IDEAL: An image pre-processing architecture for high-end professional DSC applications

Auke van der Heide, Takashi Urano, Frank Polderdijk, Wim de Haan, Jan T. Bosiers

DALSA Professional Imaging, High Tech Campus 27, 5656 AE Eindhoven, The Netherlands
Auke.van.der.Heide@DALSA.com - phone +31-40- 2599030 – fax +31-40 – 2599005

ABSTRACT

We developed and implemented a flexible image pre-processing concept to achieve an image sub-system for top-end professional digital still camera applications that ensures the highest possible image quality. It supports high-speed multiple-image acquisition and data processing for very-large resolution images, it reduces the design-in time for the customers and it can be implemented economically for high-end applications with relatively smaller volumes.

1. INTRODUCTION

The IDEAL architecture targets the top-end professional digital still camera market, where the image quality is of utmost importance [1], [2]. A few years ago a typical high-end digital still camera (DSC) imager would have a 33Megapixel CCD [3], Fig. 1, that is read out through two outputs, W and X, at 2 x 25MHz; or a similar 39Megapixel CCD [4]. The sensor pixel data are shifted to the left and right output, resulting in two half images of which one is horizontally mirrored. Today there continues to be a drive for higher resolution (>50M), more than 2 sensor outputs and higher acquisition and processing speeds. Typical examples are the recently announced digital backs from Phase One [5], Leaf (Kodak)[6] and Hasselblad [7]. As can be expected this trend is also visible in the consumer and prosumer market. Today a point-and-shoot camera with more than 8Megapixels is quite common. It is easy to imagine that this trend is causing all sorts of technical challenges that the image sensor suppliers and camera manufacturers are facing [8]. An ever-increasing number of pixels have an impact on the sensor yield, image correction algorithms, image capture rate, processing speed, power consumption and memory use.

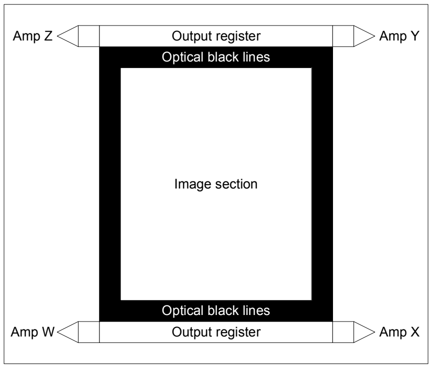


Fig. 1: Typical example of a medium-format sensor output configuration.

Part of these challenges where solved in the Mamiya ZD camera in 2005 [9], which was the first medium format D-SLR camera of a few years back. This camera uses a dual-output 22Megapixel medium format sensor and a dedicated digital preprocessing ASIC that solved several of these challenges. This ASIC stores multiple images in the externally connected SDRAM frame memory for fast image acquisition and applies several correction algorithms such as dark frame subtraction, tap matching, blemish correction and noise reduction.

Our new IDEAL concept is part of a new developed image sub-system that comprises, next to the CCD, a timing generator, the sensor readout clock drivers and an analog front-end IC (for CDS noise suppression, variable gain and AD conversion), see figure 2. The new IDEAL concept adds digital image pre-processing functionality to this sub-system, to deliver ‘perfect’ raw data at high speed to the camera signal processor (DSP). Next to that, the IDEAL concept solves architectural problems in the camera, such as high-speed image acquisition that would have been extremely difficult to solve, especially when the camera signal processor is bandwidth limited with respect to processing speed and access to the solid-state memory device of the camera.

