IEEE 2nd Ukraine Conference on Electrical and Computer Engineering , Lviv, Ukraine, July 2-6, 2019. P. 1175-1178

System of Feature Extraction for Video Pattern Recognition on FPGA

Anatolij Sergiyenko
Computer Engineering Department
Igor Sikorsky Kyiv Polytechnic Institute
Kiev, Ukraine
aser@comsys.kpi.ua

Maria Orlova
Application Specific System Department
Igor Sikorsky Kyiv Polytechnic Institute
Kiev, Ukraine
orlova@ua.fm

Pavlo Serhiienko
Application Specific System Department
Igor Sikorsky Kyiv Polytechnic Institute
Kiev, Ukraine
paulsrgnk002@gmail.com

Oleksii Molchanov
Application Specific System Department
Igor Sikorsky Kyiv Polytechnic Institute
Kiev, Ukraine
oleksii.molchanov@gmail.com

***Abstract* — A system is proposed for image recognition in a video stream, such as inscriptions, road signs. It is able to recognize the patterns in the complex lighting conditions due to processing the high dynamic range (HDR) signals. The image feature extraction is based on the method, which originated from the Retinex method but is adapted to the HDR images. Due to the new method, the bilateral filter is exchanged to the 2D edge-preserving adaptive filter. The filter output gives information for the feature extraction detectors. The experimental HDR video camera with the feature extraction is built around the Lattice HDR-60 board.**

Keywords—FPGA, HDR, feature extraction, bilateral filter

I. INTRODUCTION

Pattern recognition is a part of machine learning and artificial intelligence. Modern methods for the pattern recognition of the images are based on the artificial neural networks and techniques based on the feature point detection. The convolutional neural nets (CNN's) have become widespread for the recognition of graphic images. But they require large volumes of floating point calculations, which also entails high power consumption of the image recognition systems [1].

The image recognition with the feature point detection is based on the feature descriptor extraction from the image, splitting the descriptor space into clusters and finding the pattern at a distance to the center of the cluster. Unlike CNN method, the descriptor computing needs not performing the computing-intensive calculations [2].

The scale-invariant feature transform (SIFT) method is considered to be the famous, but also the most reliable one because it provides recognition in the poor lighting conditions, and in zoomed-out images. However, it is a rather complex method. It implements four steps: detection of feature points, point localization, point direction definition, and descriptor formation. The descriptor vector codes the feature point description, orientation, and scale parameters that are invariant to the image transformations. [3].

Many pattern recognition methods inherit the SIFT method features. Among them DSIFT method has much less complexity [4], SURF method utilizes the integer calculations instead of floating point ones, as well as the wavelet functions and Haar transform [5], the descriptors of the HOG method represent the image gradient sets in the feature points [6]. The BRIEF method is based on comparing the stencils of pixels, which are selected randomly around the selected feature point, but it is not immune to the image rotation [7]. This disadvantage was removed in the OPB method [8]. In the BRISK method more sophisticated binary descriptor is used than in the SIFT method [9], and in the FREAK method, it was improved, taking into account the properties of the eye retina [10]. The last methods decrease the computing time in two degrees of magnitude, providing recognition improvement.

The mentioned methods have the large complexity of the feature point searching and its descriptor deriving, which makes complex its FPGA implementation. In this presentation, a new method of the feature point extraction is proposed, which utilizes the side effect of the high dynamic range (HDR) image compression algorithm.

II. RETINEX HDR IMAGE COMPUTING

The HDR image has the pixels, which are twenty and more bit wide. Note, that the usual pattern recognition systems deal with eight-bit images. HDR systems are effectively used in the outdoor cameras because they perceive the information in the difficult exposition conditions. But in such systems, the problem of the signal compression without the losses of the information both in illuminated and in darkened areas needs to be solved.

