59th ILMENAU SCIENTIFIC COLLOQUIUM
Technische Universität Ilmenau, 11 – 15 September 2017
URN: urn:nbn:de:gbv:ilm1-2017iwk-042:3

SMART PARALLEL SPECTRAL IMAGER BASED ON HETEROGENEOUS FPGA SYSTEM ON CHIP

Mathias Schellhorn, Richard Fütterer, Maik Rosenberger, Gunther Notni

Technische Universität Ilmenau, Faculty of Mechanical Engineering, Department of Quality Assurance and Industrial Image Processing

ABSTRACT

In the last years, industrial image processing has been shifting to areas and tasks that are increasing in complexity. This results in new challenges in order to contrast features to be detected or evaluated. Systems for the acquisition and interpretation of multispectral images are thus becoming more and more interesting. A major issue is, depending on the sensor principle, the time to acquire this spectral data. FPGA (Field Programmable Gate Array) and in particular heterogeneous FPGA SoC (System-on-Chip) offer the possibility to accelerate these acquisition methods decisively. In addition to the image acquisition, it is also possible to calculate decisive preprocessing steps in the hardware. A frequently used algorithm for analyzing but also compressing hyperspectral data is the PCA (principal component analysis). This paper presents a research setup that combines a heterogeneous FPGA SoC with a 12-channel filter wheel camera. With the help of the device a parallel working PCA is to be integrated, which works distributed in hardware and software. The paper presents the concept for this implementation and the current state of development in the project. In addition, restrictions on the development of algorithms with hardware systems and the current distribution in hardware and software are discussed.

Index Terms - heterogeneous FPGA SoC, embedded processing, hyperspectral image processing, PCA

1. INTRODUCTION

The industrial image processing technology has continuously developed further over the past decades. From black-and-white cameras that use contrast differences to detect edges or defects, the development has gone to ever-more accurate working colour cameras, which enable the evaluation of colour deviations. Colour cameras mimic the vision of the human eye and capture therefor three spectral regions: red, green and blue (RGB) with the greatest sensitivity in the green spectral range. Conventional imaging was thus adapted to the visual perception of humans. Moreover, additional information were accessible that improves the reliability of algorithms in decision making.
However, the requirements in recent years have changed fundamentally. Various technology fields require next-generation imaging systems that allow the analysis of data sets with a variety of colour or spectral channels. These systems are used, for example, for non-destructive testing procedures, separation of valuable substances in recycling and new imaging technologies in the pharmaceutical, healthcare and nutritional industries.
By expanding the spectral range beyond the visible spectrum to near infrared and ultraviolet, the information content is significantly increased. In addition, the subdivision of the visible spectrum into several narrowly limited channels provides additional data points. Imaging systems were thus been further developed from single-channel (greyscale), over three-channel (RGB) to multi-channel systems. Depending on the number of spectral channels, these systems

are referred to as multispectral (typically 3-10 channels) or hyperspectral (> 10 channels) systems. Since the results discussed in this publication can be used for any number of spectral channels, only the term hyperspectral is used below.
Hyperspectral data analysis serves as an important tool especially in the evaluation of reflection phenomena and thus enables the measurement of spectral signatures or chemical composition of objects within the field of view of the sensor. Hyperspectral images therefore merge spectral with spatial information. The acquired data is very complex and often beyond human understanding. Nonetheless, the data can be interpreted more reliably and more widely than regular colour image data. Methods of machine recognition that interpret the reflexive behaviour of objects and even edge detection algorithms can benefit from the analysis of these data.
Hyperspectral images are organized as three-dimensional data as so-called hyperspectral data cubes (Fig. 1). The first two dimensions of the cube are spatial (*x*,*y* axes) while the third dimension (*z* axis) is the wavelength. Each *x*-*y*-layer corresponds to the view of the scene at a specific wavelength. While each pixel describes a spectral signature within the spatially resolved image.