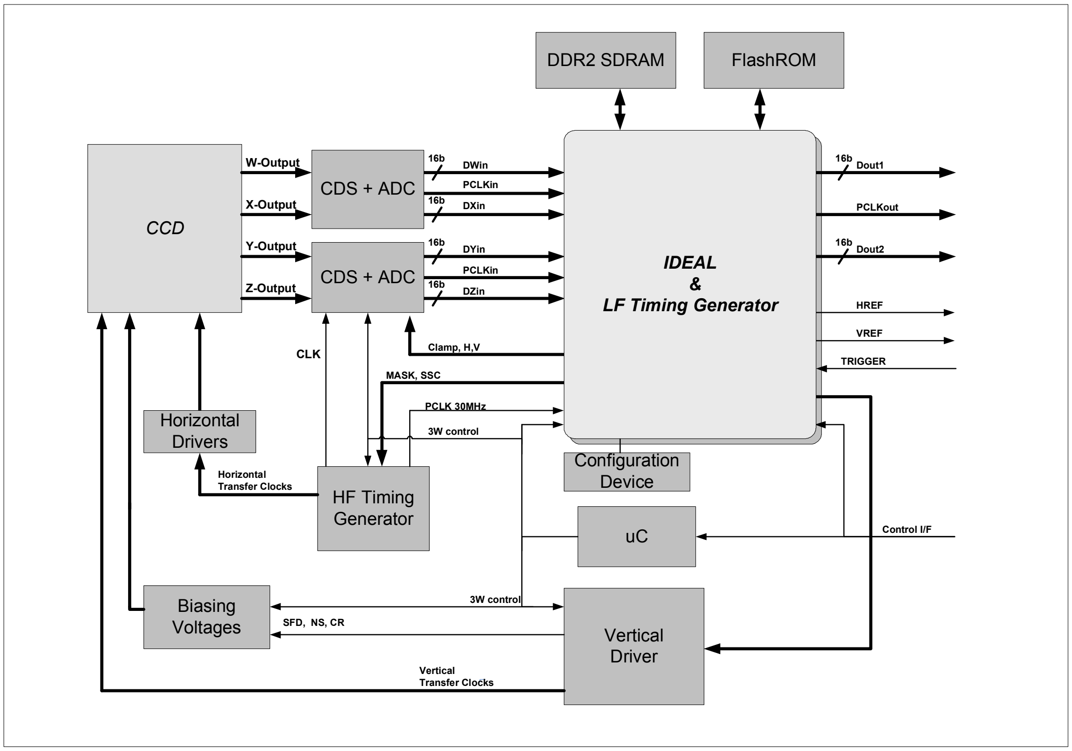


Fig. 2: Typical block diagram of a high-end professional subsystem.

The requirements for the new IDEAL concept were:

- Suitable for implementation both as ASIC and FPGA
- Capable of handling image sizes of at least 64M pixel from imagers with up to four outputs running each at maximum of 50MHz
- Sensor output tap matching to reduce output differences introduced in the analog domain (sensor amplifier output, front-end (CDS + PGA + ADC), voltage drop of the power supply)
- 16bits pre-processing compensation / correction algorithms running on a pipelined processing architecture
- Dark frame subtraction to reduce LF and HF fixed pattern noise at long and short integration times
- Blemish correction to correct single pixel defects, cluster defects and column defects
- Per-Pixel Gain compensation to reduce the Pixel Random Non-Uniformity (PRNU) and the visibility of stitch lines [10]
- Noise reduction for high ISO settings
- Measurement engine (White Balance, Histogram, Black Level)
- Output data rate up to 120MHz, 16bit parallel output
- Fast Snapshot Preview function which delivers downscaled image to camera DSP for display purposes (requires minimal additional processing on DSP)
- Flexible output interface to ensure matching with the customer’s DSP input interface requirements or data acquisition device e.g. frame grabber

2. IMPLEMENTATION

We opted for a hard-wired pipeline processing with multiple clock domains consisting of the following functional blocks (see figure 3):

- Sensor Input Interface module with pre-processing functionality
- DDR2 - SDRAM interface for temporal storage of multiple images, including short and long integration time dark frames
- FLASH ROM interface for storage of a fixed Gain / Defect table
- Digital Signal Processor module with flexible output interface
- Pulse Pattern Generator for the generation of the sensor readout pulses
- Serial or Parallel Control interface, which also allows fast (<1ms) mode switching between preview mode and still image capture mode

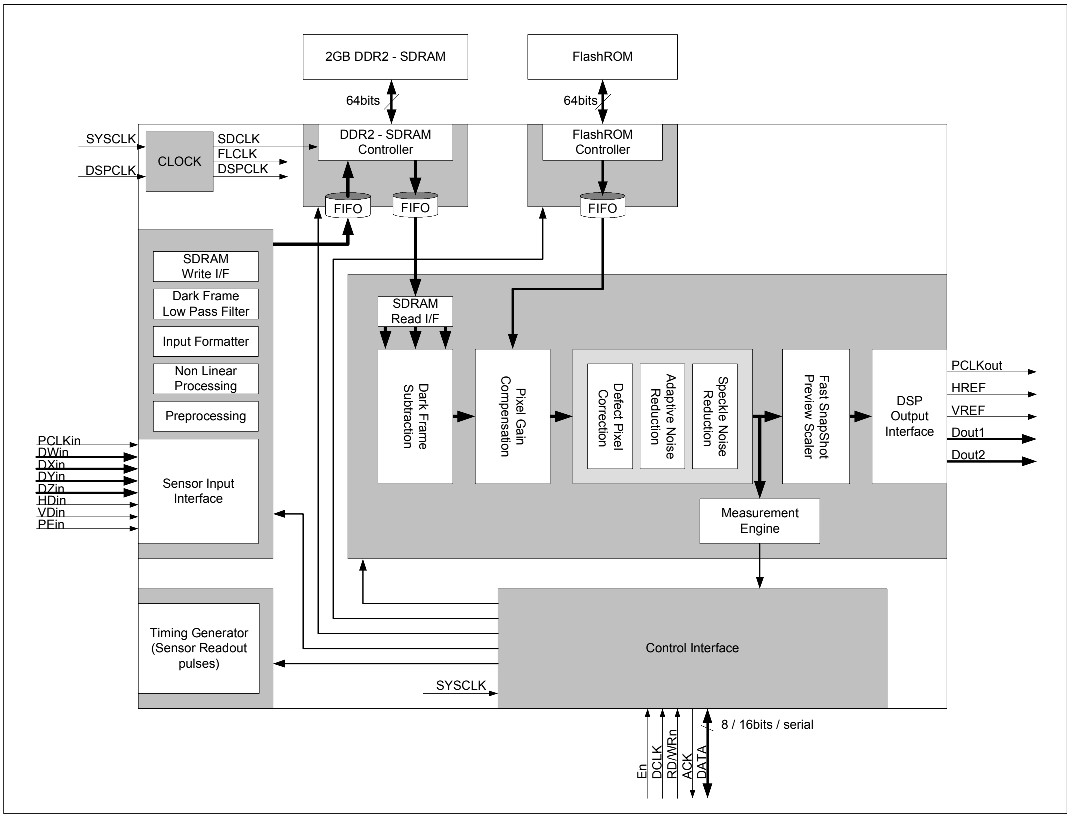


Fig. 3: Block diagram of the IDEAL pre-processing architecture for high-end professional DSC applications.