The problem of the HDR image computing is effectively solved on the base of the Retinex model theory which investigates the scene lighting. This model insists that the pixel with the coordinates (x, y) has brightness

$$I(x, y) = L(x, y) \cdot R(x, y),$$

where $L(x, y)$ is the illumination, $R(x,y)$ is the reflected brightness of the object [11]. Due to the Retinex method, the value $I(x, y)$ is decomposed to the factor $L(x, y)$, and factor $R(x, y)$. Then, the factor $L(x,y)$ is processed with the dynamic range compression, and the contrast is improved in the component $R(x,y)$. Then these image components after their multiplication give the resulting image $I'(x,y)$. The factor $L(x,y)$, which determines the scene illumination, is derived by the function $F(I)$, so that

$$L(x,y) = F(I); \qquad R(x,y) = I(x,y) / F(I);$$
$$L'(x,y) = \Gamma(L(x,y)); \qquad R'(x,y) = \beta(R(x,y)); \qquad (1)$$
$$I'(x,y) = L'(x,y)\cdot R'(x,y).$$

Here, the compression function $\Gamma(y)$ behaves as a logarithmical one, and the contrast improvement function $\beta(y)$ amplifies the signal near the level of 1.0. The illumination function $F(I)$ is approximated by a 2D low-pass filter (LPF) in the simple case. But to prevent the artifacts, appearing in the resulting image, the function $F(I)$ is one which is preserving the edges. The bilateral filter is often used as the function $F(I)$ [12]. The bilateral filter makes slight changes around the feature points and smoothes the image around them. As a result, the compressed image preserves the sharp transition of brightness.

The bilateral filter needs a large volume of computations. When processing the HDR image, the computations increase dramatically. The proposed adaptive filter preserves the edges as well as the bilateral filter does, but has much less hardware volume. Due to this, HDR image processing is much simpler and has higher throughput. As a side effect, its intermediate results are valuable. They are effectively used for pattern recognition.

III. Adaptive Filter Structure

The mentioned above adaptive filter contains an image analyzer and two-dimensional LPF which is regulated dynamically. The idea of the Harris-Laplace detector [13] is used in the image analyzer. Its output signal represents the eigenvector of the autocorrelation matrix of the image in the neighborhood of the pixel under processing. Five detectors W_*, $W_|$, W_-, $W_/$, $W_\backslash$, are constructed. They are sensitive to the blobs, vertical, horizontal, or inclined edges in the image. A separate LPF W_{LPF} serves to estimate the local brightness in the analyzed point. For example, the kernel of the filter for the horizontal line recognition has the following matrix:

$$W_- = \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{pmatrix}. \qquad (2)$$

This kernel is implemented in the detector structure illustrated by Fig.1.

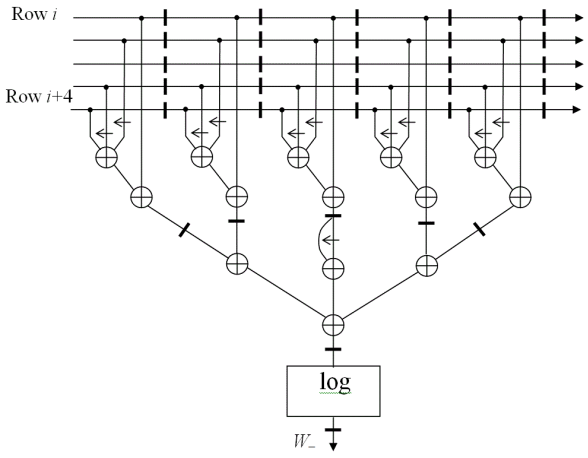

Fig. 1. One of the feature detectors

Here, Row*i* is the dataflow of the *i*-th frame row, a black bar means a single register delay, an arrow across the line means the datum shift left, a circle with a plus is an adder. The output signal of such a detector has the logarithmic scale. Such a structure provides small hardware costs, high throughput, and high dynamic range. The decision unit of the analyzer finds the maximum signal of the detectors

$$D(x,y) = \max(W_|, W_-, W_/, W_\backslash, W_*) - W_{LPF},$$

The logarithmical local brightness W_{LPF} is subtracted from this maximum providing the normalized HDR signal. The output signal is accompanying by the detector number. The results of the image analyzer for the images of the characters ⊙ and ❍ are illustrated by Fig. 2. The color of a pixel in these images codes the detector number.

The dynamically regulated LPF operation consists in the following. A set of filter kernels are stored in a table. These kernels are distinguished depending on the local image type, i.e., if it is inclined, horizontal, vertical, edge, and on its strength. This table contains 40 different kernels. All of them are pre-normalized. As a result, the filter does not need the dynamical normalization, which is very computationally intensive and is obligatory in the bilateral filters.

The image analyzer signal forms the table address, i.e., it selects the proper kernel in it for the image pixels. For example, when the edge is detected, then the filter kernel is selected, which is sensitive to this edge. When a uniform field is present, then it is filtered by the Gaussian filter kernel providing high blur. So, the adaptive filter provides the sharpness of the features and smoothes the non-informative fields. The detector output $D(x,y)$ gives valuable information about the image feature points, which needs the additional processing described in the next section.

