A HARDWARE-ACCELERATED REAL-TIME IMAGE PROCESSING CONCEPT FOR HIGH-RESOLUTION EO SENSORS

Stephan Blokzyl and Matthias Vodel and Wolfram Hardt
Department of Computer Science
Computer Engineering
Chemnitz University of Technology
09107 Chemnitz, Germany
{stephan.blokzyl,matthias.vodel,wolfram.hardt}@cs.tu-chemnitz.de

Abstract

Increasing the level of autonomy of systems demands confident controlling and task management units. To ensure a trusted system operation, several core capabilities have to be fulfilled: reliable sensing abilities, efficient data processing, and well-organised information dissemination.
Dependent on the field of application, different types of sensors are required to meet the given operational tasks. In context of pattern recognition and object surveillance scenarios, electro-optical (EO) sensors offer superior sensing capabilities. Regarding to processing of high-resolution image data, real-time aspects represent one of the most challenging issues, especially in the domain of resource-limited, embedded systems.
This paper presents a novel concept for hardware-accelerated computation of high-resolution EO sensor data using FPGAs (Field Programmable Gate Arrays). The concept focuses a complete integration of the image processing chain. Reconfigurable FPGA technologies combine the flexibility of general-purpose processors with the advantages of application-specific integrated circuits. We introduce two data processing approaches that utilise specific FPGA capabilities: data and task parallelisation. Data parallelisation reduces the amount of data to be treated by a discrete processing entity. Task parallelisation concatenates weak pattern detection methods to a strong detector. These strategies, used separately or combined, enable the conversion of sequential image processing chains to parallelised hardware design.
The concepts in this paper improve the confidence of pattern recognition results significantly. At the same time, the computation speed increases, especially in comparison to microcontroller based processing units. This allows an energy-efficient realisation of complex high-resolution image processing tasks in resource-limited, embedded environments.

Index Terms Hardware-acceleration, real-time image processing, embedded image processing, pattern recognition, parallelised hardware design, reconfigurable hardware, data parallelisation, task parallelisation, high-resolution EO Sensors, FPGA, Fuzzy fusion, Fuzzy logic

1. INTRODUCTION

Automated systems interact bi-directionally with the environment, in which they are operating. They affect the environmental state by executing actions and percept various characteristics and dynamics of the system's adjacencies [1]. Sensor data establishes the basis for all control, actuation and decision processes of an automated system. Sensor data quality has a major impact on the robustness of the world model (belief) and the reliability of system function. These considerations lead to important capabilities of automated systems:

- Robust sensing devices for environmental exploration
- Efficient, flexible (real-time) data exploitation
- High-capacity communication concepts

Depending on the application, automated systems are equipped with complex, highly interconnected sensor systems, which acquire a huge amount of heterogeneous data. In the context of pattern and object recognition in surveillance applications, electro-optical (EO) sensors are the most suitable choice. They are affordable and provide a huge amount of good-quality, high-resolution image data.

The management and exploitation of image data in a limited, finite time frame (real-time) is a crucial challenge, especially in resource-limited, embedded systems. E.g. the processing of Full-HD colour images (1920x1080 pixel resolution) with a frame rate of 30 frames per second needs over 186 million operations per second if only one operation per pixel and channel has to be executed (compare [2]). In other words - more than 1GB data needs to be processed within less than six seconds. The executional costs increase non-linearly with rising complexity of used methods or with aggregation of different image processing methods.

2. STATE OF THE ART

Hardware-based real-time image processing has been focussed by multiple research groups worldwide. Implementations for different low-end applications [3] such as barrel distortion correction [2] or scratch detection [4] were presented during the last decade. In addition to the acceleration of dedicated image processing methods, hardware has also been used to speed up other steps of the image processing chain, e.g. for classification of objects [5] [6] or faces [7].

This is only a brief extract of a wide range of applications realised by hardware-based image processing solutions. But all papers referenced above map only a share of the whole system functionality on hardware. The concept presented in this paper supports full hardware integration of the complete image processing chain including image data acquisition, processing, and result dissemination.

3. HARDWARE-ACCELERATED IMAGE PROCESSING

Due to high computational costs of high-resolution image data exploitation, general-purpose-processors, as utilised in today's automotive or avionic domain, are not capable anymore to fulfil the requirements stated above (compare [8] [9]). Application-specific, embedded processing technologies shall be introduced, which offer powerful computing capabilities to real-time applications. Primarily reconfigurable, integrated circuits like Field Programmable Gate Arrays (FPGAs) provide efficient, reliable high-performance-computing with a maximum of flexibility and scalability. FPGAs have a slightly less logic density compared to application-specific integrated circuits (ASICs) but they combine the flexibility of multi-purpose-processors with the outstanding processing speed of ASICs.