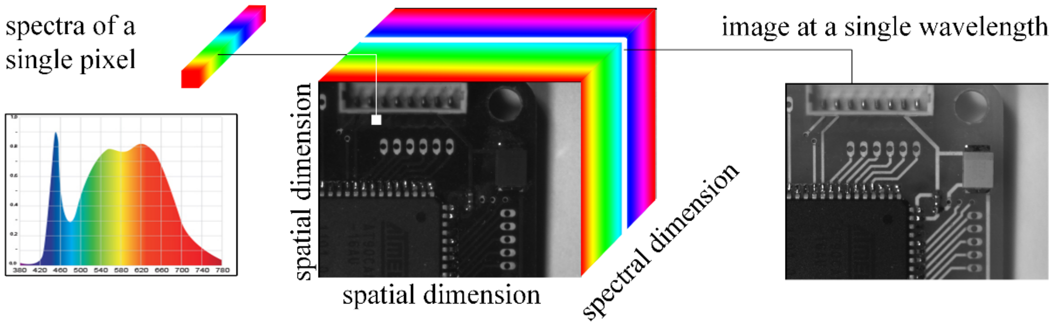


Fig. 1: Interpretation of data within the hyperspectral data cube

Sophisticated methods of image analysis, processing and visualization are needed, in order to process the high-dimensional data sets with its enormous information density. Due to the narrow-band channels and the high spectral sampling rate, hyperspectral image data is highly correlated. This results in some degree of redundancies in the data. The determination and reduction of the inherent dimensionality of the data is in many cases a necessary requirement for a successful analysis of the data.
In the case of a manageable number of features, the identification of non-meaningful or redundant features can already take place through intuition or knowledge gained through advanced data analysis and visualization. Often, however, analytical methods of dimensional reduction such as Principal Component Analysis (PCA) are used. With the PCA, a smaller number of new latent features (components) can be derived by transforming the features. These components have only a slightly lower significance than the original feature set. A reduction to two or three components allows a visualization of the objects in 2D or 3D feature plots and enables the visualization of contexts. This often allows existing image processing algorithms to be applied to spectral data without additional adjustments.

2. STATE OF THE ART HYPERSPECTRAL IMAGING TECHNIQUES

An ideal hyperspectral imager allows simultaneous acquisition of the entire hyperspectral data cube. When considering colour cameras as a special case of spectral imaging (3 spectral channels), 3-CCD colour cameras are an ideal spectral camera as they capture all three spectral channels with full lateral resolution of the sensors simultaneously. Theoretically, it is possible to expand this setup for many channels. However, a beam splitter required for this purpose cannot be realized at the present state of the art. Since available photoelectric sensors only have grids with a maximum of two spatial coordinates, current hyperspectral imager scan or interpolate at least one dimension of the spectral tube. With respect to the hyperspectral tube, there are four general approaches to the image acquisition of spectral image data: point scanning, line scanning, area scanning and single shot.

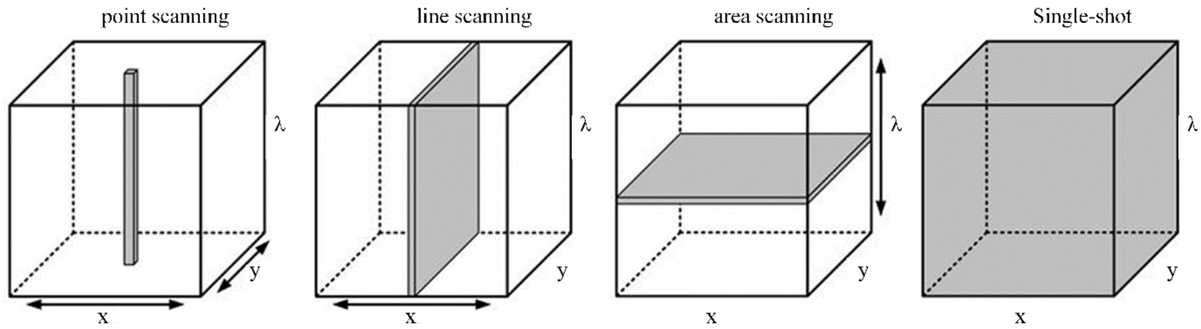


Fig. 2: General approaches for acquisition of hyperspectral data

As the name implies, point scanning systems use a point sensor to capture the spectrum of a single pixel. For example, a spectrometer can be used for this purpose. With this sensor, the entire measurement scene must be scanned. The method is thus very time-consuming, but provides, depending on the sensor, a very fine spectrum of each individual pixel. This imaging systems are also known as Whiskbroom imager.

A further development are line scanning systems, which use a line of spectral sensors. Thus the scene has only to be scanned in a spatial direction. These systems are also referred to as push-broom scanners. This configuration makes it possible to use simpler sensors for different wavelength ranges, which are arranged in lines next to another. The method is thus faster, cheaper but mostly with a lesser spectral resolution.

Area scanning systems take up the individual layers of the spectral cube one after the other. For this, mostly broadband array sensors are used, while the spectral information is separated through filters. Known designs are filter wheel cameras, in which the filters are mechanically changed, or cameras with tunable filters. These methods have a particularly advantageous relation between spatial resolution and acquisition time.

Snapshot or single-shot cameras take the complete spectral data cube with a single acquisition. Since ideal spectral cameras do not (yet) exist, matrix sensors are used in which filters were applied to the individual pixels. The individual spectral pixels usually form a square voxel, which contains a certain number of spectral channels. Similar to RGB cameras with colour filter array, the spectral information between the individual pixels of a filter type must be interpolated. These sensors work very fast, but they require expensive manufacturing processes and can not be customized afterwards.

