DOI 10.5162/sensor2013/C5.3

Imaging Sensor with Integrated Feature Extraction Using Connected Component Labeling

M. Klaiber, S. Ahmed, M. Najmabadi, Y.Baroud, W. Li, S.Simon
Institute for Parallel and Distributed Systems, Department of Parallel Systems
Universitätsstr. 38, 70569 Stuttgart, Germany
Michael.Klaiber@ipvs.uni-stuttgart.de

Abstract

High-speed imaging is one of the key methods in a wide range of applications to understand basic physical processes. In traditional high-speed imaging a certain number of images is taken and stored in a high-speed memory directly attached to the image sensor of the imaging device. However the size of this memory limits the number of frames or the image size which can be capture. In this proposal a real-time processing of image data on a reconfigurable logic device, which is directly connected to the image sensor is proposed. This architecture is able to process a very high-speed pixel stream without the need of storing full images at all and is able to extract object feature in real-time. The proposed FPGA image processing architecture uses a combination of two recently proposed single pass algorithm for the task of feature extraction and is able to process up to several 1000 frames per second.

Key words: Connected Component Labeling**,** Parallel Image Processing Architecture, FPGA, Smart Camera, Image Feature Extraction

Introduction

High-speed imaging has a wide field of application which expands from monitoring of manufacturing processes and quality inspection in general to measurement techniques like particle velocimetry [1-3]. In this context a smart camera system with integrated processing capabilities is introduced for online characterization of image objects in image data streams with several 100 or several 1000 frames per second and a data bandwidth of several Gbit/s.

The required algorithm was implemented in parallel on a field programmable gate array (FPGA) which is connected to the image sensor of the camera. Since the pixel stream is processed inside the camera and only relevant information is passed on, real-time measurements with high frame rates are feasible. From raw pixel data acquisition to the final calculation of the objects' characteristics, all necessary steps are done in a single pass without the need of storing the image. Thus the resulting hardware architecture ensures high performance with low memory usage and the ability to acquire the objects' features in a single pass. In order to achieve a high frame rate for high-speed imaging a scalable parallel architecture for feature extraction based on connected component labeling (CCL) was developed. Contrary to classical connected component labeling algorithms, where the image has to be scanned at least two times, the proposed approach is able to perform the processing in only a single pass. Therefore the amount of memory required for this algorithm is not as high as for storing a full image, and can therefore be reduced by one order of magnitude to only a single image row. Due to the memory reduction, it is possible to dispense with external memory. This is an important requirement in order to accomplish an efficient implementation on FPGAs.

The application of common single pass algorithms to hardware resources on FPGAs often lead to low processing speeds of one pixel per cycle, at best. Therefore, a scalable parallel memory-efficient algorithm for connected component labeling is used to counteract that effect and improve the performance. Compared with other parallel connected component labeling algorithms, the used algorithm reduces the memory requirements of the underlying hardware architecture, by a factor 100 or more for common image sizes. In comparison to other architectures, this method increases the possible number of simultaneously processed image slices. A processing throughput of

4.5GPixel can be achieved on mid-range FPGAs. Furthermore, the architecture is suitable for stream processing, a necessary feature for real-time image and video processing.