FPGAs consist of free configurable logic blocks (CLBs), which were meshed by a programmable switching network. They accommodate the application-specific processing modules including all data- and control-paths. This allows an optimisation of processing architecture with the respect to the functional and timing requirements of the underlying application. Together with an unlimited diversity of configurable serial and parallel on-chip-interfaces, FPGAs are qualified for applications in which high-performance-computing is required.

Custom EO sensor devices are apparelled with a multitude of interfaces (e.g. Ethernet, FireWire and USB) and standards (e.g. GigE Vision, Camera Link) for image data retrieving. Additionally all peripheral units, subscribing to image processing results communicate via arbitrary interfaces. Both sensor data acquisition and result dissemination function is provided by stand-alone IP-cores that are uncoupled from the image processing part.

Generally an image processing module comprises data acquisition, data exploitation and result dissemination. A real-time system allows a maximum processing duration less than the reciprocal of camera sensor frame rate. Therefore the sum of data acquisition time, data exploitation time and result dissemination time is a determining factor. The computational effort of data acquisition and result dissemination is assessable. But the infinite diversity of image data, high complexity of image processing algorithms and their sequential character cause a high computational effort and minor predictability of processing duration.

4. SYSTEM DESIGN CONCEPT

To enable a system to process a huge amount of image data in real-time, two processing strategies are suitable to maximise the profit of reconfigurable hardware: data parallelisation and task parallelisation.

4.1. Data Parallelisation

Data parallelisation splits the large-scale image data input into data subsets, which were processed simultaneously by multiple uniform processing modules. This strategy speeds up a single processing step by introducing a workload distribution (compare Fig. 1).

Each processing instance receives a share of the input image, which has been split prior (split step). The partial results were reassembled (merge step) to the final result after processing of each module has been completed.

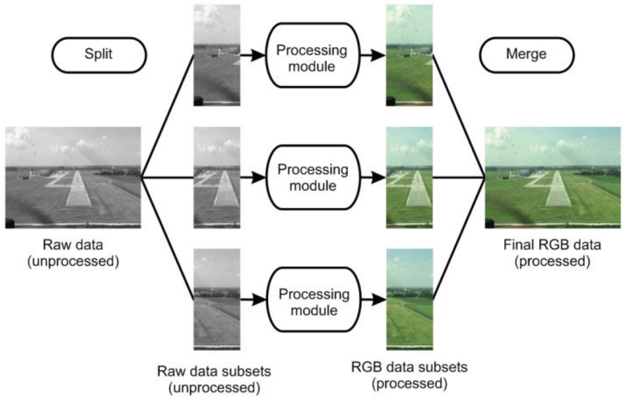


Fig. 1: Data parallelisation (here: Debayering)

4.2. Task Parallelisation

The task parallelisation utilises a set of heterogeneous image processing modules. Each module receives image data and executes simultaneously different low-level image processing methods.

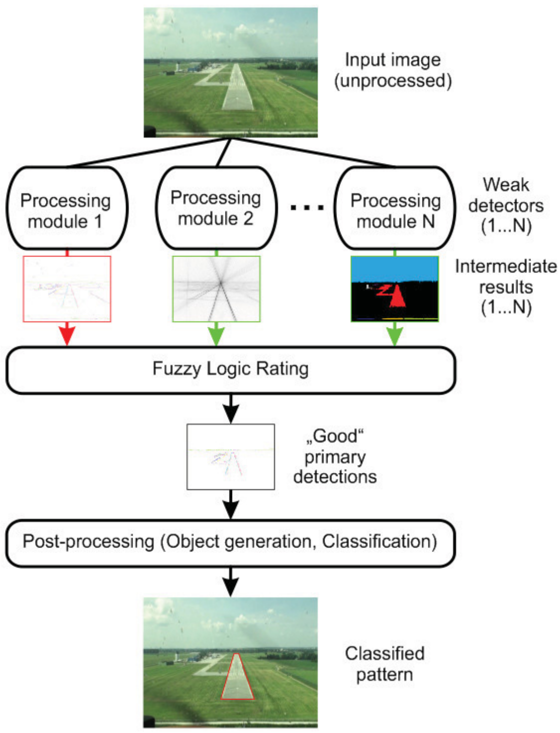


Fig. 2: Task parallelisation (N parallel processing modules)

The low-level image processing modules (weak detectors) generate heterogeneous intermediate results, which are weak and unsorted referenced to the global detection goal. It is unknown at this stage, whether the detections belong to the wanted object (true positives) or not (false positives). For this reason the intermediate results comprise primary detections (red) and various, heterogeneous secondary detections (green, see Fig. 2). The primary detections are the most significantly characterizing low-level detections due to the global recognition task. They are strongly application-specific and need to be chosen conscientiously. The secondary detections are used to estimate the potential of the primary detections. The potential denotes the detection's probability of belonging to the wanted object. The primary detector has an enhanced detection sensitivity to avoid loss of true positives.