3. SYSTEM CONCEPT FOR A SMART PARALLEL SPECTRAL IMAGER

For the research in the Department of Quality Assurance and Industrial Image Processing at the Faculty of Mechanical Engineering of the Ilmenau University of Technology, a multispectral filter wheel camera [1],[2],[3] was developed. Within the scope of the project “QualiMess next generation” (03IPT709X), the existing research setup will be revised and enhanced with a heterogeneous FPGA SoC module. The aim is to minimize the latency between image acquisition and spectral image output. Additional hardware resources will be used to implement a pre-processing of the hyperspectral data.

3.1 Research setup of a spectral imager based on filter wheel principle

Fig. 3 shows the actual hardware platform. The filter wheel of the camera system contains 12 metal interference bandpass filters in steps of 50 nm from 400 to 950 nm with a 50 nm bandwidth. The filters are exchangeable, which means that other spectral data compositions are also possible due to the use of application-specific filters.

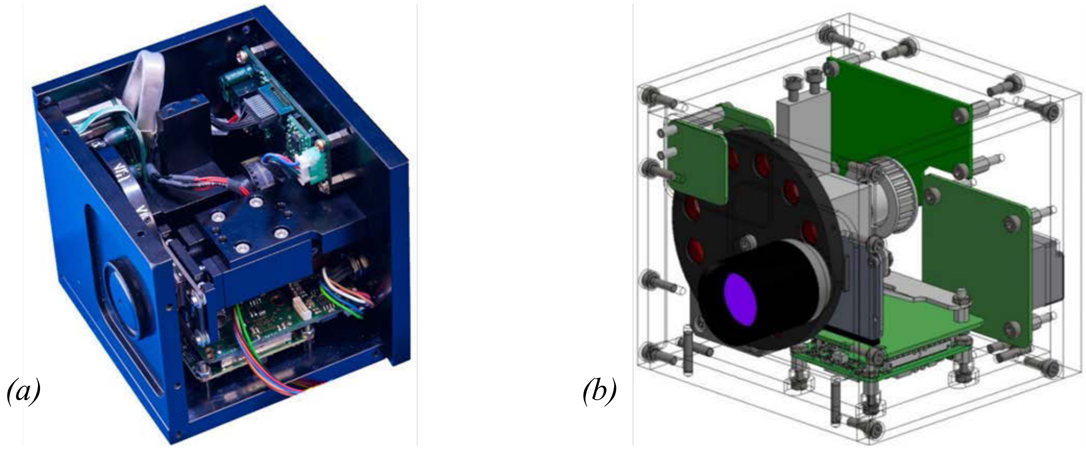


Fig. 3: Spectral imager in current hardware configuration (a) and cutaway CAD view (b)

As shown in Fig. 4, a FPGA forms the central control unit of the hardware platform. A Xilinx Spartan®-6 captures the data from the sensor and transmits the images via a standard GigE Vision (Gigabit Ethernet for machine vision) interface. The configuration of the image sensor (CMOS sensor EV76C660 e2v) is done via an implemented I^2C (inter-integrated circuit) interface and can be accessed by the user via the GigE Vision interface. The filter wheel position can also be controlled via this interface. For this purpose, specific nodes were created in the generic API (application programming interface) GeniCam. FPGA internal control tasks are managed by an implemented MircoBlaze™ softcore processor.

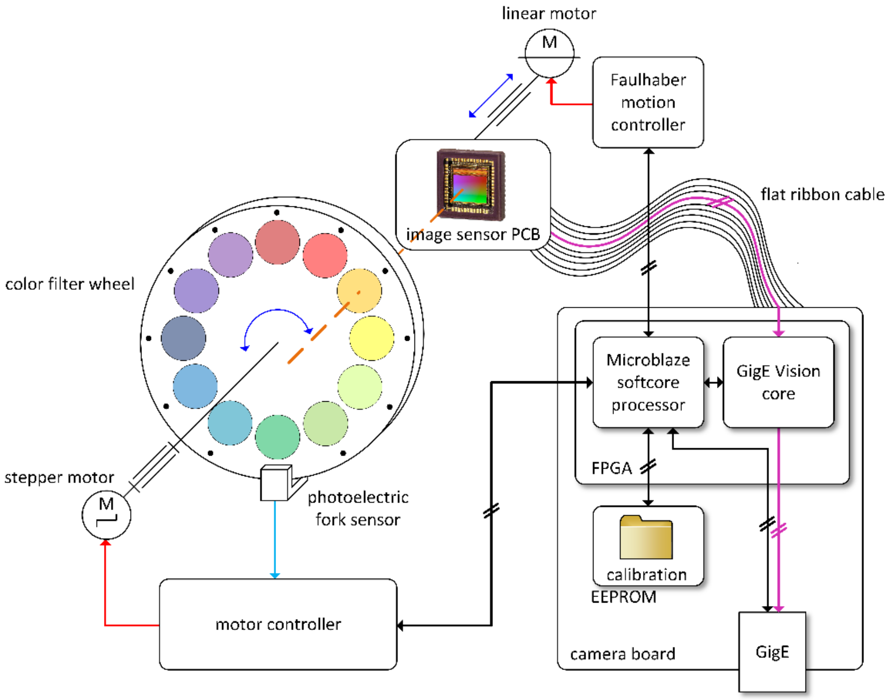


Fig. 4: Block diagram of the existing setup [3]