The data / processing flow is as follows: the signals from the four CCD outputs are preprocessed to compensate for offset and gain differences; a configurable LUT enables non-linearity compensations. The input formatter re-arranges the four incoming data streams (e.g. mirroring of 2nd half of the image incase of dual output or binned RG-GB data re-ordering [11]) before writing the images or various dark reference frames into the SDRAM. After reading the images, they are dark-frame corrected, PRNU compensated, and a blemish correction and noise reduction is applied. The white-balance data and histogram are computed simultaneously and can be read out via the micro-controller interface. The images are then output to the customer's DSP at speeds ranging from 25 to 120MHz depending on the DSP capabilities. The customer can then focus on applying his proprietary camera signal processing algorithms e.g. for color interpolation, contouring, gamma, JPEG compression, etc.

3. MODULE DESCRIPTION

Control Interface

The control interface module is a micro-controller interface that is especially designed for IDEAL to support the new key system requirements. The control interface supports serial / parallel communication for simple register control and high speed data transfer for larger data packets, such as a Look-Up Table data or gain/defect table to be stored in the external FlashROM that can easily go up to a file size of 128Mbytes or more.
This interface supports the following modes: Parallel 16bits, Parallel 8bits and Serial communication. Although the communication can be parallel or serial, in all cases the interface expects 16bits data for its communication, either 2xbytes for 8b Parallel, 16bits parallel or 16bits serial.

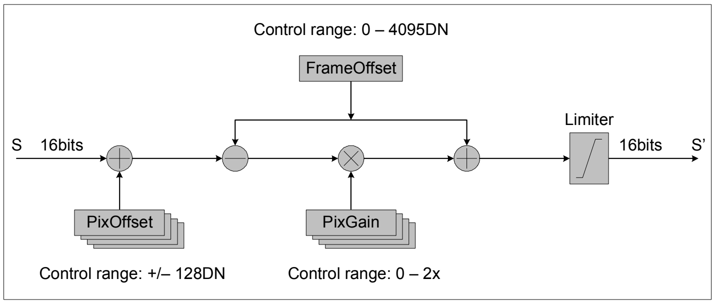
A special module is integrated in the control interface, next to the register banks, that handles the basic control register settings such as number of pixels and lines and threshold levels for the processing algorithms. This special module is responsible for the handling of the different use-cases of IDEAL in the sub-system, and therefore the camera. This function, which is called the Frame Slot Manager (FSM), interacts closely with the camera system micro-controller. The main function of the FSM is the management of the available memory slots, in other words keeping track of the memory space. The FSM can be operated in auto mode and manual mode.

Pre-processing (offset & gain and LUT)

The pre-processing module in IDEAL is the first processing step in the total processing chain that takes care of the first step for tap-matching [12]. Tap-matching is required when the image sensor is read out via multiple outputs. Offset and gain differences can cause all kinds of annoying effects, such as color differences between left and right in the image, that can easily be made visible when viewing and manipulating it in a modern photo editing program.
The total tap-matching algorithm is realized by a combination of the Pre-processing, Dark-Frame Subtraction and Pixel Gain Compensation algorithm. The pre-processing module consists of two modules, offset & gain control and programmable Look Up Table.

Offset & Gain

The offset & gain control compensates for small (linear) offset and gain differences in each front-end IC. The offset and gain of each pixel color (R, G_R, G_B or B) of each front-end data output channel can be aligned with a high accuracy. The first step in this module is to align the offset of each pixel color in each channel separately. This is done with an offset range of +/- 128DN with an accuracy of 0.25DN (based on 16-bit performance). The next step is the gain calibration. The gain calibration of each pixel color in each input channel is aligned using a multiplier with 13bit precision. This high accuracy gain calibration multiplier is used to compensate for the small gain differences for each sensor output tap. The figure and formula below show the details of this module:



