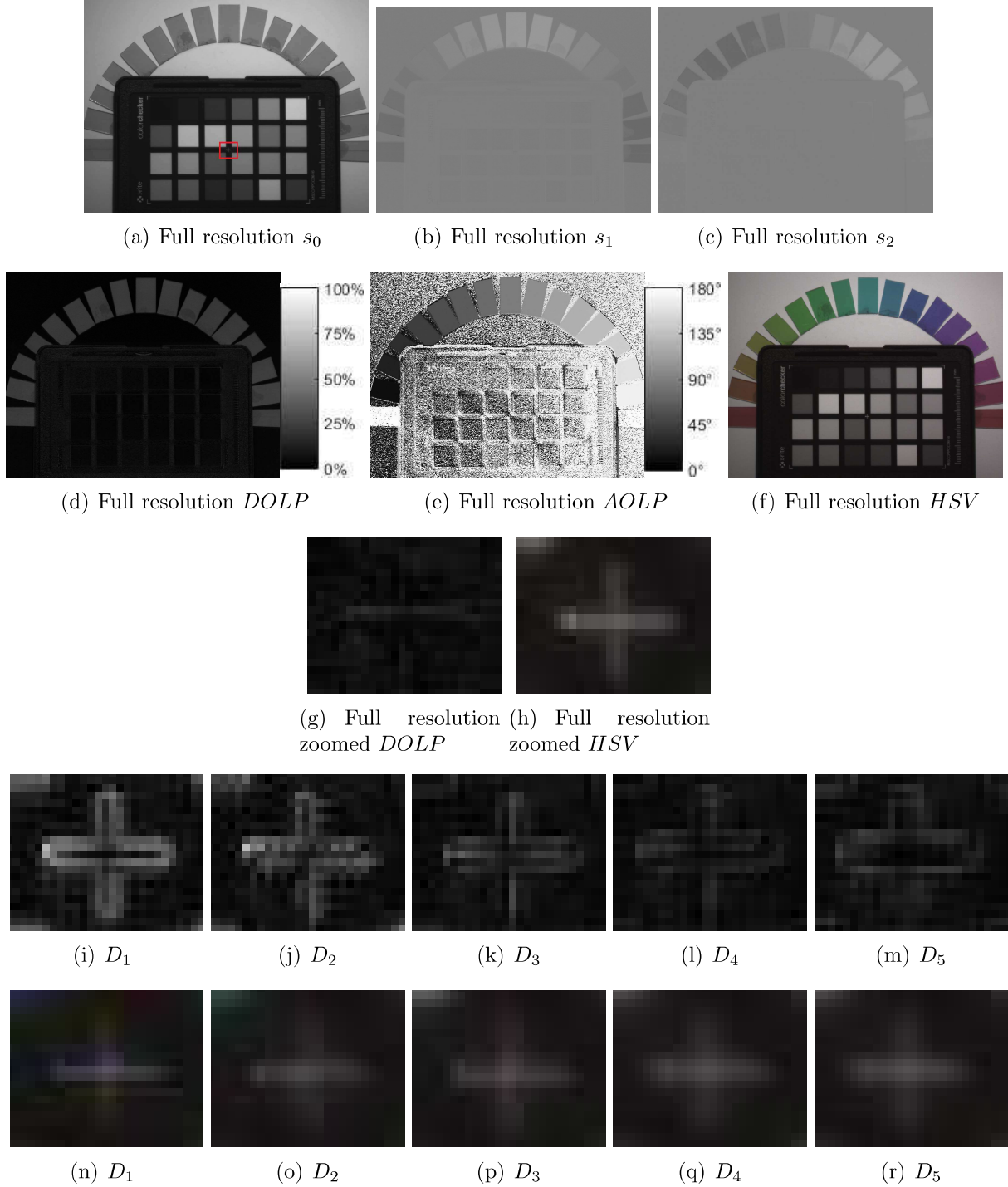


Figure 4: (a-f) Full resolution images used as reference for the demosaicing evaluation. (i-m) Zoomed $DOLP$ demosaicing results. (n-r) Zoomed HSV demosaicing results. Demosaicing is done using the five kernels applied on the test images (shown in Figure 3). The zoomed region corresponds to the white cross at the center of the color checker. We can see zipper effect and different magnitude of IFOV artifacts due to demosaicing method. The full resolution images are shown on Figure A.1.

PSNR	D_1	D_2	D_3	D_4	D_5
I0	35.7	37.9	42.1	37.6	37.1
I45	36.1	38.5	44.0	37.5	36.6
I90	35.5	37.9	43.0	37.0	36.1
I135	35.9	38.3	43.7	38.1	37.5
S0	38.6	40.8	45.2	38.0	37.6
S1	38.6	41.0	45.9	45.9	43.3
S2	39.0	41.5	47.2	46.9	43.9
DOLP	25.6	28.1	33.1	33.6	31.0
AOLP	7.0	7.2	7.3	6.6	6.3
HSVvis	30.4	32.6	36.1	34.0	33.5

SSIM	D_1	D_2	D_3	D_4	D_5
I0	0.96	0.97	0.98	0.97	0.97
I45	0.97	0.98	0.99	0.97	0.97
I90	0.96	0.97	0.99	0.97	0.97
I135	0.96	0.98	0.99	0.98	0.97
S0	0.98	0.98	0.99	0.98	0.97
S1	0.93	0.95	0.98	0.98	0.97
S2	0.93	0.96	0.98	0.99	0.97
DOLP	0.72	0.76	0.84	0.84	0.80
AOLP	0.28	0.30	0.34	0.26	0.23
HSVvis	0.92	0.94	0.97	0.95	0.95

RMSE	D_1	D_2	D_3	D_4	D_5
I0	0.016	0.013	0.008	0.013	0.014
I45	0.016	0.012	0.006	0.013	0.015
I90	0.017	0.013	0.007	0.014	0.016
I135	0.016	0.012	0.007	0.012	0.013
S0	0.012	0.009	0.005	0.013	0.013
S1	0.012	0.009	0.005	0.005	0.007
S2	0.011	0.008	0.004	0.005	0.006
DOLP	0.052	0.039	0.022	0.021	0.028
AOLP	0.447	0.436	0.432	0.466	0.482
HSVvis	0.030	0.024	0.016	0.020	0.021

Corr.	D_1	D_2	D_3	D_4	D_5
I0	1.00	1.00	1.00	1.00	1.00
I45	0.99	1.00	1.00	1.00	1.00
I90	0.99	1.00	1.00	1.00	1.00
I135	1.00	1.00	1.00	1.00	1.00
S0	1.00	1.00	1.00	1.00	1.00
S1	0.87	0.90	0.94	0.96	0.95
S2	0.86	0.90	0.94	0.96	0.94
DOLP	0.63	0.76	0.88	0.90	0.89
AOLP	0.39	0.43	0.47	0.41	0.38
HSVvis	0.99	0.99	0.99	0.99	0.99

Table 2: Demosaicing results for kernels D_{1-5} and the four metrics. Best scores are highlighted in green whereas bad scores in red.