For the control of the mechanical components, two additional microprocessors were integrated into the imager. The first microcontroller takes control of the rotation of the filter wheel. In addition, the current filter position is detected via a photoelectric fork sensor and a trigger signal is generated for synchronous image acquisition. The second microcontroller controls the translational movement of the sensor PCB (printed circuit board) to regulate the distance between the image sensor and the filter. This is necessary to compensate the chromatic aberration, which is caused by the working principle [4], [5], [6]. Corresponding position data are determined by calibration and are stored on an EEPROM (electrically erasable programmable read-only memory) integrated in the system. This data can also be accessed and updated via the GigE Vision interface.

3.2 System concept for a revised smart spectral imager using the Xilinx Zynq® All Programmable SoC

The individually distributed microprocessors require, in addition to their own hardware, corresponding firmware as well as communication interfaces to the FPGA. During communication, latencies occur which influence the acquisition time and, in the worst case, lead to desynchronization. Therefore, in the revised version of the system, these functionalities are to be integrated into the central FPGA. This could be done by using additional softcore processors in the programmable logic. A better approach, however, is the use of SoC devices, which provide a hardcoded processor system (PS) in addition to the programmable logic (PL). The advantage over distributed systems lies in the tight connection between PS and PL, whereby the FPGA resources can be fully utilized for parallel processing. This is not only cheaper but also saves energy and space on the PCB.

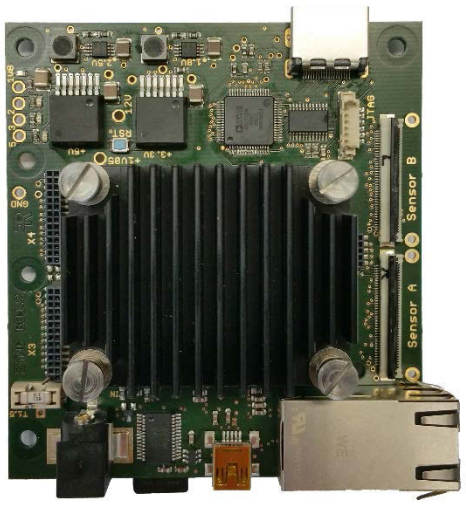
With the Zynq® All Programmable SoC, Xilinx has developed a solution for this purpose. The device combines an ARM Dual-Core Cortex A9 Core, programmable logic and essential peripheral functions on a single chip. It is supported by a wide range of IP modules and a tool infrastructure, which also allows very demanding tasks to be realized. Versatile prefabricated IP (intellectual property) modules and a tool infrastructure, which also enable demanding tasks to be implemented, support the development process.

The use of a SoC provides numerous advantages for the further development of the spectral imager. By combining the control and processing blocks, more complex functional sequences are possible. The ARM processors, clocked up to 800 MHz, provide higher data processing speed than the 200 MHz clocked MicroBlaze™ processors, enabling more efficient processing in software. The Zynq platform has physical I/O elements such as AXI bus (Advanced eXtensible Interface) systems, USB 2.0 (universal serial bus) host and Gigabit Ethernet, which allow the processor to be used without the use of PL resources. In the PL itself, 28 k to 350 k logic cells as well as dedicated DSP (digital signal processing) slices with hardware multiplier and adders are available.

The additionally available logic cells are used to implement preprocessing steps of the spectral image processing. Thus, on the one hand, the acquisition of the spectral data cube is to be accelerated and, in addition, subsequent analysis steps are to be integrated. As a sample application, a PCA will be implemented in the framework.

As hardware platform, a carrier board created in the research project “QualiMess next generation” is used. It connects an TE0720 FPGA micro module from Trenz Electronic with corresponding peripherals, which were selected with regard to the use in image processing solutions. The technical specifications are listed in Table 1.