The output value S’ represent the offset and gain calibrated output signal of one channel

$$S' = \begin{cases} (S + O_R - O_{FR}) \times G_R + O_{FR} & for \quad R - pixels \\ (S + O_{G_R} - O_{FR}) \times G_{G_R} + O_{FR} & for \quad G_R - pixels \\ (S + O_{G_B} - O_{FR}) \times G_{G_B} + O_{FR} & for \quad G_B - pixels \\ (S + O_B - O_{FR}) \times G_B + O_{FR} & for \quad B - pixels \end{cases} \tag{1}$$

where

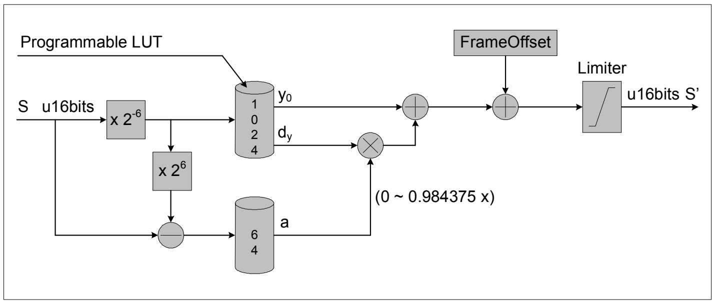
O_{FR} is Frame Offset value,
O_R, O_{GR}, O_{GB}, O_B are the offset calibration values of the Red, Green, Green and Blue pixels,
G_R, G_{GR}, G_{GB}, G_B are the gain calibration values of the Red, Green, Green and Blue pixels.

Look-Up Table

The Look-Up Table (LUT) can be used to compensate for differences in the non-linear behavior in the analog processing for each sensor output + front-end. The LUT can be initialized via the controller interface. There are four look-up tables, one for each input channel. In order to save memory, each LUT consists of two smaller LUTs: y_0LUT and d_yLUT. Each LUT can store 1024, 16bits values. A linear interpolation method is used inside the LUT module to calculate the LUT output data back to 16bits full-scale output values.

The LUT contains the following data:

- The first address of the StartLUT (indicated by y_0) contains the start value of the data input, which is smaller then 64DN. The next address of the same LUT contains the start value of the input data, which is between 64 and 128DN, etc.
- The second LUT, the DeltaLUT (indicated by d_y) contains the delta values between the data in the current and next address in the LUT.



The output value S’ represents the output signal of the LUT of one sensor output channel

$$S' = y_0(S) + \alpha(S) \times d_y(S) \tag{2}$$

where

$$y_0(S) = y_0 LUT\left(trunc\left(2^{-6} S, 0\right)\right) \tag{3}$$

$$d_y(S) = d_y LUT\left(trunc\left(2^{-6} S, 0\right)\right) \tag{4}$$

$$\alpha(S) = 2^{-6}\left(S - 2^6 \times trunc\left(2^{-6} S, 0\right)\right) \tag{5}$$

Input Formatter (Pixel Reordering)

The Input Formatter is responsible for the reordering of the incoming pixel data before it is written into the SDRAM memory. IDEAL supports several input modes such as, Normal Mode (single, dual and quad), Sub-sample (single, dual and quad) and Color Binning Mode (single, dual and quad).

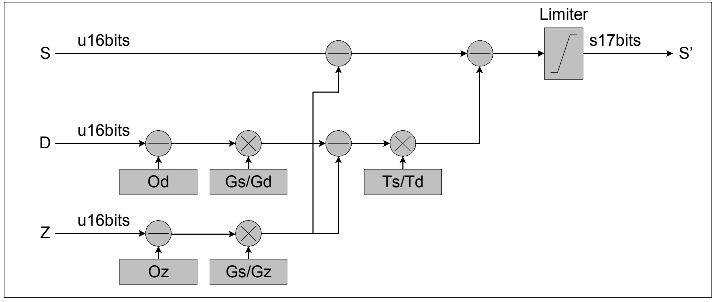
- In case of Normal Mode, dual and quad input, the second half of the image is mirrored.
- In case of Color Binning Mode, the order of the pixels is also changed to reconstruct the normal Bayer pixel order.

The Normal Mode is considered a readout mode that does not use any kind of sub-sampling or binning.

Dark Frame Subtraction

Fixed Pattern Noise (FPN) is generated by the non-uniformity of the dark current in the silicon. The generation of dark current is primarily dependant on temperature [13]. Next to that, the total amount of generated dark current and as a result the FPN is also depending on the integration time. A good method for FPN reduction is Dark Frame Subtraction. This method can be used to:

- Extend the long integration time capturing mode of the camera by reducing the FPN
- Remove run-in effects that can become visible in the picture caused by the analog design (= offset differences caused by run-in effects, start of vertical transfer)



The output value S' represents the output signal of the Dark Frame Subtraction module

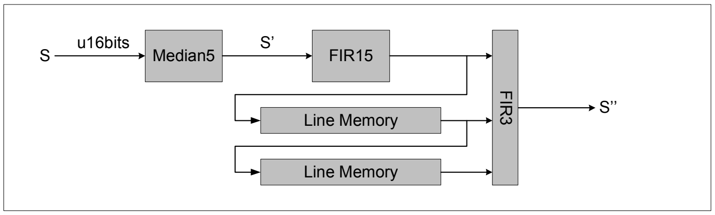
$$S' = \begin{cases} S - \dfrac{Gs}{Gz}\times(Z-Oz) - \dfrac{Ts}{Td}\times\left(\dfrac{Gs}{Gd}\left(D-Od\right) - \dfrac{Gs}{Gz}(Z-Oz)\right) & for \quad 2\,frameDFS \\ S - \dfrac{Ts\times Gs}{Td\times Gs}\times\left(D-Od\right) & for \quad 1\,frameDFS \\ S & for \quad noDFS \end{cases} \tag{6}$$

where

S is the source frame,
Z is the short integration dark frame,
D is the long integration dark frame,
Gs is the analogue front-end gain setting of the source frame,
Gz is the analogue front-end gain setting of the short integration dark frame,
Gd is the analogue front-end gain setting of the long integration dark frame,
Os is the optical black level of source frame,
Oz is the optical black level of short integration dark frame,
Od is the optical black level of long integration dark frame,
Ts is the integration time of the source frame,
Td is the integration time of the long integration frame.