Related work

The algorithm which is used for the extraction of features in the image taken by the image sensors is connected component labeling (CCL). This class of algorithms has been continuously improved from its early introduction as a two-pass algorithm to more sophisticated variants.
Traditional algorithms are based on two scans of the same image and therefore a high amount of storage and memory is required [5]. For the storage of data such as region labels, it is essential to implement a memory matrix with the same size as the image to be processed. The labels are determined by the position of the pixel within the image. If the pixel is located in the background area of the image a special background label is used. In all other cases, the labels of adjacent pixels need to be included in the assessment to determine the current pixel's labels. The CCL algorithm in [4] was one of the first FPGA based single pass architectures. Its memory requirements in the worst case are depending on the number of objects and are therefore related to the height and width of the processed image. The lack of parallelism and the limitation of the architecture unable to process more than one pixel per clock cycle could become a performance bottleneck in some cases. Due to the reuse of the labels Ma et al. managed to reduce the memory requirements [6] of the algorithm described in [7]. This improvement led to an enormous reduction of the memory requirements of the algorithm. The memory needed now corresponds only to the width of the processed image. In proposing a parallel architecture for CCL, Kumar et al. enhances the single pass algorithm used in [8]. His method stores whole images in prior to processing. In order to increase the performance, different CCL units have to independently process several slices of the image. In order to do so, one line at a time is fetched sequentially from memory in a round robin fashion. As a result, each CCL unit will pass a vector to a global FIFO memory. The vector mainly describes the regions of the processed image slice excluding the edges, the FIFO memory acts as a vector collector. A coalescing unit (CU) processes one or both borders of an image slice and determines whether two regions of adjacent slices are connected and therefore belong to the same object. If they do, the feature vectors will then be merged by the CU. For the merge operation it is crucial to connect the CCL units to the CU and perform the merging operation of neighboring image slices in a round-robin manner. Furthermore each edge region needs to have a unique, distinguishable label in its image slice for the merging process to be successful. This means that this approach has to store the full image in prior to parallel processing and needs to consider the worst case number of objects depending on the image width and image height. In order to cope with this a modified version of [6] which has two types of labels – slice local label and slice global labels – which reduced the amount of memory necessary for processing drastically, was proposed [9]. In this proposal a coalescing unit was introduced which has the ability to merge results of image slices processed with the modified CCL architecture on the fly and can therefore be used for real-time processing in a memory-efficient way even for images with the maximum number of objects.

System architecture

The overall system is composed of a high-speed image sensor and a high performance reconfigurable logic device – a field programmable gate array (FPGA). As shown in Figure 1 the image sensor is connected via several high speed connections to the FPGA due to its high bandwidth. This system, also referred to as smart camera, is able to capture images with a very high frame rate and is able to process the received pixel data in real-time. Therefore this smart camera system can be equipped with integrated processing capabilities such as feature extraction. At first the pixel data of each image taken is sent to the FPGA. The following processing steps, which enable a massive data reduction from GBytes of Pixel data to only KBytes of abstract object descriptors - so called feature vectors, are explained in detail in the next section. This reduction enables the smart camera system to output information on every object in every frame even for very high speed imaging in real-time. The amount of data which has to be transferred from the smart camera system is reduced by several orders of magnitude in this way.

Additionally control signals, e.g. for step motors or any other kind or actuators, can be generated in real-time. By feeding this control signal with a minimal delay directly to the corresponding actuators even a control loop system can be established.

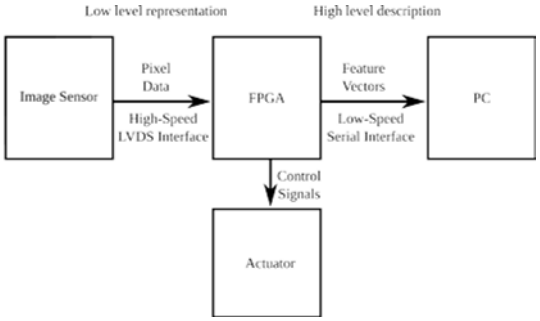

Fig. 1: Block diagram showing the connection of the sensor system with peripheral devices.

Image processing architecture

The image processing architecture realized on the FPGA has the tasks of image segmentation and feature extraction. For segmentation, which is the process of separating the image background from the objects visible in the image, thresholding is applied. Using this method all pixels having an intensity value over a certain threshold are converted to black and all pixels below this threshold are converted to white. In this way a binary image is generated from the original grayscale image. The threshold value for separating the objects from the background is in this case of major importance. If it is chosen wrong, objects could be considered to be background (see Figure 4), or background could be considered to be a foreground object (see Figure 3). Only the use of an appropriate threshold value can lead to a reasonable segmentation and therefore has great influence on further steps in the image processing architecture.