All primary and secondary detections are forwarded to a Fuzzy Logic Rating layer (FLR). The FLR evaluates the primary detections involving all secondary ones. It aggregates intermediate results generated by various processing modules.

The primary detections rated with a minimum potential value are forwarded to the post-processing. It generates object candidates based on the knowledge of the searched pattern (geometric characteristics) and classifies the candidates into wanted objects and rest.

Both data and task parallelisation strategy incorporate into the real-time image processing concept. Data parallelisation is used within the different image processing modules of task parallelisation (see Fig. 2).

4.3. Image Processing Modules

Task parallelisation strategy for image processing on programmable hardware utilises a high degree of parallelisation. Discrete, independent picture elements (pixels) can be computed simultaneously using logic instead of computing them serially as done by general-purpose-processors. E.g. an exposure correction, which is a multiplication of each pixel value with a fixed factor, is a location-independent operation that can be executed on all image pixels simultaneously. The computational speed-up increases proportional to the number of pixels computed parallelly and the image resolution.

To maximise the benefit of reconfigurable hardware, mainly low-level detectors shall be utilised. Their elementary algorithms don't use iterations or recursions and their scale depends on source image size only. The processing time of low-level image processing modules can be determined explicitly. They reduce implementation complexity and minimize logic consumption of the FPGA, which is need for a maximum of processing parallelism.

Hardware implementations suitable low-level processors could be separated into pixel-, edges-, model-, region-, texture-based, and interest point detection methods. Pixel-based methods assume pixels with intensity or colour greater than a threshold as foreground and the remaining as background. Edges-based techniques extract high-frequency components of the image data by convolving the source image with a derivation kernel. Model-based pattern extraction methods base on the knowledge of unique characteristics of the wanted pattern. A well-known example is the Hough transformation for line extraction. Region- and texture-based methods detect segments of coherent pixels with similar attributes (intensity, colour or texture) and interest point detectors educe significant points like e.g. SIFT key points [10] or corners.

4.4. Fuzzy Logic Rating

The Fuzzy Logic Rating layer receives all primary and secondary detections from the low-level image processing modules. It evaluates all primary detections involving all secondary ones based on a Fuzzy logic approach. The FLR combines heterogeneous intermediates of common source data. This separates this approach from conventional fusion systems that combine sensor data from different types of sensors. The major advantage of the FLR is the aggregation of different weak detectors to a strong detector, producing confidential, robust detections with higher detection robustness.

Fuzzy logic is a form of probabilistic logic theory that uses blurred descriptions to decide whether a featured element belongs to a set or not. It uses parameterised membership functions of e.g. exponential (1) or potential type (2). The function $\mu(x)$ indicates the probability of set membership of an element with feature x. The function parameters determine feature expectation m, deviation tolerance c and sharpness d [11].

$$\mu_e(x) = \mathrm{e}^{\left(\frac{|m-x|}{c}\right)^d} \tag{1}$$

$$\mu_p(x) = \frac{1}{1+\left(\frac{|m-x|}{c}\right)^d} \tag{2}$$

The benchmark μ is in the interval $[0;1]$, while 0 means the featured element does not belong to a set and 1 indicates a certain membership. The decision threshold is defined at 0.5.

The potential function type μ_p proved its suitability in a wide range of technical and non-technical applications. It has advantageous mathematical properties as derived in [11]. Due to that fact all potential estimations in this paper base on potential type μ_p.

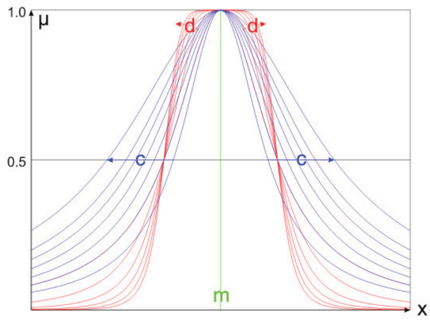


Fig. 3: One-dimensional potential function μ_p

In general applications, elements are characterized by more than one feature. They are described by a feature vector with dimension n. In that case the *Modified Hamacher* operator is used to treat all features of the n-dimensional feature vector $\underline{x} = (x_1, x_2, \dots x_n)^T$. The potentials $\mu(x_i)$ of each feature x_i are linked by

$$\mu_p(\underline{x}) = \frac{n}{\sum_{i=1}^{n}\left({}^{1}/_{\mu_{p_i}(x_i)}\right)} \quad . \tag{3}$$