Dark-Frame Low-Pass Filter

The Dark-Frame Low-Pass Filter is implemented in the Sensor Input Interface of IDEAL, which is intended to be used for capturing a zero-second dark frame. This reference frame should only contain the black level of each pixel and should be free from temporal noise and defect pixels. This temporal noise needs to be suppressed when taking a zero second dark frame for dark frame subtraction. The filter is designed in such a way that it will also eliminate large deviating pixels (defects). These pixels need to be removed from the image because these pixels will have a negative effect on the result of dark frame subtraction.



The output signal S'' represent the output signal of low-pass filtered short integration time dark frame

$$S''_{[n,m]} = \frac{\sum_{y=-1}^{1}\sum_{x=-7}^{7} S'_{[n-x,m-y]}}{n \times m} \tag{7}$$

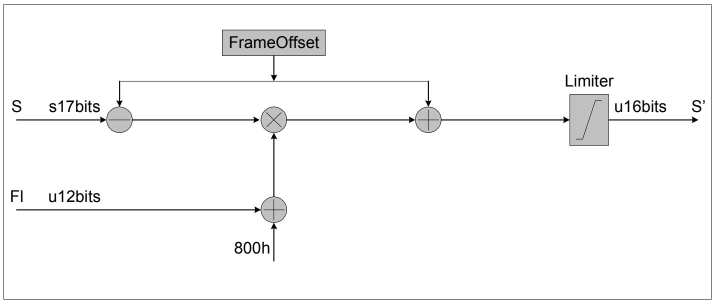
where

$$S'_{[n]} = median\left(S_{[n-2]}, S_{[n-1]}, S_{[n]}, S_{[n+1]}, S_{[n+2]}\right) \tag{8}$$

Pixel Gain Compensation

The Gain Compensation module is used for the compensation of the random non-uniformity of each pixel. The module reads data from a 12bits combined gain/defect table, which is stored in the external FlashROM. The table holds a gain compensation value for each pixel. Some special codes are reserved for the Defect Pixel Correction algorithms. The algorithm uses a 1x gain compensation value, incase the algorithm reads such a reserved defect pixel code.

$$FL = \begin{cases} 000h & PixelDefect & GainFactor = 1.0 \\ 001h & ClusterDefect & GainFactor = 1.0 \\ 002h & ColumnDefect & GainFactor = 1.0 \\ Others & NormalPixel & GainFactor = \dfrac{FL + 800h}{1000h} \end{cases} \tag{9}$$



The output signal S' represent the output of the Pixel Gain Compensation module

$$S' = \begin{cases} \dfrac{Fl + 800h}{1000h} \times (S - O_{FR}) + O_{FR} & if \quad Fl > 002h \\ S & else \end{cases} \tag{10}$$

where

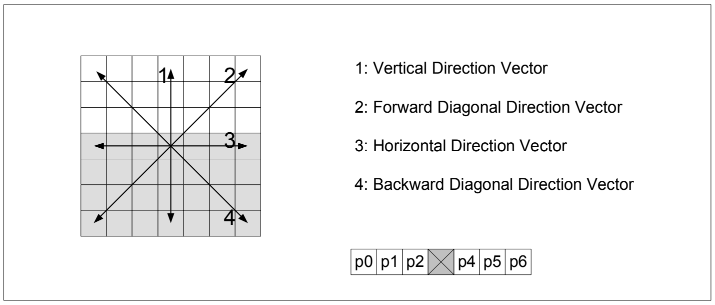
S is the sensor input data,
Fl is the 12bits gain table data,
O_{FR} is Frame Offset value.

Defect Pixel Correction

The Defect Pixel Correction module corrects all the defect pixels of the sensor that are present in the defect map. The algorithm is capable of handling single, cluster and column defects in an effective way. It uses the combined gain/defect table that is stored in the external FlashROM to determine if a certain pixel is a defect. Special codes are reserved for this purpose. The algorithm uses a 7x7 kernel with four direction vectors [14]. Each vector is analyzed on image content and the appearance of multiple defects, e.g. incase of a cluster or column defect. The best vector is then selected in order to determine whether or not a vector is capable of delivering a reliable interpolation value. For instance, it may be preferred not to correct a single defect pixel, when this single defect with a small to medium deviation is surrounded by high frequent image content. In this case, interpolation can results in a larger error.