In the proposed architecture the method proposed in [10] is used. It generates a histogram for the captured grayscale image, as seen in Figure 2. The threshold value is calculated by using the arithmetic mean value of the two peaks detected in the histogram, representing the objects and the background. The calculated threshold value is used for the segmentation.

For the process of feature extraction the main challenge is dealing with the high image frame rates which result in high bandwidth to process in the range of 10 Gbit/s or more and process them in real-time. For this reason a highly optimized and sophisticated architecture has to be used in order to achieve the high throughput which is necessary. Because of the limited number of resources available on FPGAs, algorithms which are specially dedicated to the FPGA's architecture have been proposed [4,6,7,9].

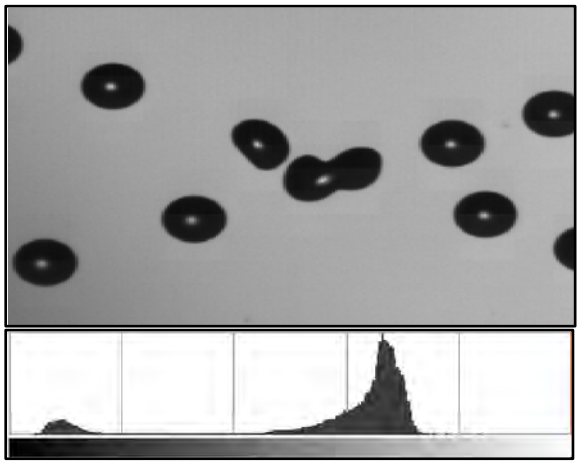

Fig. 2: Grayscale raw image taken by image sensor and its histogram.

Fig. 3: Segmentation of grayscale image with threshold value, which is too low

Fig. 4: Segmentation of grayscale image with threshold value, which is too high

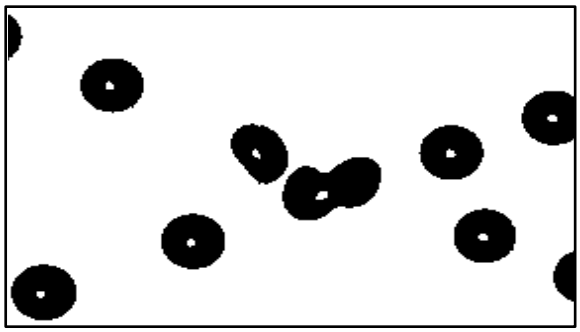

Fig. 5: Segmentation of grayscale image with calculated threshold value.

For our image processing architecture we chose the connected component labeling architecture proposed by Ma [6] along with an extension for parallel processing [9]. The architecture proposed by Bailey is depicted in Figure 6. It consists of several components which are described as follows: The neighborhood context block consists of four registers A, B, C and D containing 4 pixels of the eight-neighborhood of the currently processed pixel. The labels processed in the neighborhood context block, which works in a similar way as a window filter forwards its pixel labels from the currently processed line to the row buffer and receives pixel labels from the previous row from the translation table. The fact that labels are reused in every row, which is the key difference from [4] to [6] is the major reason for the reduction in the need for memory. If there are two different labels in the neighborhood context block after label translation, the regions belonging to these two labels have to be merged. This merger is recorded in the merger table. Since there can be several mergers, which have data dependencies, label pairs to be merged are pushed on a stack and evaluated at the end of the row. When the end of a row is reached, the content of the stack is read in reverse direction it was filled. In this way the merger table can be updated and all data dependencies for the mergers which occurred in the current row are taken into account. The data merging unit records the features of each region by monitoring the labels of the pixels in the neighborhood context block. Each region has one entry in the data merging unit indexed by the region's label. Whenever a region is updated, its entry in the data merging unit is updated as well.

In [4,6] the author described how to extract the area information for each object. In order to extract the bounding box for each object the structure and the merging process within the data merging unit has to be changed. The bounding box is defined by two coordinates A and B. Coordinate A represents the upper left corner with lowest x and lowest y value of any contained pixel and coordinate B the lower right corner with highest x and highest y value of any pixel of the object. In order to extract the bounding box as well as the area or several other object features, the data merging unit has to be changed as shown in Figure 7. It has two inputs – one for giving information on the neighbor pixels label and one providing the currently processed pixel's coordinates.