3. Hardware design

3.1. Global architecture

Here we describe the complete hardware architecture that composes our system. It is derived from the pipeline from the previous section, which is shown on Figure 1(b).

3.1.1. Demosaicing

The demosaicing process requires a pixel with the intensities of its neighborhood to estimate the missing intensities. The filtering which is described in VHDL is shown on Figure 5. This work is developed for our particular MPA images containing polarizers arranged as shown on Figure 1(a). It could be extended and adapted to any other MPA filter design (without loss of generality).

We use the 3×3 filtering mask \mathbf{F} described below and sampled channel images $\mathbf{P}_k(\mathbf{I}_{\text{raw}}(i))$, where i indexes the 1-D pixel position in the raw image \mathbf{I}_{raw} , and k indexes the angles of polarization $\{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$. We define the sampling function \mathbf{P}_k , where locations of available channels in a mosaiced image \mathbf{I}_{raw} are sampled as:

$$\mathbf{P}_k(\mathbf{I}_{\text{raw}}(i)) = \begin{cases} \mathbf{I}_{\text{raw}}(i) & \text{if channel } k \text{ is at pixel position } i \text{ in } \mathbf{I}_{\text{raw}} \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

where $k \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$.

Now let us consider the convolution filter [48]:

$$\mathbf{F} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (8)$$

227 We can now compute each channel component $\hat{\mathbf{I}}_{\mathbf{k}}$ using the same convolution filter \mathbf{F} ,
 228 along with the sampled image plane $\mathbf{P}_{\mathbf{k}}$ as this:

$$\hat{\mathbf{I}}_{\mathbf{k}} = \mathbf{F} * \mathbf{P}_{\mathbf{k}}(\mathbf{I}_{\text{raw}}), \quad (9)$$

229 For the hardware design, we need **two** FIFO buffers to store the first **two** image rows,
 230 and **six** shift registers that are responsible for holding the **eight** neighboring pixels for the
 231 current pixel interpolation. The serial connection of the FIFO memories emulates the vertical
 232 displacement of the mask. The transfer of values from the FIFO to the shift registers
 233 emulates the horizontal scrolling. The nine pixels are multiplied by their corresponding
 234 coefficients in \mathbf{F} using nine products. Then, eight accumulators add those pixels. Shift
 235 registers perform single clock delay in order to respect the pipeline timing coherency across
 236 pixels. The output streaming pixels for the corresponding $\mathbf{F} \times \mathbf{P}_{\mathbf{k}}(i)$ is finally transmitted
 237 to the rest of the pipeline.

238 The bilinear filtering processing is applied four times in the hardware design, as we have
 239 to interpolate spatial data for recovering the **four** polarization images $\hat{\mathbf{I}}_{\mathbf{k}}$. The four masks
 240 $\mathbf{P}_{\mathbf{k}}$ are created directly from the input pixel stream $\mathbf{I}_{\text{raw}}(i)$, by multiplexing the channel
 241 intensities. We take one pixel out of two and one line out of two and let other pixels to
 242 zero. It is important to note that this design could be easily adapted to other demosaicing
 243 methods, by changing the \mathbf{F} coefficients, and extending or reducing the neighborhood.

244 3.1.2. DRM

245 Figure 6 shows the VHDL entity of the DRM module. This module is responsible for the
 246 Stokes parameter computation s_{0-3} , as described in Section 2. Inputs are global common
 247 signals (*pixel_clk* and *reset*) and pixel stream $\hat{\mathbf{I}}_{\mathbf{k}}$ from the demosaicing block.

In case of using a sensor that provides directly I_0 , I_{45} , I_{90} and I_{135} , a simplified DRM could be used, as this:

$$\mathbf{DRM} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (10)$$

248 For other sensors that do not provide directly these specific polarization angles, or when
 249 polarizing elements are not considered to be ideals, a calibration step must be done to
 250 recover the proper *DRM* matrix [53] prior to measurements.

251 3.2. Stokes parameters

252 Stokes processing needs the data to be manipulated with decimal numbers. From there,
 253 there are several possibilities. We will have to take into account the precision required for