The algorithm works in three stages:

1. Select best direction vector
2. Apply linear interpolation of the defect
3. Clip correction value


1
2
3
4
1: Vertical Direction Vector
2: Forward Diagonal Direction Vector
3: Horizontal Direction Vector
4: Backward Diagonal Direction Vector
p0
p1
p2
p4
p5
p6

Stage 1. Select best direction vector

$$BestVector = \begin{cases} Horizontal & if & Grad(hor) = Flattest \\ Vertical & elsif & Grad(vert) = Flattest \\ Forward & elsif & Grad(forward) = Flattest \\ Backward & else & \end{cases} \tag{11}$$

where

$$Flattest = \min\left(Grad\left(Horizontal\right), Grad\left(Vertical\right), Grad\left(Forward\right), Grad\left(Backward\right)\right) \tag{12}$$

$$Grad(vector) = \left|P_2 - P_4\right| \tag{13}$$

Stage 2. Apply linear interpolation of the defect

$$P_3 = \frac{P_{3a} + P_{3b}}{2} \tag{14}$$

where

$$P_{3a} = P_1 + \left(P_2 - P_0\right) \tag{15}$$

$$P_{3b} = P_5 + \left(P_4 - P_6\right) \tag{16}$$

Stage 3. Clip correction value

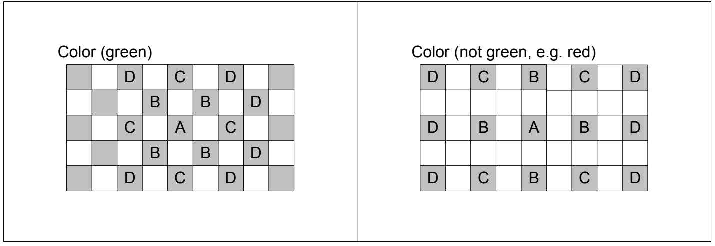
$$P_3 = \begin{cases} \min(\mathit{refpixels}) & \mathit{if} & P_3 < \min \\ P_3 & \mathit{elsif} & \min \le P_3 \le \max \\ \max(\mathit{refpixels}) & \mathit{else} & \max < P_3 \end{cases} \tag{17}$$

where

The reference pixels are the closest surrounding pixels with the same color filter.

Noise Reduction

The simplest noise reduction algorithms targeted for image applications use the assumption that the signal has a high local correlation. It uses some form of spatial averaging to minimize the noise. For some images the assumption of local correlation is more or less correct; only small part of these images fail to meet this assumption. They perform well in typical low frequency regions of the image, but they fail especially in high frequency regions [15][16]. In fact they remove the high-frequency content because of the averaging function of all the elements in the kernel. The implemented Noise Reduction module is an adaptive algorithm that can be used for the reduction of temporal noise and speckle noise. It makes a distinction between temporal noise and speckle noise. A speckle is a single deviating pixel with a large deviating value that should be handled differently.



Color (green)

▨		D		C		D		▨
	▨		B		B		D	
▨		C		A		C		▨
	▨		B		B		D	
▨		D		C		D		▨

Color (not green, e.g. red)

D		C		B		C		D
D		B		A		B		D
D		C		B		C		D

The algorithm works in two stages:

1. Determine noise type
2. Apply speckle or temporal noise reduction algorithm

Determine noise type

$$\mathit{NoiseType}(P_{ref}) = \begin{cases} \mathit{SpeckleNoise} & \mathit{if} & \mathit{Similarity}(P_{ref}) < 2 \\ \mathit{TemporalNoise} & \mathit{else} & \end{cases} \tag{18}$$

where

$$\mathit{Similarity}(P_{ref}) = \sum_{n=1}^{14} \mathit{Similar}(P_{ref}, n) \tag{19}$$

$$Similar(P_{ref}, P_n) = \begin{cases} 1 & if \quad \left|P_n - P_{ref}\right| \le Thresh(P_{ref}) \\ 0 & else \end{cases} \tag{20}$$

Speckle Noise Reduction

A 5-tap median filter is used for the reduction of speckle noise

$$Filter(P_{ref}) = median(P_A, P_{B1}, P_{B2}, P_{B3}, P_{B4}) \tag{21}$$

Temporal Noise Reduction

$$Filter(P_{ref}, P_n) = \frac{\sum_{n=1}^{15} P_n \times Similar\left(P_{ref}, P_n\right)}{\sum_{n=1}^{15} Similar\left(P_{ref}, P_n\right)} \tag{22}$$

where

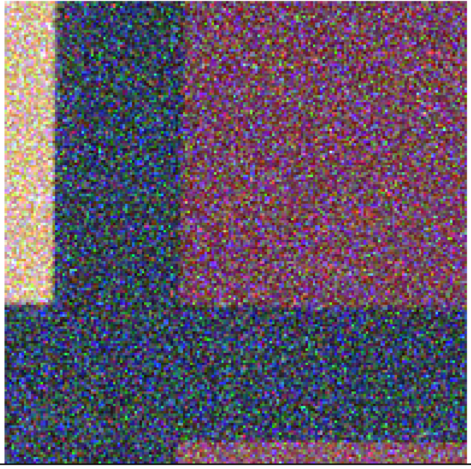
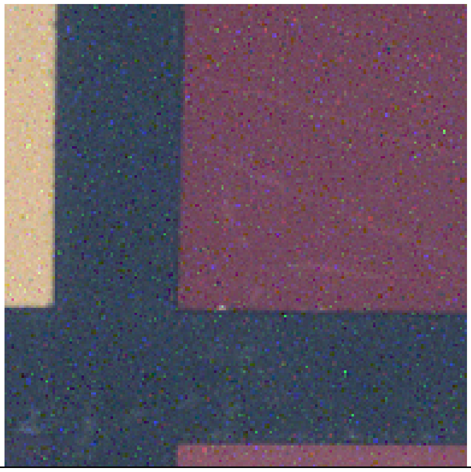
$$Similar(P_{ref}, P_n) = \begin{cases} 1 & if \quad \left|P_n - P_{ref}\right| \le Thresh(P_{ref}) \\ 0 & else \end{cases} \tag{23}$$