Table 1: Hardware platform and technical specifications

<table>
<tr><td rowspan="5">
</td><td>FPGA:</td><td>Zynq-7000 All Programmable SoC (AP SoC) with ARM dual-core Cortex-A9 MPCore (XA7Z020-1CLG484Q)</td></tr>
<tr><td>Clock speed (ARM):</td><td>up to 666 MHz</td></tr>
<tr><td>Memory:</td><td>32-Bit-wide 1 GB DDR3 SDRAM
2K serial EEPROM</td></tr>
<tr><td>Programming Interface:</td><td>Micro-SD Card Reader, integrated USB-JTAG</td></tr>
<tr><td>Interfaces:</td><td>HDMI, RS-232, SPI, I²C, PL connected Gigabit Ethernet, 2 FFC for image sensor connectivity, 2 additional connectors with PL connection</td></tr>
</table>

For the hardware, a corresponding image processing framework has already been developed within “QualiMess next generation”, which will be used in the revision. It is based on a memory-decoupled processing architecture and allows flexible image acquisition including the processing of reference images from the SD card. Image data can be buffered directly on the connected SDRAM (synchronous dynamic random-access memory) during processing. This also forms the basis for the processing of the individual spectral channels, which must be buffered during the sequential image acquisition.

4. IMPLEMENTATION OF THE PRINCIPAL COMPONENT ANALYSIS

The biggest challenge that occurs when analyzing multidimensional data sets is that they cannot be represented two-dimensionally. Also multidimensional representations do not always reflect the essential information of the data. Even the representation of hyperspectral data sets as a cube serves the purpose of data organization rather than an adequate representation. The more variables (dimensions) a data record has, the more complicated and more obscure the situation becomes. This means that existing patterns cannot be recognized at first sight.
The central idea of the principal component analysis (PCA) is to project the data to a lower-dimensional (in the best case, two-dimensional) plane so the hidden patterns become visible. The visible structure of the projected data depends on the direction of this projection. Assuming that the information can only be obtained from the data when the variance along an axis is a maximum, the direction of this axis must be determined as the new basis of the projection. In addition, the new axes of the projection should be orthogonal to each other.
For this purpose, an orthogonal linear transformation of the original variables into a new set of uncorrelated variables, the principal components, is performed. The correlation of multidimensional data is minimized by the transfer into a vector space with a new basis. This allows the identification of patterns within the data set. The principle components thus form an important starting point for further analysis algorithms. In addition, the PCA is suitable for general data reduction. A certain loss of information is unavoidable, but this should only include noise components.

4.1 Theory of principal component analysis in image processing

Since the mathematical principles of the PCA are well documented [7],[8], this is not explicitly addressed here. Instead, this paper presents only the application on hyperspectral image data.

The initial data for a PCA is corellated multidimensional data. This prerequisite is given for spectral images, in particular for hyperspectral images, since the individual spectral bands are very narrow and close to each other.
In order to apply operations of the matrix calculation, required in PCA to images, an interpretation of the image data is useful. For this, the spatial dimensions are neglected and the image becomes as a vector of the sequential grey values. An image of the dimension of *i* columns and *j* rows therefore yields a vector with *i·j* elements. The first *i* elements of the vector thus correspond to the first line of the image, and so on. This could also be considered a one-dimensional image. This also corresponds to the output of digital image sensors, which typically output the individual pixels one after the other. This is also referred to as the image data stream.

$$\vec{x} = (x_1, x_2\ , x_3, \dots x_{ij})$$

Extending into the hyperspectral data space with *n* spectral channels, this results in an *ij* by *n* matrix. In other words *ij* (for each pixel) *n*-dimensional spectral vectors.

$$X(\lambda) = \begin{pmatrix} \vec{x}_{\lambda_1} \\ \vdots \\ \vec{x}_{\lambda_n} \end{pmatrix} = \begin{pmatrix} x_1(\lambda_1) & \cdots & x_{ij}(\lambda_1) \\ \vdots & \ddots & \vdots \\ x_1(\lambda_n) & \cdots & x_{ij}(\lambda_n) \end{pmatrix} = \begin{pmatrix} \vec{\lambda}_{x_1} & \cdots & \vec{\lambda}_{x_{ij}} \end{pmatrix}$$

Next, the covariance matrix is calculated. For this, the covariances of the individual dimensions must be calculated. For the hyperspectral data, the covariance of each image, at a particular wavelength, must be calculated respectively to each image of the other wavelengths. The image vector is therefore re-centered so that the mean value is zero. This is achieved by subtracting the mean value μ of the image from each pixel. This adjustment is made since PCA deals with the variance among the original variables, the means are irrelevant.

$$cov(\vec{x}_{\lambda_a}, \vec{x}_{\lambda_b}) = \frac{\sum_{k=1}^{i\cdot j}(x_k(\lambda_a) - \vec{x}(\lambda_a))(x_k(\lambda_b) - \mu(\lambda_b))}{(i\cdot j - 1)} \qquad with\ \ a, b = 1 \dots n$$

An n-dimensional data set thus results in a $n\ x\ n$ covariance matrix. On the main diagonal of the matrix are the variances of the individual wavelengths. Since the covariances $cov(\vec{x}_{\lambda_a}, \vec{x}_{\lambda_b})$ and $cov(\vec{x}_{\lambda_b}, \vec{x}_{\lambda_a})$ are identical, the matrix is mirrored to the main diagonal.