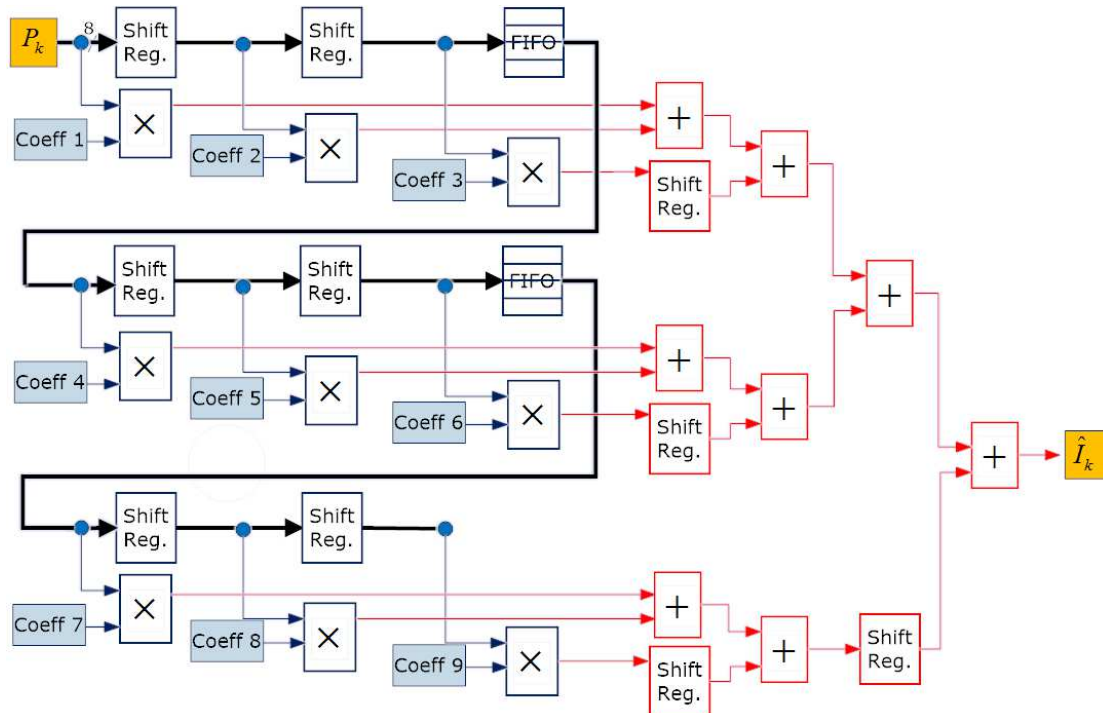


Figure 5: Demosaicing block used in our experiment. It proceeds with a 3×3 window of neighboring pixels. Coefficients are from those of Equation 8 in our hardware implementation.

254 our calculations, to know approximately the range of values that will be used. Fixed-point
 255 and floating-point formats could be considered. The representation of decimal numbers in
 256 the CPU and GPU architecture is underlying and all numbers and manipulation of numbers
 257 are done using single or double precision representations with the IEEE 754 floating-point
 258 standard. We are aware that some new FPGA architectures are coming on the market by
 259 embedding hardware blocks dedicated to floating point computation (e.g. Arria 10 from
 260 Altera). Nevertheless, these devices are very expensive and are still in a niche market. For
 261 a common FPGA architecture, the designer can choose his own mode of representation.
 262 Maximizing the accuracy along with the bit-depth is an optimization procedure, resulting in
 263 low complexity, low power and increasing the maximum operating frequency of the system.

264 *AOLP* and *DOLP* image processing have been described using the IEEE fixed-point
 265 library included in the VHDL 2008 standard. The computation of these components requires
 266 resource consuming and time consuming operators, like divisions (computationally expensive
 267 in hardware real-time design), an arc tangent and a square root computation. For the
 268 division operator, it could not be bypassed, so we use the divider contained in the VHDL
 269 fixed-point library. For the square root and arc tangent implementations, there are three
 270 possible methods :

- 271 1. using CORDIC (COordinate Rotation Digital Computer) algorithm [54],
- 272 2. using a polynomial approximation,
- 273 3. using a customizable LUT.

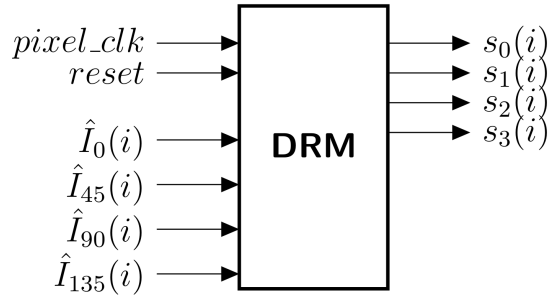


Figure 6: Entity of the Data Reduction Matrix block (DRM). It is the first block dedicated to Stokes processing.

274 The CORDIC algorithm is known to be the most hardware efficient method for the
 275 implementation of trigonometric, hyperbolic and square root equations [55]. It only needs
 276 shift-add handling, **which is** the less time/resource consuming. It avoids additional multipli-
 277 ers and dividers, which are widely used for a polynomial approximation. Cordic is directly
 278 available in FPGA software design tools on the market. The problem could be the big
 279 latency introduced; typically it is 32 clock cycles in our system. With a 125MHz clock,
 280 it corresponds to **0.26 μs** which is very low but could be significant in hard constrained
 281 applications.

282 **If the user wants a very low latency system, a LUT implementation with a one clock**
 283 **cycle per operation would be preferred. This technique consumes a lot of LUT blocks to**
 284 **support the possible input dynamic range of values (e.g. $s_1^2 + s_2^2$ for the square root), and**
 285 **needs bigger FPGA with sufficient LUT resources. In the rest of our work, we choose the**
 286 **Cordic algorithm, as we want to keep the maximum precision, along with low hardware**
 287 **resource utilization, and avoid dividers for the system.**

288 3.3. Fixed-point study

289 A study on how to select the appropriate bit-depth at the expense of image quality is
 290 done. PSNR and SSIM quality metrics are applied on images resulting directly from fixed-
 291 point operations, i.e. *DOLP*, *AOLP* and *HSV* images (see Section 2 for description). As s_0 ,
 292 s_1 and s_2 are integer, it is easy to define the pixel bit-depth required before the radix point.
 293 s_1 and s_2 are varying between -255 and $+255$, whereas s_0 is varying between 0 and 510.
 294 We know that *DOLP* is varying between 0 and 1, so we deduce that the $s_1^2 + s_2^2$ operation
 295 should not have dynamic greater than **260 100**. **That means** that 18 *bits* are necessary for
 296 the integer part to keep the best accuracy.

297 From that point, we could evaluate PSNR and SSIM for the other processed images using
 298 an increasing number of bit after the radix point. Native Matlab fixed-point numeric objects
 299 are constructed and used through the whole processing pipeline. We varied the length of
 300 the decimal part of the numbers, incrementing by 1, starting from an accuracy of 0-bit for
 301 the fraction length, and going up to 32-bit precision. All results are then compared with
 302 the floating-point processing using cast as double type in Matlab. Metrics are then applied
 303 between the fixed-point generated images and double-type processing images. The results of

304 these comparisons are shown in Figure 7. With this method, we could select proper accuracy
 305 of our calculations, depending on the word length and fraction length. For our pipeline and
 306 for the rest of the paper, we selected 14 bits as fractional depth. It is assumed that typical
 307 PSNR values for an 8-bit image and with a relatively good quality, range between 20 and
 308 40dB [56].