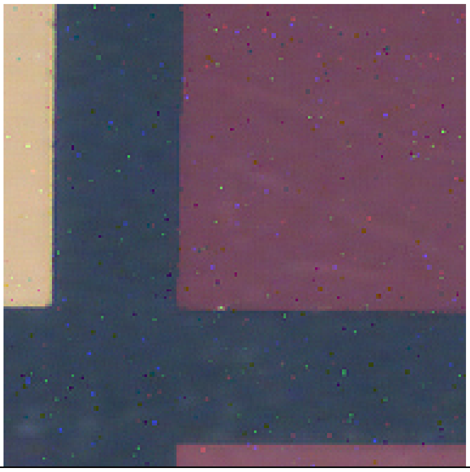
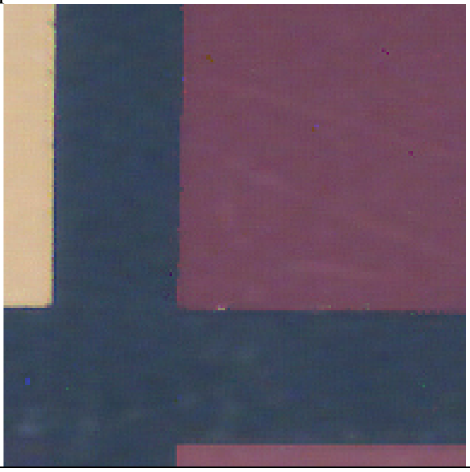
4. RESULTS

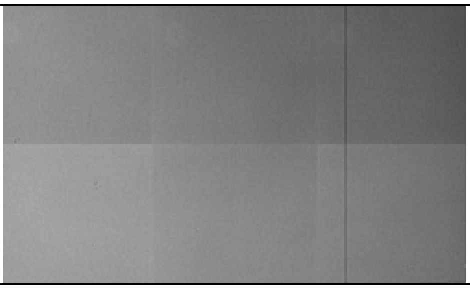
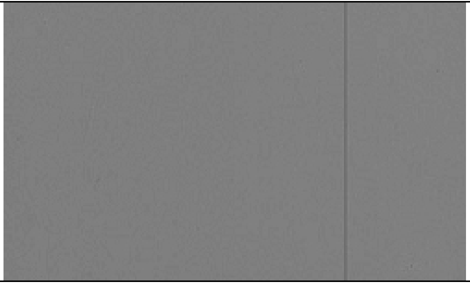
We have implemented this system using a high-end Stratix-II [17] FPGA-based prototyping platform, capable of running in quad-input mode at 4 x 50MHz, with output speeds up to 85MHz using CAMLINK base configuration. Using a 33M CCD, this allows image acquisition at 4.5 frames per second, and processing speeds of 2.5 frames per second.
The concept is suitable for migration to a low-cost FPGA family like Cyclone-III allowing cost reduction, while maintaining most of the specs.

5. RESULTS OF ALGORITHMS

The following images illustrate the results of some of the algorithms.

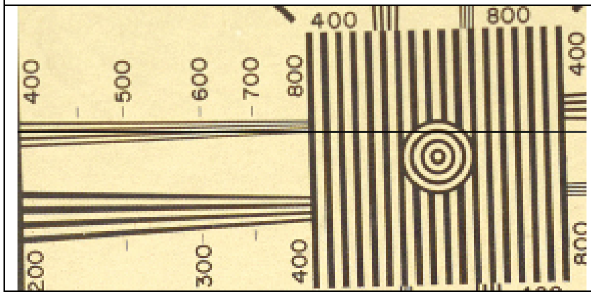
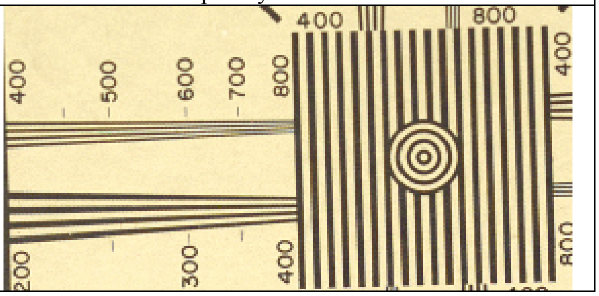
Dark Frame Subtraction off	Dark Frame Subtraction on
The picture below is a small part of a Macbeth Color Checker that was taken with a front-end gain of 2.5x and an integration time of 14seconds. 	The picture below is the result of DFS with a Dark Frame taken with a front-end gain of 8x, and an integration time of 8seconds. 