The *Modified Hamacher* operator (3) is used to evaluate all elements of the primary low-level detections. Based on the knowledge of the global detection goal, intelligent unique features have to be developed applying the secondary intermediates. Considering e.g. a set of edge points as primary detections, the edge point features are:

- Magnitude
- Orientation
- Location in image space
- Intensity or colour in source image

It is not possible to decide if an edge point belongs to a wanted object by analysing the features above. Introducing more knowledge and a second low-level detector, the information content of an edge point can be enhanced. Imagine the searched object has strong boundaries that are easily recognisable by a Hough transformation-based detector. The resulting Hough lines are used to determine edge point's probability of belonging to a searched object. Edge points near a Hough line feature a higher probability compared to edges far away from Hough line.

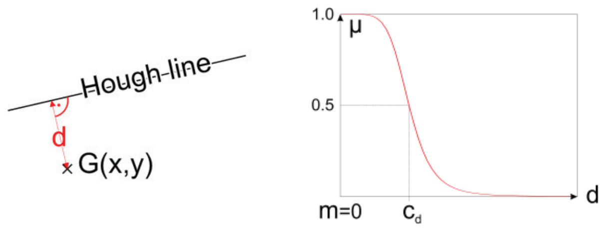


Fig. 4: Orthogonal distance d between edge point $G(x,y)$ and Hough line (left) and appropriate potential function (right)

The right graph in Fig. 4 shows the falling probability for edges $G_i(x,y)$ with increasing orthogonal distance d from Hough line. The potential function is given as

$$\mu_p(d) = \frac{1}{1 + \left(\frac{\mathrm{d}}{c_d}\right)^{d_d}} \quad . \tag{4}$$

Not all secondary intermediate results are true positive detections. The approach reinforces false positive primary intermediates as well. To cope with that problem additional secondary detectors are introduced that conduce to the application's detection objective. The use of numerous different weak detectors corporates with the good parallelisation capabilities of reconfigurable hardware. An arbitrary diversity of parallel low-level processors can be configured in addiction to application and the available hardware resources. The heterogeneous results of secondary detectors contribute dimensions of the primary detection's feature vector $\underline{x} = (d_h, d_r)^T$ (e.g. distance to nearest Hough line d_h, distance to nearest region d_r). Each entry in $\underline{x}$ determines a potential $\mu_{p_i}(x_i)$ by equation (2) and all potentials are linked by (3) to an overall potential $\mu_p(\underline{x})$.

$$\underline{x} = \begin{pmatrix} d_h \\ d_r \\ \vdots \\ x_n \end{pmatrix} \overset{(2)}{\rightarrow} \underline{\mu}(x_i) = \begin{pmatrix} \mu_{p_1}(d_h) \\ \mu_{p_2}(d_r) \\ \vdots \\ \mu_{p_n}(x_n) \end{pmatrix} \overset{(3)}{\rightarrow} \mu_p(\underline{x}) \tag{5}$$

The primary detections with overall potentials $\mu_p(\underline{x})$ greater equal the decision threshold are forwarded to the post-processing step while the remaining are rejected (compare "Good" primary detections in Fig. 2).

4.5. Post-Processing

The post-processing step treats all "good" rated primary detections of the task parallelisation. Its design strongly depends on the application and generally comprises

- Feature estimation
- Classification and
- Scene interpretation.

If the low-level primary results are not directly interpretable regarding the global detection goal, an optional object reconstruction step might be necessary prior feature estimation. Typical features describing a segment are e.g. translation-/rotation-/scale-invariant moments, contour code, width, height, volume, and compactness.

Object features depend on position and orientation of both sensing system and the observed object. Rotation, translation and EO sensor's lens imperfects cause distortion in image space. The application of in- and extrinsic EO sensor parameter solves this problem. With their help all measurements in image are transformed to world and become independent from translation and rotation between sensing system and observed object. The classification robustness in world space increases significantly.

Multiple features of one object are combined to a feature set that represents a searched object candidate. All feature sets are forwarded to a ready-trained classifier, which arranges all candidates to different classes like wanted objects (positives) and rest (negatives). All positive classified objects correspond to successfully recognised objects of the target class.

Next, object's state (e.g. position, orientation) is computed (scene interpretation). Introducing multi-frame processing, dynamic properties like velocity or angular velocity of the positive classified objects can be predicted. Multi-frame processing uses more than one image frame and tracks detections over discrete time steps. The application of stochastic filters like e.g. Kalman filter can reduce noise and increase detection accuracy [1].