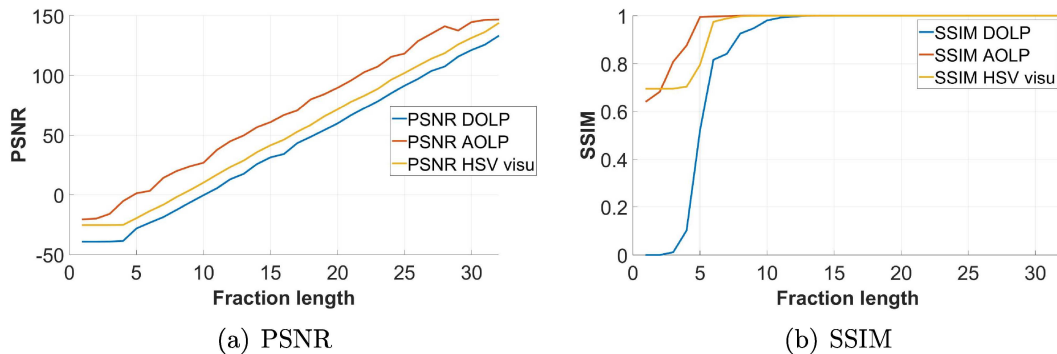


Figure 7: Fixed-point Matlab study results on polarimetric descriptors, by extending the bit-depth of the fixed-point fractional part.

309 3.4. Hardware simulation

310 After describing the pipeline in hardware, simulation is done. The method is based on co-
 311 simulation using Simulink HDL Verifier conjointly with Modelsim Vsim (VHDL simulator)
 312 from Mentor[®]. The simulation environment in Simulink is shown on Figure 8. The mosaiced
 313 image data, the same as in Section 2.4, is sent to the simulator in a streaming manner.
 314 Image data is first arranged as 1-D vector using the frame-to-packet Simulink block. Then,
 315 an unbuffer serializes data at the rate of one pixel per clock tick. The whole VHDL design
 316 is interpreted inside Modelsim and the processed output is hence sending back to Simulink
 317 and all output images are displayed/saved.

318 4. Experimental results

319 In this section, the design is now implemented on an FPGA board and tested with a
 320 video from an MPA sensor.

321 4.1. Implementation

322 Results of the complete implementation of the pipeline design is presented in Table 3. We
 323 implemented the design targeting the Zedboard (xc7z020 Zynq-7000 FPGA) with Xilinx[®]
 324 Vivado tool. This FPGA has a total of 85K programmable logic cells, 4.9Mb of block RAM
 325 and 220 DSP Slices.

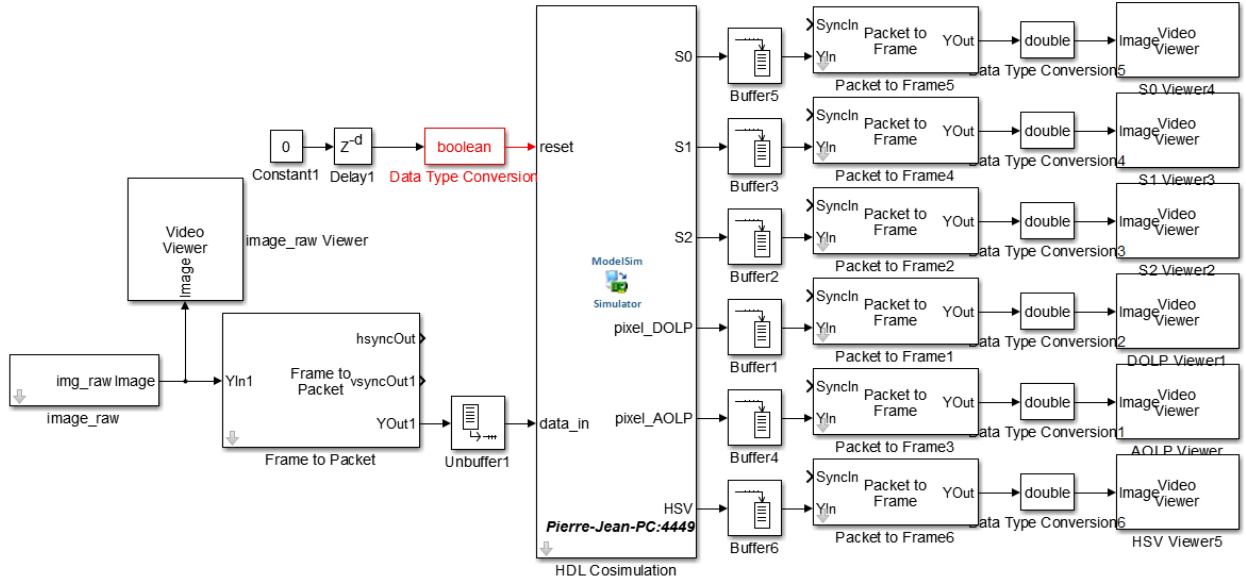


Figure 8: Simulation environment used to simulate the complete pipeline design in Figure 1(b).

326 4.2. Experimental setup

327 Video sample used for the experiment was taken from the PolarCam by 4D technol-
 328 ogy [14]. The full resolution is 648×488 and 8-bit per pixel. We assume that the camera
 329 output is linear and that there is no need to produce additional dark and flat corrections
 330 for using the data. The captured scene is composed of pieces of linear polarizers stuck on a
 331 glass, that are moved by hand in front of the camera.

332 To verify the hardware implementation, Simulink was used along with the FPGA-in-
 333 the-loop (FIL) tool. The FIL tool is a communication interface that sends the streaming
 334 video data to the FPGA via JTAG connection (approximately 13Mbit/s of transferring
 335 bandwidth), and the FPGA sends it back to the CPU after processing. As the FPGA
 336 processes the data faster (125MHz) than the JTAG bandwidth, it contains a *clock enable*
 337 which is synchronized and activated/deactivated depending on the load of the JTAG data
 338 buffer (responsible for transmitting the data). The processed data is then retrieved in the
 339 FIL tool and saved/displayed into Matlab workspace.

340 The video results, showing the outputs of our hardware pipeline, are available online ³.

341 4.3. Discussion

342 Summaries of hardware implementation reports of our design are shown in Table 3 and
 343 4. It appears that *DOLP* and *AOLP* are blocks that consume the most of resources.
 344 This is due to the implementation of CORDIC for the square root and arc tangent. The
 345 demosaicing process consumes 956 slice LUTs for four filtering operations. We compared our
 346 utilization report with the one that would be implemented using a C++-based synthesized

³http://pierrejean.lapray.free.fr/MPA_HW_polarimetry/

Table 3: Detailed report of hardware implementation of the imaging pipeline on the Zynq xc7z020clg484-1.