$$C = \begin{pmatrix} cov(\vec{x}_{\lambda_1}, \vec{x}_{\lambda_1}) & \cdots & cov(\vec{x}_{\lambda_1}, \vec{x}_{\lambda_n}) \\ \vdots & \ddots & \vdots \\ cov(\vec{x}_{\lambda_n}, \vec{x}_{\lambda}\ ) & \cdots & cov(\vec{x}_{\lambda_n}, \vec{x}_{\lambda_n}) \end{pmatrix}$$

The Eigenvalues and Eigenvectors of the covariance matrix are calculated according to the calculation process of the PCA. The Eigenvalues and their associated Eigenvectors are sorted according to their value in descending order. A feature vector can now be formed from the Eigenvectors. For a data reduction, any number of the most significant Eigenvectors can be selected.

$$F = (\overrightarrow{eig}_1\ \overrightarrow{eig}_1 .. \overrightarrow{eig}_n)$$

The last step is the calculation of the transformed data in which the mean-corrected original data set is multiplied by the transposed feature vector. If the feature vector is reduced, the dimension is reduced to the selected number of Eigenvectors. This can also be used for data compression since a back-transformation with the feature vector returns the original data. However, this results in data loss as a function of the image content and the compression factor.

4.2 Algorithm development and reference in MATLAB

In order to prepare the algorithm for the heterogeneous FPGA SoC platform, a general implementation was evaluated in MATLAB. In this context, the realization with synthesizable operations as well as the general feasibility with regard to scaling and parallelization were examined. In addition to the fast implementation of the algorithm, the MATLAB environment provides a way to generate reference data for final evaluations. In the algorithm, the simplest possible calculation operations should be used to enable subsequent optimization of the HW-SW partitioning.

A special challenge was the solution for the Eigenproblem, since it was necessary to dispense with the inclusion of arithmetic libraries. In principle, a linear system of equations has to be solved, which has just as many equations and unknowns as the dimension of the spectral data. This becomes more difficult with increasing number of dimensions or individually considered wavelengths. Therefore, the implementation of an approximation method was attempted.

A common method used in such cases is the QR algorithm. It is based on the QR decomposition a method from the fields of linear algebra and numeric, which describes the decomposition of a matrix A into the product of two matrices:

$$A = Q \cdot R$$

Q describes an orthogonal ($Q^T Q = I$) and unitary matrix ($Q * Q = I$) and R an upper triangular matrix. In the QR algorithm, a recursive matrix sequence is formed starting from A:

$$A_1 = A, A_1 = Q_1^{-1} A_1 Q_1, \dots, A_{k+1} = Q_k^{-1} A_k Q_k$$

Since all transformations in the recursion are similarity transformations, all matrices of the matrix sequence have the same Eigenvalues. Under the condition that A is a quadratic and symmetric matrix and no Eigenvalues λ, μ exist with $\lambda = -\mu$, the sequence of the matrices A_k constructed by QR-algorithm converges to a diagonal matrix containing the Eigenvalues of A. The columns of the matrix $P = Q_1 \cdot Q_2 \cdot \dots \cdot Q_k$ converge against the corresponding Eigenvectors. The QR-algorithm converges considerably faster for tridiagonal matrices (symmetric Hessenberg matrices), which can be reached in particular by a householder transformation.

The mathematical basics of these methods are well documented in [8],[9] and are therefore not further elaborated at this point. The presented procedures were used for the implementation of the PCA in MATLAB. A validation of the data on real spectral data sets showed very good results. Thus, the values of the diagonal of A_k converge to the Eigenvalues after a few iteration steps. For this validation, data sets such as AVIRIS (Airborne Visible / Infrared Imaging Spectrometer) and HYDICE (hyperspectral digital imagery collection experiment) with up to 227 channels were used in addition to the data records of the filter wheel camera. This shows that the algorithm is scalable in terms of the number of spectral channels. Fig. 5 shows the results of the calculation on the spectral image data set of the filter wheel camera. The dimensions have been reduced to the three most relevant channels in order to generate a false colour image suitable for display. Metallic surfaces, traces under solder resist and plastic surfaces are reliably separated.