The next chapter introduces an example of this concept successfully applied to a practical recognition problem.

5. EXAMPLE APPLICATION

The aviation domain is an adequate field to demonstrate the practicability of embedded real-time image processing. The automation of flight-effecting functions based on visual sensors is very time critical and demands efficient, real-time sensor data exploitation.

The chosen example application implements a runway recognition function for an aircraft autoland system. The system operates in an automated, unmanned aerial vehicle during final approach phase.

5.1. Image Processing Modules

The EO sensor provides an 8-bit grayscale image with each pixel characterised by an intensity value from 0 (black) to 255 (white). The following, typical properties are characterising runway representations in the test scenario:

- Almost homogeneous runway surface

- Surface is brighter than surrounding area
- Clearly recognisable runway boundaries
- Long left and right, parallel boundary pair
- Runway main orientation is vertical to horizon line

Following these considerations, the input image is processed by numerous processing modules. A blob detector searches for the homogeneous runway surface, and edge and borderline detectors are suitable for boundary detection. A horizon detector searches for the line separating earth surface from sky.

The blob detector for runway surface detection consists of three separate steps. A threshold step separates fore- and background objects and generates a binary image B. Threshold-based binarization is possible because of the homogeneous runway surface, which is much brighter than surrounding area. Second a morphologic closing (6) composed of dilatation $\oplus$ and erosion $\ominus$ with a circular structuring element K fills holes in all fore- and background segments (closed image B_{closed}).

$$B_{closed} = B \bullet K = (B \oplus K) \ominus K \tag{6}$$

Finally all detected foreground pixels are grouped to region segments. The runway surface detection results in a labelled image respectively a set of region segments defined by bordering points. The Fig. 6 (b) shows the detection results of the blob detector. The runway surface (orange) is well-extracted and labelled as a single region without internal gaps. Even the sky (image top, labelled green) is clearly separated from the earth ground. Challenging are connected taxiways left of the landing strip. They are identified as component of the runway and should be eliminated in final runway detection.

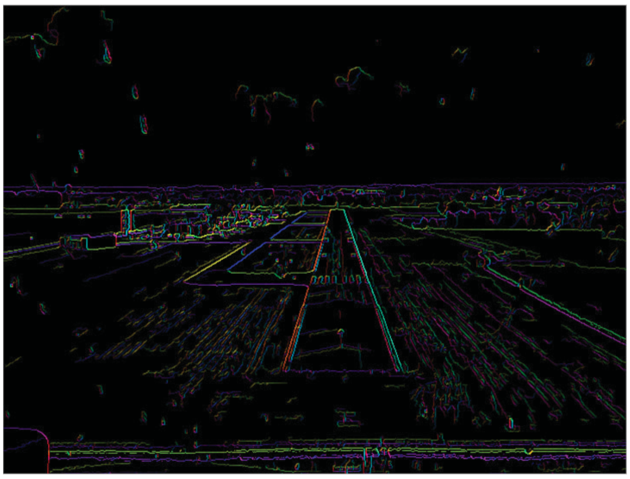


Fig. 5: Resulting edge image, the colour indicates the orientation and colour saturation figures gradient magnitude

The edge detector processes the image by convolving the source data S using two Sobel kernels $g_{x/y}$ (in x- and y-direction):

$$S(x,y) * g = \sum_{i=-1}^{1} \sum_{j=-1}^{1} g(i,j) S(x-i, y-j) \tag{7}$$

$$g_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \qquad g_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \tag{8}$$

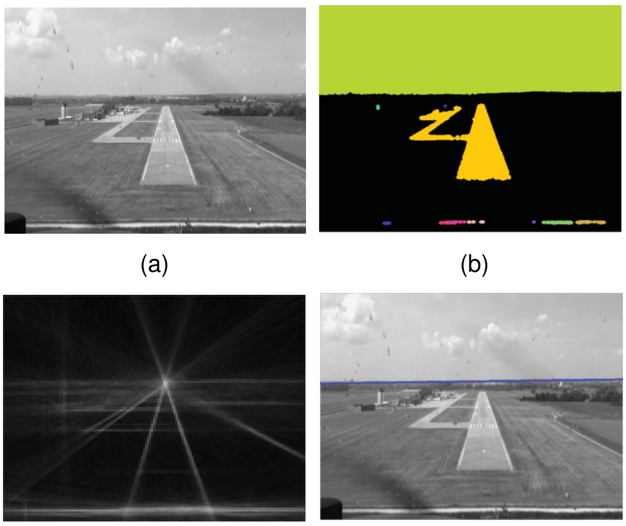
It generates a set of edge points with location, magnitude and orientation. The Fig. 5 approves the decision to declare edges as primary detections. The distinctive runway boundaries produce very significant, well-adjusted low-level detections characterising a runway in image space.