	Used	Available	Percentage
Power Consumption	0.55 W		
Logic utilization			
Number of Occupied Slices	3,211	13,300	24.1%
Complexity distribution			
Number of Slice registers	7,328	106,400	6.9%
Number of Slice LUTs	9,558	53,200	18.0%
Demosaicing	956		1.8%
DRM processing	225		0.4%
DOLP processing	4,311		8.1%
AOLP processing	4,066		7.6%
Number of DSP	12	220	5.5%
Number of FIFO/BRAMs	4	140	2.9%
Number of DCM-ADVs	1	4	25%

Table 4: Summary of hardware implementation reports on several Xilinx devices for comparison.

FPGA	Artix-7 (xc7a200t)		Kintex-7 (xc7k325t)		Virtex-7 (xc7vx690t)		Zynq (xc7z045)	
Power consumption (W)	0.50		0.51		0.68		0.58	
	Number	Utilization	Number	percentage	Number	Utilization	Number	Utilization
Slices	3,211	9.5%	3,149	6.2%	3,141	2.9%	3,204	5.9%

347 design, i.e. the High Level Synthesis (HLS) tool from Xilinx. We found that four bilinear
 348 filters implemented targeting the same FPGA chip consume 1817 slice LUTs, which is more
 349 compared to our implementation (956 slice LUTs). This is due to the inherent complexity
 350 added (bus and buffer structure around the processing block) by HLS when the design is
 351 synthesized.

352 In terms of performance, pixel latencies are variable depending on blocks. For the de-
 353 mosaicing block, the latency is **two** times the image width plus three, because pixel can
 354 not be computed since enough neighboring pixels are available in buffers. Other processing
 355 latencies are low as each processing block is pipelined. Fixed point limited precision permits
 356 to perform one operation per clock cycle, even for dividers. Respectively, it takes 4, 40 and
 357 39 clock cycles to process *DRM*, *DOLP* and *AOLP*. The color visualization is not time
 358 consuming as it is just a combination of s_0 , *DOLP* and *AOLP* outputs. The total pixel
 359 latency needed is 1343 clock cycles for the 648×488 resolution, that corresponds to $10.74\mu s$
 360 at $125MHz$ in our case. This latency could meet a lot of fast response needs in machine
 361 vision and industry applications.

Table 5: Comparison among the existing state-of-the-art works.

Work	Architecture	Power consumption	Frame processing time	Output
[57]	GPU (GeForce 9400 GS)	$\approx 50W$	$33.6ms$	$S_0, S_1, S_2, \text{DOLP}$
[58]	8-core DSP	$18W$	$17.0ms$	$S_0, S_1, S_2, \text{AOLP, DOLP, HSV}$
[59]	FPGA	$2.45W$	$20.0ms$	$S_0, \text{AOLP, DOLP}$
Ours	FPGA	$0.55W$	$16.6ms$	$S_0, S_1, S_2, \text{AOLP, DOLP, HSV}$

362 All designs tested in Table 3 can process the pixel stream using a maximum frequency of
 363 125MHz (this was the required frequency during place and route steps) without introducing
 364 timing problems, i.e. no negative setup or hold slacks in the paths. So any combination
 365 of image resolution and framerate that could match this maximum streaming pixel clock
 366 constraint is achievable. For example, a 1080p format with a resolution of 1920×1080 at 60
 367 frames per second can be considered, as it needs $1920 \times 1080 \times 60 = 124\,416\,000$ operations
 368 per second to process the streams. We want to point out that due to blank video timing,
 369 processing pixel clock can be different and thus lower than the video pixel clock that is
 370 usually specified in the standard video timing requirements.

371 Table 5 shows the comparison among different state-of-the-art realizations of efficient
 372 Stokes imaging processing. It appears that our work can achieve better performance with
 373 minimal power consumption compared to other state-of-the-art works.

374 5. Conclusion

375 We proposed the design of a Stokes imaging pipeline in FPGA dedicated to MPA. We
 376 validated the processing blocks in hardware simulation using Simulink/Modelsim, and made
 377 studies about fast interpolation methods and fixed-point approximations. We tested the
 378 pipeline in real conditions using a Zynq implementation, and showed different implementa-
 379 tion resource utilization among existing Xilinx FPGAs. The hardware-dedicated pipeline
 380 is capable of processing all Stokes vectors plus numerous already analyzed polarimetric de-
 381 scriptors at an achievable 1080p60 format, and a low fixed latency. The design has a low
 382 hardware complexity, low latency, and the achievable performance is promising for future
 383 high performance embedded cameras and critical applications.

384 As future work, the design will be interfaced with a camera communication protocol,
 385 using the standard interface GigeVision, a framebuffer and a simple streaming interface
 386 (AXI stream or Avalon-stream) bus. Many standard interfaces as Gigevision are not directly
 387 available, and have to be purchased or developed. A straightforward solution would be to
 388 use the system-on-chip FPGA capability of Zynq, which embeds a processor architecture
 389 (a Dual-core ARM Cortex-A9 MPCore) and logic blocks. A Linux driver for interfacing
 390 the GigeVision protocol along with a memory bridge that share data from user-space Linux
 391 memory to the FPGA side would be a solution.