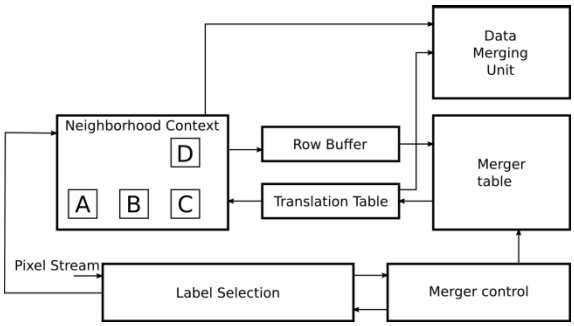

Fig. 6: Connected Component Labeling Architecture proposed in [6].

For each object feature which should be extracted, a merging unit as well as a data table is necessary. Each merging unit has two inputs, a and b, and one output c. The input data is read from the corresponding data table. In order to find the bounding box for two entries of the data table the following equations are used.

$$x_{1c} = \min(x_{1a}, x_{1b}) \quad (1)$$

$$y_{1c} = min(y_{1a}, y_{1b}) \quad (2)$$

$$x_{2c} = \max(x_{2a}, x_{2b}) \quad (3)$$

$$y_{2c} = \max(y_{2a}, y_{2b}) \quad (4)$$

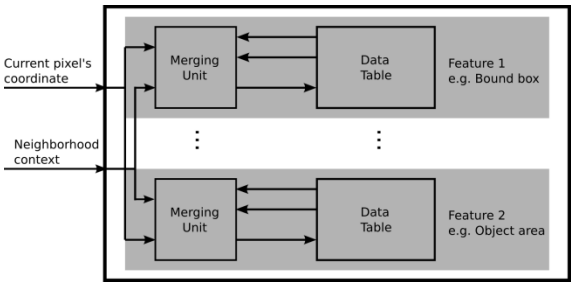

Fig. 7: Data Merging Unit for extracting several features of the image objects simultaneously.

In order to deal with a very high throughput a parallelization approach is used as described in [9]. Therefore the image is cut into several slices before feeding to the CCL architectures and each image slice is processed in parallel. After they have successfully identified objects in their corresponding slice, the objects are gathered by a central unit called the coalescing unit and merged together. In this way several pixels can be processed in parallel and a speedup according to the number of image slices can be achieved. The processing steps for feature extraction are shown in Figure 8 through Figure 10. Figure 2 shows the raw image taken by the image sensor. In order to get a binary image which is needed for the further processing steps, the aforementioned

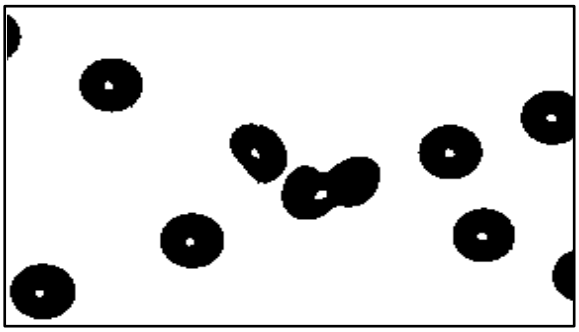

Fig. 8: Binarized image after thresholding.

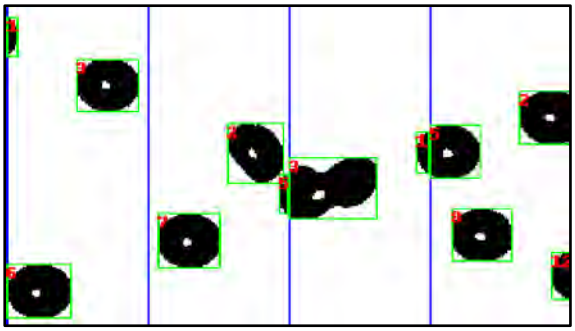

Fig. 9: Extracted object features for all sub-images.