IV. Feature Point Extraction

Two additional computational steps are needed to search for the feature points in the image. First of them is the noise filtering. The linear digital filtering methods are not fit for the image $D(x,y)$ because of its logarithmical scale. The maximum homogeneity neighbor (MHN) filter can do with such a signal [14]. The MHN filter essence consists in the following. It contains a set of image stencils. During the MHN filtering, the resulting pixel with coordinates (x,y) is equal to the average value of pixels, which belong to some stencil if they have the homogeneous brightness. It is worth to mention that the pixels $D(x,y)$ have the small bit width (six bits of brightness and three bits of color code). Therefore, the filtering process is simplified during the feature point extraction. The stencil is found, which covers the pixels of the same color as the pixel $D(x,y)$ is. Then, these pixels are averaged. If the result is less than the range then the pixel is assigned to the background color. This filter is effectively implemented using the look-up tables of FPGAs.

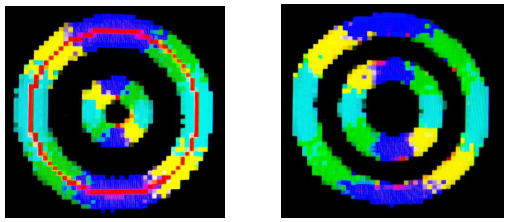

Fig. 2. Results of the image analyzer

The feature points are intersections of the lines, as well as corners, blobs, etc. They are searched in the second step. The searching for the feature points can be implemented by the modified MHN method.

The method modification consists in that, that the spatial stencils are used. Such a stencil is adapted to the feature under consideration. The feature point of the corner with the given orientation and angle has the stencil which is formed by the pixels with the respective colors but with the constant magnitude.

If the cyan pixel is accompanied by yellow pixels below and above, then it is contained in a horizontal line. If the color of this pixel and pixels in its neighborhood is cyan, then it belongs to a blob, etc. The found feature point is coded by its coordinates (x,y), feature type, magnitude, spatial direction, and parameter like the corner angle.

This process is improved by the distribution diagram, which is built according to the Hough method [15]. The examples of the feature points and respective distribution diagrams are shown in Fig.3.

So, it is no need to perform an analysis of whether the considered feature point belongs to the edge of the image or not, as in the original SIFT algorithm. At the same time, much less computation is performed. Comparing to the Harris-Laplace detector [13] this feature point detection is much simpler when it is performed in FPGA.

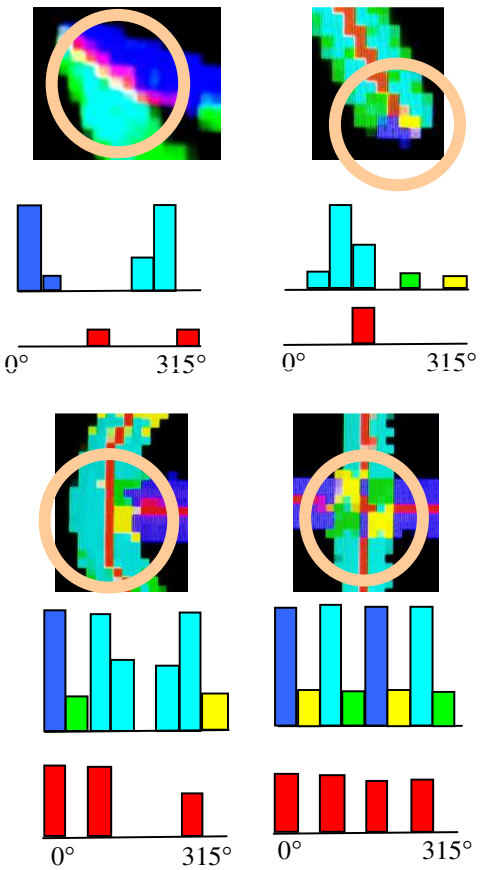

Fig. 3. Feature points and respective distribution diagrams

The derived logarithmic image $D(x,y)$ is decimated for getting the scalable feature point extraction. Such a decimation is performed using the MHN method except that it is performed to each even pixel. By this process, the feature point coordinates coincide with each other in the images with different scales.