392 **References**

- 393 [1] T. Gehrels, et al., Planets, stars and nebulae: studied with photopolarimetry, Vol. 23, University of
 394 Arizona Press, 1974.
- 395 [2] K. Nagendra, J. Stenflo, Solar polarization, Vol. 243, Springer Science & Business Media, 2013.
- 396 [3] L. Kolokolova, J. Hough, A.-C. Lvasseur-Regourd, Polarimetry of stars and planetary systems, Cam-
 397 bridge University Press, 2015.
- 398 [4] J. S. Tyo, D. L. Goldstein, D. B. Chenault, J. A. Shaw, Review of passive imaging polarimetry for
 399 remote sensing applications, *Applied optics* 45 (22) (2006) 5453–5469.
- 400 [5] M. Novak, J. Millerd, N. Brock, M. North-Morris, J. Hayes, J. Wyant, Analysis of a micropolarizer
 401 array-based simultaneous phase-shifting interferometer, *Appl. Opt.* 44 (32) (2005) 6861–6868. doi:
 402 10.1364/AO.44.006861.
 403 URL <http://ao.osa.org/abstract.cfm?URI=ao-44-32-6861>
- 404 [6] N. Ghosh, I. A. Vitkin, Tissue polarimetry: concepts, challenges, applications, and outlook, *Journal of*
 405 *biomedical optics* 16 (11) (2011) 110801–11080129.
- 406 [7] T. Novikova, A. Pierangelo, A. De Martino, A. Benali, P. Validire, Polarimetric imaging for cancer
 407 diagnosis and staging, *Optics and photonics news* 23 (10) (2012) 26.
- 408 [8] M. Dubreuil, P. Babilotte, L. Martin, D. Sevrain, S. Rivet, Y. Le Grand, G. Le Brun, B. Turlin,
 409 B. Le Jeune, Mueller matrix polarimetry for improved liver fibrosis diagnosis, *Optics letters* 37 (6)
 410 (2012) 1061–1063.
- 411 [9] T. W. Oates, T. Shaykhtudinov, T. Wagner, A. Furchner, K. Hinrichs, Mid-infrared gyrotropy in
 412 split-ring resonators measured by mueller matrix ellipsometry, *Optical Materials Express* 4 (12) (2014)
 413 2646–2655.
- 414 [10] D. Schmidt, Characterization of highly anisotropic three-dimensionally nanostructured surfaces, *Thin*
 415 *Solid Films* 571 (2014) 364–370.
- 416 [11] M. D. Presnar, J. P. Kerekes, Modeling and measurement of optical polarimetric image phenomenology
 417 in a complex urban environment, in: *Geoscience and Remote Sensing Symposium (IGARSS), 2010*
 418 *IEEE International, IEEE, 2010*, pp. 4389–4392.
- 419 [12] G. Stokes, On the composition and resolution of streams of polarized light from different sources,
 420 *Transactions of Cambridge Philosophical Society* 9 (1852) 339–416.
- 421 [13] P.-J. Lapray, X. Wang, J.-B. Thomas, P. Gouton, Multispectral filter arrays: Recent advances and
 422 practical implementation, *Sensors* 14 (11) (2014) 21626–21659. doi:10.3390/s141121626.
 423 URL <http://www.mdpi.com/1424-8220/14/11/21626>
- 424 [14] N. J. Brock, C. Crandall, J. E. Millerd, Snap-shot imaging polarimeter: performance and applications,
 425 Vol. 9099, 2014, pp. 909903–909903–12. doi:10.1117/12.2053917.
 426 URL <http://dx.doi.org/10.1117/12.2053917>
- 427 [15] D. H. Goldstein, Mueller matrix dual-rotating retarder polarimeter, *Appl. Opt.* 31 (31) (1992) 6676–
 428 6683. doi:10.1364/AO.31.006676.
 429 URL <http://ao.osa.org/abstract.cfm?URI=ao-31-31-6676>
- 430 [16] A. Peinado, A. Lizana, J. Campos, Optimization and tolerance analysis of a polarimeter with ferro-
 431 electric liquid crystals, *Appl. Opt.* 52 (23) (2013) 5748–5757. doi:10.1364/AO.52.005748.
 432 URL <http://ao.osa.org/abstract.cfm?URI=ao-52-23-5748>
- 433 [17] L. Gendre, A. Foulonneau, L. Bigué, Full stokes polarimetric imaging using a single ferroelectric liquid
 434 crystal device, *Optical Engineering* 50 (2011) 50 – 50 – 10. doi:10.1117/1.3570665.
 435 URL <http://dx.doi.org/10.1117/1.3570665>
- 436 [18] W. A. Woźniak, M. Pretka, P. Kurzynowski, Imaging stokes polarimeter based on a single liquid crystal
 437 variable retarder, *Appl. Opt.* 54 (20) (2015) 6177–6181. doi:10.1364/AO.54.006177.
 438 URL <http://ao.osa.org/abstract.cfm?URI=ao-54-20-6177>
- 439 [19] J. M. López-Téllez, N. C. Bruce, O. G. Rodríguez-Herrera, Characterization of optical polarization
 440 properties for liquid crystal-based retarders, *Appl. Opt.* 55 (22) (2016) 6025–6033. doi:10.1364/AO.
 441 55.006025.
 442 URL <http://ao.osa.org/abstract.cfm?URI=ao-55-22-6025>