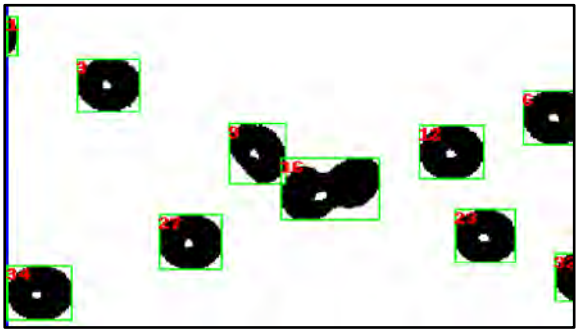

Fig. 10: Merged object features for input image.

image segmentation techniques are applied. The results of them are given in Figure 8. After binarization, the image is cut into several image slices for processing in CCL processing units which are working in parallel. As depicted in Figure 9, each CCL processing unit extracts the bounding box of the objects in its image slice. Finally the extracted features of the objects touching an image stripe's border have to be merged in order to get correct results for the objects, which are separated by the image stripe's borders. This process is illustrated in Figure 10.

Experimental Results

The need for obtaining a higher processing bandwidth than possible with the implementation described in [6] leads to a modified version of this architecture together with the coalescing unit proposed in [9]. The FPGA resource utilization of the proposed changed CCL unit is given in Table 1. In order to achieve the goal of extracting the bounding box of each image object, the data table of the original architecture had to be changed accordingly. Implemented on the Xilinx Virtex 6 XC6VLX240T FPGA device a maximum frequency of 128.9 MHz can be reached, which leads to a processing bandwidth of approximately 100 MPixels/s. In order to achieve a higher bandwidth the stripping approach of [9] was applied using several of the mentioned CCL units. Depending on the number of slices for parallel processing and on the image size the coalescing unit proposed in [9] can achieve a processing rate of several thousand frames per second for typical image sizes. The diagrams given in Figure 11 through 13 show a detailed overview on the frame processing rate for different image sizes for the coalescing unit. The FPGA utilization of the coalescing unit is mainly dominated by the number of image slices which are processed in parallel and therefore the more image slices are processed in parallel, the more FPGA resources are needed.

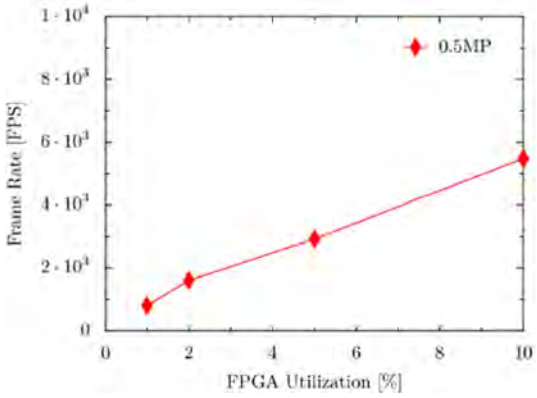

Fig. 11: Achievable frame rate for an image with 0.5 Megapixels.

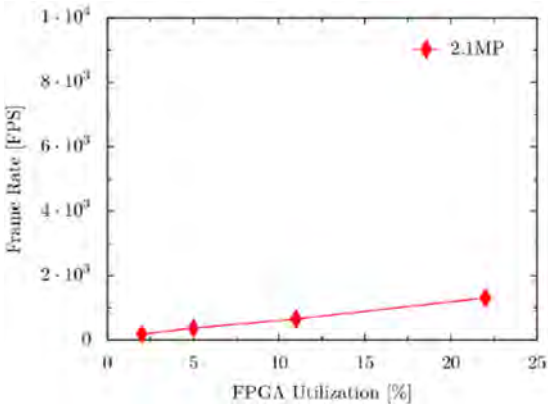

Fig. 12: Achievable frame rate for an image with 2.1 Megapixels.