Temporal Noise Reduction on	Temporal and Speckle Noise Reduction on
The picture on the left is the result of DFS and temporal noise reduction. 	The picture on the left is the result of DFS and temporal and speckle noise reduction. 

Pixel Gain Compensation off	Pixel Gain Compensation on
The downscaled part of the picture on the left shows the stitching blocks that are introduced during manufacturing of the sensor (incl. a def. Column).	The picture on the left shows the result of the Pixel Gain Compensation algorithm with an optimal gain calibration table that is stored in the external FlashROM (Defect pixel correction in not applied).
	

An equal amount of digital gain is applied on the left and right to show:

- on the left side the visibility of the sensor stitching lines when Pixel Gain Compensation switched off
- on the right side the effect of the Pixel Gain Compensation algorithm when switched on

Defect Pixel Correction off	Defect Pixel Correction on
Defect column visible	Defect column completely concealed
	

5. SUMMARY & CONCLUSIONS

In this paper we presented an image pre-processing architecture that solves many of the challenges that the high-end professional DSC camera manufactures are faced with. The presented concept provides a solution to deal with the increasing image resolution and high acquisition speed in for instance burst mode using multiple sensor outputs. Furthermore, the presented concept provides at the same time high performance, high-speed image pre-processing algorithms in the raw data (RGB Bayer) domain in order to deliver a clean ideal sensor signal that is free from blemishes and other artifacts.

The following table summarizes the results.

	IDEAL target specification	IDEAL realized to date
Supported imager formats	FT-CCD, FF-CCD, CMOS	FT-CCD, FF-CCD: OK CMOS to be evaluated
Maximum input speed (full-frame, single shot)	4 outputs @ 50MHz pixel clock	4 outputs @ 50MHz pixel clock with DDR1: OK DDR2: implementation to be checked
Maximum input speed (frame-transfer, cont. video)	2 outputs @ 60MHz pixel clock	2 outputs @ 60MHz pixel clock with DDR1: OK DDR2: to be checked
External memory	DDR1-SDRAM: 2GB	DDR1-SDRAM: 2GB OK DDR2-SDRAM: 4GB possible

References

[1] J. Bosiers et al., “A 35-mm format 11M pixel full-frame CCD for professional digital still imaging”, IEEE T-ED, Vol. 50, pp. 254 - 265, January 2003

[2] J. Bosiers, B. Dillen, C. Draijer, I. Peters, “ Imagers for professional digital photography”, Proceedings ICIS 2006, Rochester, NY

[3] http://www.dalsa.com/sensors/products/productdetails.asp?productID=FTF5066c

[4] E.J. Meisenzahl et al., “31Mp and 39Mp full-frame CCD image sensors with improved charge capacity and angle response”, Proc. SPIE vol. 6069, January 2006

[5] http://www.phaseone.com/Content/p1digitalbacks/P65plus/Introduction.aspx

[6] http://www.leaf-photography.com/ShowProductDetails/MenuID/964/ParentMenuID/314

[7] http://www.hasselblad.com/

[8] A.J.P. Theuwissen, “Image processing chain in Digital Still Cameras”, VLSI Circuits, 2004. Digest of Technical Papers. 2004, 17-19 June 2004 Page(s):2 – 5

[9] http://www.mamiya.com/

[10] G. Kreider et al., “An mK × nK bouwblok CCD image sensor family—Part I: Design”, IEEE Trans. Electron Devices, Vol. 49, pp. 361 - 369, March 2002

[11] C. Draijer, F. Polderdijk, A. van der Heide, B. Dillen, W. Klaassens, J. Bosiers, “A 28 Mega Pixel Large Area Full Frame CCD with 2x2 On-Chip RGB Charge-Binning for Professional Digital Still Imaging”, IEDM2005

[12] M.J. Kiik, ; C.J. Flood, G.P. Weale, G.P.; S. Gareth Ingram, “Frame transfer area array sensor with vertical antiblooming and novel readout for enhanced performance”, Electron Devices Meeting, 1997. Technical Digest., International 7-10 Dec. 1997 Page(s):189 – 192, 650309

[13] B. Botte, “Digital automatic pixel correction in new generation CCD broadcast cameras”, Broadcasting Convention, 1992. IBC., 3-7 Jul 1992 Page(s):474 - 478

[14] A. Tanbakuchi, A. Sijde, B. Dillen, A. Theuwissen, W. Haan, “Adaptive pixel defect correction” Proc. SPIE Int. Soc. Opt. Eng., Vol. 5017, 2003

[15] A. Bosco; M. Mancuso, “Adaptive filtering for image denoising” Consumer Electronics, 2001. ICCE International Conference, 19-21 June 2001 Pages 208 – 209, 935277

[16] T. Rabie, “Adaptive hybrid mean and median filtering of high-ISO long-exposure sensor noise for digital photography” Journal Electron. Imaging, Volume 13, 264 (2004)

[17] http://www.altera.com/products/devices/stratix-fpgas/stratix-ii/stratix-ii/st2-index.jsp