- 443 [20] F. Goudail, P. Terrier, Y. Takakura, L. Bigué, F. Galland, V. DeVlaminck, Target detection with a
444 liquid-crystal-based passive stokes polarimeter, *Appl. Opt.* 43 (2) (2004) 274–282. doi:10.1364/AO.
445 43.000274.
446 URL <http://ao.osa.org/abstract.cfm?URI=ao-43-2-274>
- 447 [21] J. M. Bueno, Polarimetry using liquid-crystal variable retarders: theory and calibration, *Journal of*
448 *Optics A: Pure and Applied Optics* 2 (3) (2000) 216.
- 449 [22] O. Aharon, I. Abdulhalim, Liquid crystal wavelength-independent continuous polarization rotator,
450 *Optical Engineering* 49 (3) (2010) 034002–034002–4. doi:10.1117/1.3366545.
451 URL <http://dx.doi.org/10.1117/1.3366545>
- 452 [23] M. Vedel, S. Breugnot, N. Lechocinski, Full stokes polarization imaging camera, in: *Proc. SPIE*, Vol.
453 8160, 2011, p. 81600X.
- 454 [24] N. Gupta, D. R. Suhre, Acousto-optic tunable filter imaging spectrometer with full stokes polarimetric
455 capability, *Appl. Opt.* 46 (14) (2007) 2632–2637. doi:10.1364/AO.46.002632.
456 URL <http://ao.osa.org/abstract.cfm?URI=ao-46-14-2632>
- 457 [25] E. Compain, B. Drevillon, Broadband division-of-amplitude polarimeter based on uncoated prisms,
458 *Appl. Opt.* 37 (25) (1998) 5938–5944. doi:10.1364/AO.37.005938.
459 URL <http://ao.osa.org/abstract.cfm?URI=ao-37-25-5938>
- 460 [26] T. Mu, C. Zhang, Q. Li, R. Liang, Error analysis of single-snapshot full-stokes division-of-aperture
461 imaging polarimeters, *Opt. Express* 23 (8) (2015) 10822–10835. doi:10.1364/OE.23.010822.
462 URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-23-8-10822>
- 463 [27] J. S. Tyo, Hybrid division of aperture/division of a focal-plane polarimeter for real-time polarization
464 imagery without an instantaneous field-of-view error, *Optics letters* 31 (20) (2006) 2984–2986.
- 465 [28] G. Myhre, W.-L. Hsu, A. Peinado, C. LaCasse, N. Brock, R. A. Chipman, S. Pau, Liquid crystal
466 polymer full-stokes division of focal plane polarimeter, *Opt. Express* 20 (25) (2012) 27393–27409. doi:
467 10.1364/OE.20.027393.
468 URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-20-25-27393>
- 469 [29] K. A. Bachman, J. J. Peltzer, P. D. Flammer, T. E. Furtak, R. T. Collins, R. E. Hollingsworth, Spiral
470 plasmonic nanoantennas as circular polarization transmission filters, *Opt. Express* 20 (2) (2012) 1308–
471 1319. doi:10.1364/OE.20.001308.
472 URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-20-2-1308>
- 473 [30] X. Zhao, A. Bermak, F. Boussaid, V. G. Chigrinov, Liquid-crystal micropolarimeter array for full stokes
474 polarization imaging in visible spectrum, *Opt. Express* 18 (17) (2010) 17776–17787. doi:10.1364/OE.
475 18.017776.
476 URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-18-17-17776>
- 477 [31] W.-L. Hsu, G. Myhre, K. Balakrishnan, N. Brock, M. Ibn-Elhaj, S. Pau, Full-stokes imaging polarimeter
478 using an array of elliptical polarizer, *Opt. Express* 22 (3) (2014) 3063–3074. doi:10.1364/OE.22.
479 003063.
480 URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-22-3-3063>
- 481 [32] A. Peinado, A. Lizana, A. Turpín, C. Iemmi, T. K. Kalkandjiev, J. Mompart, J. Campos, Optimiza-
482 tion, tolerance analysis and implementation of a stokes polarimeter based on the conical refraction
483 phenomenon, *Opt. Express* 23 (5) (2015) 5636–5652. doi:10.1364/OE.23.005636.
484 URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-23-5-5636>
- 485 [33] I. Estévez, V. Sopo, A. Lizana, A. Turpin, J. Campos, Complete snapshot stokes polarimeter based on
486 a single biaxial crystal, *Opt. Lett.* 41 (19) (2016) 4566–4569. doi:10.1364/OL.41.004566.
487 URL <http://ol.osa.org/abstract.cfm?URI=ol-41-19-4566>
- 488 [34] K. Oka, T. Kato, Spectroscopic polarimetry with a channeled spectrum, *Opt. Lett.* 24 (21) (1999)
489 1475–1477. doi:10.1364/OL.24.001475.
490 URL <http://ol.osa.org/abstract.cfm?URI=ol-24-21-1475>
- 491 [35] K. Oka, T. Kaneko, Compact complete imaging polarimeter using birefringent wedge prisms, *Opt.*
492 *Express* 11 (13) (2003) 1510–1519. doi:10.1364/OE.11.001510.
493 URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-11-13-1510>

- 494 [36] P. J. Lapray, B. Heyrman, M. Rossé, D. Ginhac, Hdr-artist: High dynamic range advanced real-time
 495 imaging system, in: 2012 IEEE International Symposium on Circuits and Systems, 2012, pp. 1428–1431.
 496 doi:10.1109/ISCAS.2012.6271513.
- 497 [37] P.-J. Lapray, B. Heyrman, D. Ginhac, Hdr-artist: an adaptive real-time smart camera for high dy-
 498 namic range imaging, *Journal of Real-Time Image Processing* 12 (4) (2016) 747–762. doi:10.1007/
 499 s11554-013-0393-7.
 500 URL <http://dx.doi.org/10.1007/s11554-013-0393-7>
- 501 [38] S. B. Powell, V. Gruev, Calibration methods for division-of-focal-plane polarimeters, *Opt. Express*
 502 21 (18) (2013) 21039–21055. doi:10.1364/OE.21.021039.
 503 URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-21-18-21039>
- 504 [39] F. Goudail, M. Boffety, Fundamental limits of target detection performance in passive polarization
 505 imaging, *J. Opt. Soc. Am. A* 34 (4) (2017) 506–512. doi:10.1364/JOSAA.34.000506.
 506 URL <http://josaa.osa.org/abstract.cfm?URI=josaa-34-4-506>
- 507 [40] D. Goldstein, *Polarized Light*, 2nd Edition, Marcel Dekker, New York & Basel, 2003.
- 508 [41] M. Bass, E. W. Van Stryland, D. R. Williams, W. L. Wolfe, *Handbook of optics*, Vol. 2, McGraw-Hill
 509 New York, 2001.
- 510 [42] J. S. Tyo, Optimum linear combination strategy for an n-channel polarization-sensitive imaging or
 511 vision system, *J. Opt. Soc. Am. A* 15 (2) (1998) 359–366. doi:10.1364/JOSAA.15.000359.
 512 URL <http://josaa.osa.org/abstract.cfm?URI=josaa-15-2-359>
- 513 [43] G. D. Bernard, R. Wehner, Functional similarities between polarization vision and color vision, *Vision*
 514 *Research* 17 (9) (1977) 1019 – 1028. doi:http://dx.doi.org/10.1016/0042-6989(77)90005-0.
 515 URL <http://www.sciencedirect.com/science/article/pii/0042698977900050>
- 516 [44] L. B. Wolff, Polarization camera for computer vision with a beam splitter, *J. Opt. Soc. Am. A* 11 (11)
 517 (1994) 2935–2945. doi:10.1364/JOSAA.11.002935.
 518 URL <http://josaa.osa.org/abstract.cfm?URI=josaa-11-11-2935>
- 519 [45] J. S. Tyo, E. N. Pugh, N. Engheta, Colorimetric representations for use with polarization-difference
 520 imaging of objects in scattering media, *J. Opt. Soc. Am. A* 15 (2) (1998) 367–374. doi:10.1364/
 521 JOSAA.15.000367.
 522 URL <http://josaa.osa.org/abstract.cfm?URI=josaa-15-2-367>
- 523 [46] J. S. Tyo, B. M. Ratliff, A. S. Alenin, Adapting the hsv polarization-color mapping for regions with low
 524 irradiance and high polarization, *Opt. Lett.* 41 (20) (2016) 4759–4762. doi:10.1364/OL.41.004759.
 525 URL <http://ol.osa.org/abstract.cfm?URI=ol-41-20-4759>
- 526 [47] B. M. Ratliff, C. F. LaCasse, J. S. Tyo, Interpolation strategies for reducing ifov artifacts in microgrid
 527 polarimeter imagery, *Opt. Express* 17 (11) (2009) 9112–9125. doi:10.1364/OE.17.009112.
 528 URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-17-11-9112>
- 529 [48] O. Losson, L. Macaire, Y. Yang, Comparison of color demosaicing methods, *Advances in Imaging and*
 530 *Electron Physics* 162 (2010) 173–265. doi:10.1016/S1076-5670(10)62005-8.
 531 URL <https://hal.archives-ouvertes.fr/hal-00683233>
- 532 [49] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility
 533 to structural similarity, *IEEE Transactions on Image Processing* 13 (4) (2004) 600–612. doi:10.1109/
 534 TIP.2003.819861.
- 535 [50] D. Su, P. Willis, Demosaicing of color images using pixel level data-dependent triangulation, in: *Pro-*
 536 *ceedings of Theory and Practice of Computer Graphics, 2003.*, 2003, pp. 16–23. doi:10.1109/TPCG.
 537 2003.1206926.
- 538 [51] G. D. Evangelidis, E. Z. Psarakis, Parametric image alignment using enhanced correlation coefficient
 539 maximization, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (10) (2008) 1858–
 540 1865. doi:10.1109/TPAMI.2008.113.
- 541 [52] B. K. Gunter, J. Glotzbach, Y. Altunbasak, R. W. Schafer, R. M. Mersereau, Demosaicking: color
 542 filter array interpolation, *IEEE Signal Processing Magazine* 22 (1) (2005) 44–54. doi:10.1109/MSP.
 543 2005.1407714.
- 544 [53] J. S. Tyo, D. L. Goldstein, D. B. Chenault, J. A. Shaw, Review of passive imaging polarimetry for