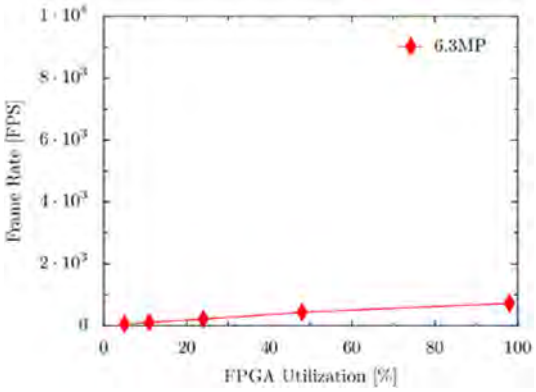

Fig. 13: Achievable frame rate for an image with 6.3 Megapixels.

Tab. 1: FPGA utilization and maximum frequency for implementation of the CCL architecture using Xilinx Virtex 6 XC6VLX240T FPGA

CCL architecture capable of processing an image of 1024 x 512 pixels			
Registers	1665	of 301k	<1%
LUTs	4235	of 150k	2%
Block RAMs	5	of 416	1%
f_{max}	128.9 MHz		

Conclusion

High speed imaging is an essential measurement technique for many applications. In order to deal with frame rates in the range of several thousand frames per second an imaging sensor with integrated feature extraction capabilities was proposed. By the use of a field programmable gate array, which contains a high-speed image processing architecture, the pixel data given by the image sensor can be processed in real-time and object features can be output with a minimal latency. The proposed image processing architecture is based on the connected component labeling algorithm and uses a combination of two recently proposed architectures to acquire the desired throughput. Using this single pass architecture the real-time feature extraction with several thousand frames per second is possible.

Acknowledgements

The authors would like to thank the German Research Foundation (DFG) for the financial support. This work has been carried out within the research project Si 586 7/1 which belongs to the priority program DFG-SPP 1423 “Prozess- Spray”.

The authors also would like to thank the working group of Prof. Sommerfeld, MVT Universität Halle-Wittenberg for providing images of their experiments.

References

[1] S. T. Thoroddsen, T. G. Etoh, and K. Takehara. High-Speed imaging of drops and bubbles. Annual Review of Fluid Mechanics, 40(1):257–285, 2008.

[2] Rao, K., Winterbone, D., and Clough, E., "Laser Illuminated Photographic Studies of the Spray and Combustion Phenomena in a Small High Speed DI Diesel Engine", SAE Technical Paper 922203, 1992

[3] Y.-C. Cho, "Digital image velocimetry," Appl. Opt. 28, 740-748,1989.

[4] D. Bailey and C. Johnston, “Single pass connected components analysis,” in Proceedings of Image and Vision Computing New Zealand 2007, dec. 2007, pp. 282–287.

[5] A. Rosenfeld and J. L. Pfaltz, “Sequential operations in digital picture processing,” J. ACM, vol. 13, pp. 471–494, October 1966.

[6] N. Ma, D. Bailey, and C. Johnston, “Optimised single pass connected components analysis,” in ICECE Technology, 2008. FPT 2008. International Conference on, dec. 2008, pp. 185 –192.

[7] D. Bailey, C. Johnston, and N. Ma, “Connected components analysis of streamed images,” in Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on, sept. 2008, pp. 679 –682.

[8] V. Kumar, K. Irick, A. Al Maashri, and N. Vijaykrishnan, “A scalable bandwidth aware architecture for connected component labeling,” in VLSI ISVLSI), 2010 IEEE Computer Society Annual Symposium on, july 2010, pp. 116 –121.

[9] M. Klaiber; L. Rockstroh; Z. Wang; Y. Baroud , S. Simon "A Memory-Efficient Parallel Single Pass Architecture for Connected Component Labeling of Streamed Images" 2012 International Conference on Field-Programmable Technology (FPT), Seoul, Korea, Dec. 2012.

[10] R. C. Gonzalez and R. E. Woods. Digital Image Processing (3rd Edition). Prentice Hall, August 2007, pp.741-761 , ISBN 013168728X.