The Sobel edge detector is very robust but generates many false positives in addition to the real boundary edge points. Simple thresholding of the edge detections to eliminate points with lower magnitude is no appropriate solution. It causes loss of edges with lower response due to e.g. motion blur (see bottom edge line of the landing strip in Fig. 5).

The borderline detector is derived from the Hough line detection, but incorporates also edge point's gradient magnitude $G(x_E, y_E)$ as well its orientation $\theta(x_E, y_E)$. The value $G(x_E, y_E)$ is added to each accumulator point $(x_A, y_A)^T$ at:

$$\begin{pmatrix} x_A \\ y_A \end{pmatrix}(t) = \begin{pmatrix} x_E \\ y_E \end{pmatrix} \pm t \cdot \begin{pmatrix} \cos(\theta(x_E, y_E)) \\ \sin(\theta(x_E, y_E)) \end{pmatrix} \tag{9}$$

The length parameter t runs from base point $(x_E, y_E)^T$ to upper and lower accumulator space boundaries. The accumulator space is equally sized to image space. After transforming all edge points, the resulting accumulator space A is normalised with $\max(\{A(x_A, y_A) | \forall (x_A, y_A) \in A\})$, so that each point of A has a magnitude in the val $[0;1]$. The resulting parameter space indicates long edge lines with strong magnitude as show in Fig. 6 (c). White pixels indicate a high boundary rating and black pixel vice versa. The resulting accumulator space A contains location-depending boundary ratings and forms a look-up-table for all image points. Points with a higher rated location have a stronger plausibility to belong to a continuous edge line.



(a) (b)

(c) (d)

Fig. 6: Result images of (a) source image, (b) blob detector, (c) borderline detector, (d) horizon detector

The horizon detector separates earth from the sky that is much brighter than earth surface during day time. Furthermore, the sky is on image top if we assume normal aircraft operation in final approach phase. All pixels representing the sky are marked as foreground objects by blob detector. So horizon detection processes simultaneously the closed binary intermediate B_{closed} of the blob detector.

Starting with the first row (from image top to bottom) the first 0-1-discontinuity point is detected and stored. After processing all image columns, the horizon line slope m and intercept n are estimated by the least square method:

$$m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}; \quad n = \bar{y} - m\bar{x} \tag{10}$$

The horizon detection generates a linear horizon function of $y_h = f_h(x) = mx + n$ type (see blue line in Fig. 6 (d)). The orientation of the horizon in image space is $\theta_h = \tan^{-1}(m)$. The result of the horizon detector is robust in case of a two major requirements: The system operates during daytime at good weather conditions and no high buildings (e.g. in urban environment) or mountains are in the field of view. If these points are fulfilled, the horizon orientation is used to evaluate primary detection's orientation (see section 5.2). Furthermore, robust horizon detection can be used to gate the primary detections because detections above horizon line could not belong to the runway detection.

5.2. Intermediate Result Fusion

The in section 5.1 introduced image processing modules generate several low-level results separated into primary (a) and secondary intermediates (b-d):

(a) Edge points with location, orientation and magnitude
(b) Regions with border point locations
(c) Accumulator space with boundary ratings
(d) A horizon line given by intercept and orientation

Various features for each primary intermediate need to be determined prior primary detection rating on FLR layer (compare section 4.4).

Following the considerations in the previous section, all edges tight to region boundary (Fig. 7) have a higher potential μ_{p_r} to be part of the searched pattern, compared to edges located farther. The distance from region d_r is rated by equation (2).

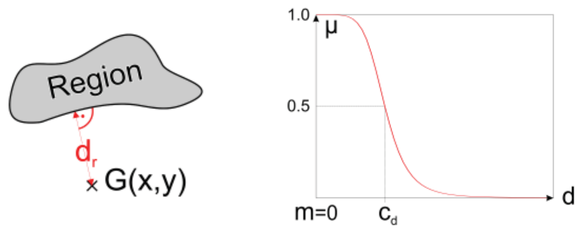


Fig. 7: Orthogonal distance d between edge point $G(x,y)$ and a border of a region (left); appropriate potential function (right, compare equation (4))

In addition to the region detections, the accumulator space A offers measurements to determine another feature: Edge points located on position with high boundary rating a_{xy} have a major potential μ_{p_a} to be part of the runway boundaries.