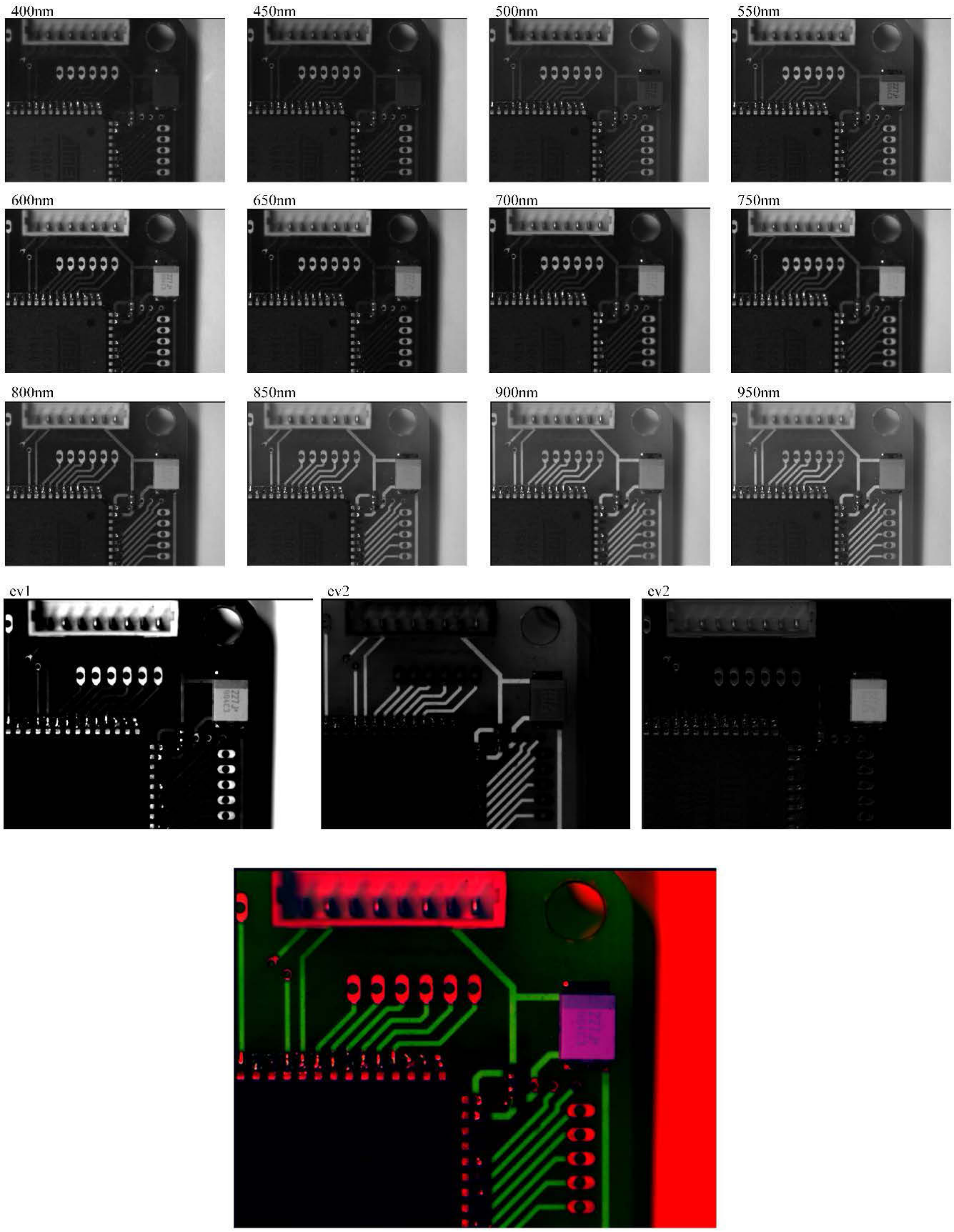


Fig. 5: Evaluation spectral data set with 12 channels (400 - 950 nm), transformed and compressed data with 3 channels (ev1 ev3) and merged false colour image.

4.3 Functional concept of the hardware implementation

When developing with FPGA devices, the developer faces the following challenges:

- No native floating-point operations
- Divisions in hardware consume many resources and are time-intensive
- Internal memory is expensive and limited
- External memory access is often the bottleneck
- Optimization strategies are time-consuming