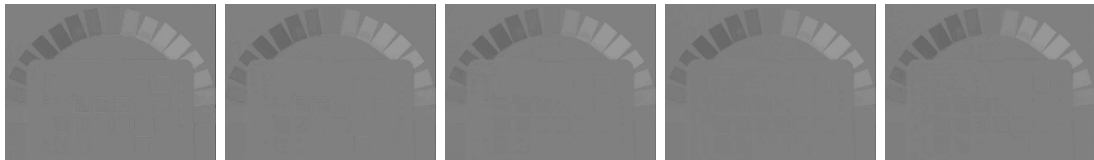
- 545 remote sensing applications, *Appl. Opt.* 45 (22) (2006) 5453–5469. doi:10.1364/AO.45.005453.
546 URL <http://ao.osa.org/abstract.cfm?URI=ao-45-22-5453>
- 547 [54] J. E. Volder, The cordic trigonometric computing technique, *IRE Transactions on Electronic Computers*
548 *EC-8* (3) (1959) 330–334. doi:10.1109/TEC.1959.5222693.
- 549 [55] R. Andraka, A survey of cordic algorithms for fpga based computers, in: *Proceedings of the 1998*
550 *ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays, FPGA '98*, ACM,
551 New York, NY, USA, 1998, pp. 191–200. doi:10.1145/275107.275139.
552 URL <http://doi.acm.org/10.1145/275107.275139>
- 553 [56] D. Salomon, *Data compression: the complete reference*, Springer Science & Business Media, 2004.
- 554 [57] H. Patel, Gpu accelerated real time polarimetric image processing through the use of cuda, in: *Pro-*
555 *ceedings of the IEEE 2010 National Aerospace Electronics Conference, 2010*, pp. 177–180. doi:
556 10.1109/NAECON.2010.5712943.
- 557 [58] K. C.-K. Marcus Bednara, Real time polarization sensor image processing on an embedded fpga/multi-
558 core dsp system, Vol. 9506, 2015, pp. 9506 – 9506 – 13. doi:10.1117/12.2178823.
559 URL <http://dx.doi.org/10.1117/12.2178823>
- 560 [59] V. G. Timothy York, Samuel Powell, A comparison of polarization image processing across different
561 platforms, Vol. 8160, 2011, pp. 8160 – 8160 – 7. doi:10.1117/12.894633.
562 URL <http://dx.doi.org/10.1117/12.894633>



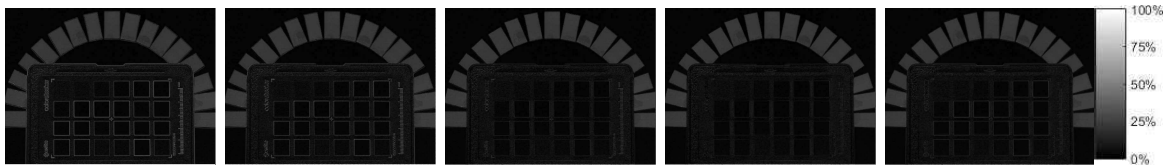
(a) s_0 . The dynamic range $[0; 511]$ is mapped to $[0; 255]$ for visualization.



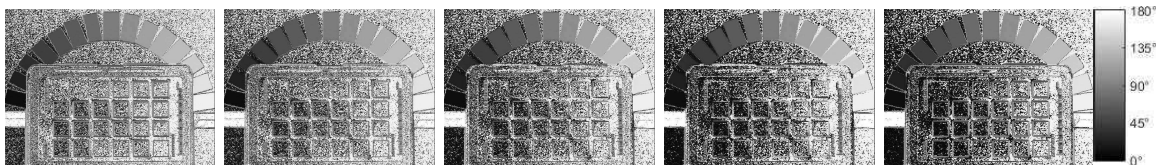
(b) s_1 . The dynamic range $[-255; 255]$ is mapped to $[0; 255]$ for visualization.



(c) s_2 . The dynamic range $[-255; 255]$ is mapped to $[0; 255]$ for visualization.



(d) DOLP



(e) AOLP



(f) HSV visualization

Figure A.1: Demosaicing results using the five kernels applied on the test images (shown in Figure 3). The **five** demosaicing methods D_{1-5} are described in Section 2.4. By zooming numerically on these images, we can see different magnitude of IFOVs artifacts due to demosaicing (especially for DOLP). (f) could be only visualized on the pdf color version of this paper.