Due to the fact that the major runway orientation is vertical to horizon line, the orientation of all edge points representing the long runway boundaries shall be vertical to horizon's orientation ($\theta_h^* = \theta_E + 0.5\pi$). The orientation of short runway boundaries shall be similar to horizon orientation θ_E.

$$\mu_{p_h}(\theta_E) = \frac{1}{1 + \left(\frac{|\theta_h^* - \theta_E|}{c_\theta}\right)^{d_\theta}} + \frac{1}{1 + \left(\frac{|\theta_h - \theta_E|}{c_\theta}\right)^{d_\theta}} \tag{11}$$

All features contribute a dimension in the feature vector $\underline{x}$:

$$\underline{x} = \begin{pmatrix} d_r \\ a_{xy} \\ \theta_E \end{pmatrix} \xrightarrow{(2),(11)} \underline{\mu}(\underline{x}) = \begin{pmatrix} \mu_{p_r} \\ \mu_{p_a} \\ \mu_{p_h} \end{pmatrix} \overset{(3)}{\rightarrow} \mu_p(\underline{x}) \tag{12}$$

All edge points with overall potentials $\mu_p(\underline{x})$ greater equal the decision threshold 0.5 are forwarded to the post-processing step and the remaining are rejected. The Fig. 8 visualizes all "good" primary detections.

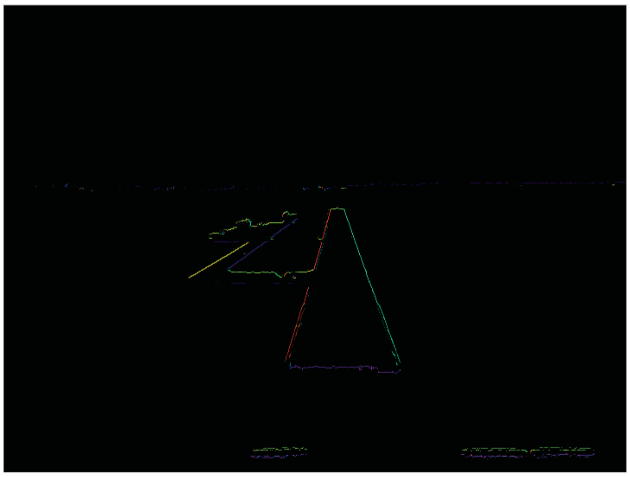


Fig. 8: Good primary detections forwarded to Post-processing step, the colour indicates the orientation and colour saturation the primary detection potential. There is a significant false positive reduction compared to Fig. 5

5.3. Post-processing

The post-processing step works on all edge points with an overall potential greater equal the decision threshold. All good rated primary detections have a high potential to be a part of the searched object. But edge points are not directly interpretable regarding the global detection goal of runway recognition. They need to be put in relationship by assembling them to more abstract geometries using specified combination rules. Therefore a *multi-level grouping approach* is used to detect objects with higher complexity in a set of lower-level structures. The runway boundary is a combination of simple shapes like lines and parallels. A line detector groups all adjacent edge points with similar gradient to a line segment. The resulting lines are combined to closed structures by directly combining neighboured line segments (Fig. 9). An appropriate choice of the grouping conditions (e.g. minimum line length, accepted gradient deviation) produces good detection results as shown below. A detailed explanation of the multi-level grouping algorithm is given in [16].

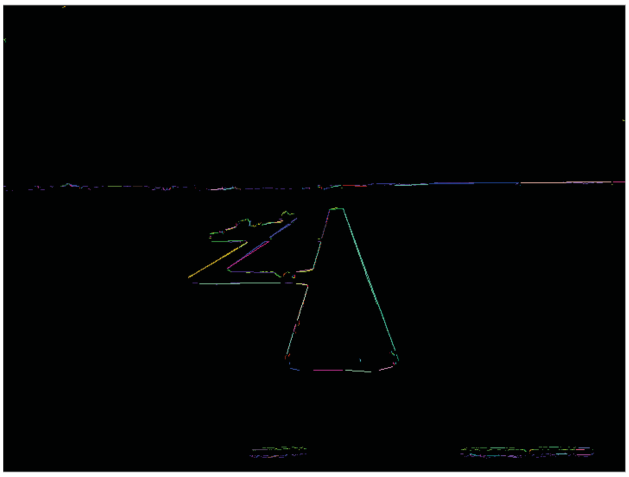


Fig. 9: Lines grouped from primary detections

The finally resulting closed structures are now characterisable with robust features. A conscientious choice of unique, application-beneficial features with good separation quality is important and has an obvious impact on classifier performance.

The presented application example has been implemented in software and evaluated with both synthetic and real image data. Running the runway recognition system on a standard workstation achieves no real-time quality. But the image processing using task parallelisation demonstrates a robust elimination of false positives in the set of primary detections. This first software implementation proves the feasibility of the proposed concept for task parallelisation for hardware-accelerated processing of high-resolution image data.