However, there are advantages, which are not to be underestimated. Thus, the structure of FPGA fabric allows real parallel processing and pipelining. Current devices provide internal powerful DSP resources in hardware. In SoC, there is a close link between PL and the PS as well as broadband connections to fast external memory.
With regard to these limitations and abilities, the developed algorithm was partitioned. For the implementation in the PL, the arithmetic operations requiring direct access to image data have been prioritized. This includes the mean value determination, the calculation of the covariances as well as the final calculation of the transformed data. The calculation of the Eigenvalues and Eigenvectors together with the determination of the feature vector was determined for an implementation in the PS. Parallel processing is to be used in particular with regard to simultaneous image acquisition and evaluation.
In the filter wheel camera described here, the images of the individual filters are taken one after the other. During the acquisition of an image, the grey values can be added consecutively to calculate the mean value of the image at the end. The division of the sum by the total number of pixels can take place in the PS, which spares limited hardware calculators. For further calculations, especially for the calculation of the covariances, the individual images have to be stored temporarily. This is done in the external RAM, which is accessed with a video-optimized DMA (direct memory access) core.
By completing the calculation of the mean value for the second filtered image, the calculation of the covariances can start. With each filter stage, the number of covariances to be calculated in parallel increases. After the second image two covariances, after the third three more and so on. This is one of the biggest problems during the calculation, because as the number of filters increases, more images have to be calculated in parallel and read from the external memory. RAM accesses must be performed with scatter-gather methods. In addition, data must be buffered in the block RAM. With a processing clock that is much larger than the incoming pixel clock as well as parallel processing stages, this problem can be solved within certain limits. Nonetheless, memory access in particular limits the maximum number of images to be processed in parallel.
The iterative solution of the Eigenproblem can be calculated in software. With the feature vector, the transformed data can be calculated within the PL. For this, similar calculation and buffering structures must be used as in the covariance calculation. Internally, however, the calculations can continue until the end of the recording of the second filter image. An exception is the case when, in addition to the transformed images, the RAW data must also be transmitted

5. CONCLUSIONS AND CURRENT STATE OF WORK

In this paper, the current concepts for a smart parallel spectral imager were presented. The revised hardware as well as the approaches for the integration of preprocessing algorithms were addressed. In particular, the preparation of the PCA algorithm for implementation in a hardware-software-co-design was discussed. The results of these preparatory works have just been part of the current work within the research project “QualiMess next generation”.
Currently the hardware implementation is in development. For this purpose, individual portionable elements of the algorithm are programmed as independent IP cores. The IP is mainly implemented in VHDL (very high-speed integrated circuit Hardware description language) but also in HLS (high-level synthesis). For this, development tools from Xilinx are used. Therefore, the implementation will be tailored to Xilinx SoC and not freely portable. The IPs will flow into the framework of the filter wheel camera, where they are finally evaluated.

6. ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support of the research project *"QualiMess next generation"* (03IPT709X) within the framework InnoProfile Transfer funded by the German Ministry of Education and Research (BMBF).

REFERENCES

[1] Preißler, M.; Rosenberger, M.; Correns, M.; Schellhorn, M.; Linß, G.: "Investigation on a modular high speed multispectral camera" In: Proceedings of the 20th IMEKO TC2 Symposium on Photonics in Measurement, proceedings, pp. 59-62 Aachen (Germany): Shaker, 2011.
[2] Preißler, M.; Rosenberger, M.; Linß, G.: "Smart multispectral imager" In: IEEE Photonics Conference (IPC), 2012, proceedings, pp. 591-592, Piscataway (USA): IEEE, 2012.
[3] Rosenberger, M.; Celestre, R.: "Smart multispectral imager for industrial applications" In: IST: 2016 IEEE International Conference on Imaging Systems and Techniques , proceedings, pp, 7-12, Chania (Greece): IEEE, 2016.
[4] Rosenberger, M.; Fütterer, R.; Ziegler, F.; Schellhorn, M.: "Nearfield sensing and actuation for multispectral imaging systems", In: 2015 IEEE Sensors Applications Symposium (SAS 2015), proceedings, Piscataway (USA) : IEEE, 2015.
[5] Rosenberger, M.; Linß, G.: "Multispectral image correction for geometric measurements" In: Joint IMEKO TC1-TC7-TC13 Symposium: Measurement Science Behind Safety and Security, proceedings, page 231-236, Red Hook (USA): Curran, 2015.
[6] Rosenberger, M.; Preißler, M.; Fütterer, R.; Zhang, C.; Celestre, R.; Notni, G.: "Development and characterization of a high speed linear-moving-stage for multispectral measurements", In: 2016 Joint IMEKO TC1-TC7-TC13 Symposium: metrology across the sciences: wishful thinking?, proceedings, Bristol (USA): IOP Publishing, 2016.
[7] Smith, L. I. "A tutorial on principal components analysis.", Cornell University, Ithaca (USA): 2002. .
[8] Parlett, B. N.: The Symmetric Eigenvalue Problem. Prentice-Hall Series in Computational Mathematics, Englewood Cliffs (USA): Prentice-Hall, Inc. 1980.
[9] Francis, J. G. F.: "The QR transformation: A unitary analogue to the LR transformation", Parts I and II, The Computer Journal Issue 4 1961, pp. 265–272 and pp. 332–345, Oxford (UK): Oxford University Press, 1961.

CONTACTS

Dipl.-Ing. M. Schellhorn	mathias.schellhorn@tu-ilmenau.de
Dipl.-Ing. R. Fütterer	richard.fuetterer@tu-ilmenau.de
Dr.-Ing. M. Rosenberger	maik.rosenberger@tu-ilmenau.de
Univ.-Prof. Dr. rer. nat. G. Notni	gunther.notni@tu-ilmenau.de