6. CONCLUSION AND FUTURE WORK

High-resolution image data exploitation is challenging regarding the real-time aspect. The proposed concept in this paper faces these challenges by application of data and task parallelisation strategies for FPGA-based hardware acceleration platforms. Therefore multiple, parallel low-level image processing modules are used and the intermediate results are fused. The deployed low-level detectors comprise input independent algorithms without iterations or recursions.

The fusion of all low-level results is realised by the Fuzzy Logic Rating layer. The FLR reliably removes false positive detections of the primary detector. The weak primary detections are strengthened by several secondary low-level results. This minimizes the computational effort of the post-processing step by reducing the data amount to be processed.

The modularity of task parallelisation allows flexible, toolbox-like adaption of the image processing system. It flexibly supports adaption of the processing architecture based on well-modelled detection goal and application knowledge (static configuration). Furthermore, task parallelisation offers system modification during operation (dynamic reconfiguration). Through the independence of the multiple low-level modules they can be replaced if the application changes or re-parameterised to adapt a detector to modified detection conditions. Also sensitivity of a low-level processing module can be enhanced if it generates no valid results. More sensitive re-analysing of image areas with probable detection candidates could achieve additional valid results (back-loop concept [12]).

Developers have to consider application-specific environmental parameters to choose a reliable primary detector. If no back-loop concept is used, the sensitivity of the primary detector shall be very high to avoid loss of true positives.

Due to the strong sequential character of software-based image processing of a huge amount of high-resolution image data, a significant speedup will only be reached by a hardware implementation. Software does not support intensive data and task parallelisation as essential for this concept. Therefore, we are currently porting various low-level image processing modules to a FPGA hardware platform. We focus efficient spreading and parallelisation of the image processing chain using the characteristic parallelisation strength of FPGAs. Goal is a complete hardware implementation of the entire image processing chain starting with data acquisition, exploitation including final result dissemination.

7. ACKNOWLEDGEMENT

The concept presented in this paper has been developed within the Open Innovation Project, a cooperation of EADS Cassidian, research institutes and universities. The research program sponsored by EADS Cassidian intends to develop and support maturation of next-generation technologies for unmanned air systems.

8. REFERENCES

[1] Thrun, Burgard and Fox, Probabilistic Robotics - Intelligent Robotics and Autonomous Agents, USA, Massachusetts, Cambridge: The MIT Press, 2005.

[2] Gribbon, Johnston and Bailey, “A Real-time FPGA Implementation of a Lens Distortion Correction Algorithm with Bilinear Interpolation,” in *Proceedings of the Image and Vision Computing New Zealand Conference 2003, Massey University*, Palmerston North, New Zealand, 2003.

[3] Gribbon, Johnston and Bailey, “Implementing Image Processing Algorithms on FPGAs,” in *Proceedings of the Eleventh Electronics New Zealand Conference, ENZCon’04*, New Zealand, Palmerston North, 2004.

[4] Saldanha, Hartmann and Bobda, “Scratch Detector – A FPGA Based System for Scratch Detection in Industrial Picture Development,” Trento, 2010.

[5] Papadonikolakis and Bouganis, “A Novel FPGA-based SVM Classifier,” London, UK, 2010.

[6] Meng, Appiah, Hunter and Dickinson, “FPGA Implementation of Naive Bayes Classifier for Visual Object Recognition,” London, 2011.

[7] M. Cho, Oberg and Kastner, “FPGA-Based Face Detection System Using Haar Classifiers,” San Diego, United States, 2009.

[8] Pacholik, Muller, Fengler, Machleidt and Franke, “GPU vs FPGA: Example Application on White Light Interferometry,” in *Proceedings of International Conference on Reconfigurable Computing and FPGAs*, Mexico, Cancun, 2011.

[9] Rosenband and Rosenband, “A design case study: CPU vs. GPGPU vs. FPGA,” in *Proceedings of International Conference on Formal Methods and Models for Co-Design*, USA, Massachusetts, Cambridge, 2009.

[10] Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of*

Computer Vision 60/2, pp. 91-110, 2004.

[11] Scheunert, Fuzzy-Mengen-Verknüpfung und Fuzzy-Arithmetik zur Sensor-Daten-Fusion, Duesseldorf: VDI Verlag GmbH, 2002.

[12] Lindner, Blokzyl, Wanielik and Scheunert, “Applying Multi Level Processing for Robust Geometric Lane Feature Extraction,” in *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, University of Utah, Salt Lake City, UT, USA, Sept. 5-7, 2010, 2010.