V. Experimental Results

An experimental system is designed for proving the feature extraction for the video pattern recognition. Besides, it executes the HDR compression and functions of the smart video camera. It is designed on the base of the HDR-60 FPGA board of the Lattice company. The HDR video stream with the frames 720 by 1280 pixels are processed at a rate of 60 fps. Their dynamic range achieves 110 dB and higher.

A set of designed IP cores like debayerization unit, median filter, brightness corrector, color converter, histogram former are configured in FPGA. The IP core of the feature point extraction and HDR compression plays the main role. All of them operate in real-time.

The feature detectors have the kernels of the form (2). They are designed as the multiplier-less adder trees, which are pipelined. The structure of such a detector is shown in Fig.1. This structure implements both real-time computations of the HDR signal and small hardware costs.

The pipelined datapaths of the IP cores mentioned above are developed according to the spatial synchronous data flow graph (SDF) method. This method consists in representing the signal processing algorithm as SDF. This SDF is placed in the three-dimensional space using the special theorems. Then, this SDF is described in VHDL language using the formal rule set [20]. The method is distinguished in the design of pipelined datapaths providing the optimum ratio of the throughput to the hardware volume.

The project of the system, configured in the Lattice FPGA, has the maximum pixel frequency up to 135 MHz. This frequency for FPGAs of other series can be much higher. The hardware complexity of the feature point extractor for FPGAs of different types are shown in Table 1. The figures in the table show that the hardware volume takes a small part of the FPGA resources. It assures us that development of a system for the simple pattern recognition in a single FPGA is possible.

TABLE I. Hardware costs of the feature point extractor

FPGA hardware type	**Lattice ECP3**	**Xilinx Artix-7**	**Intel Cyclone V**
CLBs or ALMs	1905	840	1439
Registers	2273	2264	2284
LUTs or ALUTs	3090	2554	2336

Comparing to the similar systems [16,17], the proposed one provides sophisticated computations in real time using the FPGA chip of medium volume.

In [18], an example of the video camera is described, in which the described system is built. This camera use is

shown when it can reliably select the blood cells in the biochemical analysis image showing their features.

VI. Future work

At present, the artificial intelligence engine is under development, which is implemented in FPGA of medium or large size. The input data is an HDR video camera data flow. The pattern recognition process has two distinguished stages. On the first stage, the image is transformed into a pyramid of feature frames. For this process, an adaptive filter is used as in Fig. 1, which compresses the pixel dynamic range preserving the edges in it. The image adaptive filter uses the Harris-Laplace detector method. But for the noise filtering, and for the frame pyramid forming the MHN filter is used.

The feature points are searched in the second step. They are searched using the MHN method too. Then, the feature descriptors are formed as like as in the SIFT method.

At the period of learning, the computed feature descriptors are stored to the database, and at the working period, the found feature descriptors are compared to ones in the database.

The structure of the artificial intelligence engine configured in FPGA consists of a set of application-specific processors. Each of them contains the specific pipelined datapath and the controller. The datapath performs the high-speed processing of the video stream. The controller provides the processor task synchronization, data loading from the outer memory, loading the adjusting coefficients. The processor structure provides the operation in the data flow mode. Its operation looks like the TensorFlow system operation proposed by Google.

The controller has the stack processor architecture, which is similar to one described in [19]. Due to this, it has minimized both hardware and software volume and can be frequently programmed using the Forth language. The Forth compiler for it is under design.

To design the pipelined datapath in a short time, a framework is designed which helps to compile the data path algorithm representation into the FPGA hardware described in VHDL. The algorithm can be represented in the TensorFlow language as well.

VII. Conclusion

The feature extraction method is proposed which deals with the HDR images. The method consists of detecting the feature properties and searching for the feature points. The feature properties are represented in the logarithmical scale. The feature points are coded by its coordinates, feature type, magnitude, spatial direction, and other parameters. They are searched by the modified MHN method using the distribution diagram.

The feature detectors form the feature point extractor which effectively selects the pixels which belong to the lines, edges or blobs. Its implementation in FPGA has both high throughput and small hardware volume. It is built in an experimental example of the HDR video camera, which provides the sharp images both in darkened and in illuminated areas of the image. The design of the artificial intelligence engine is planned, which is implemented in the FPGA of medium or large size.

References

[1] V. K. David, S. Rajasekaran, “Pattern Recognition Using Neural and Functional Networks”, Springer, 196 p., 2009.

[2] F. Y. Shin, “Image Processing and Pattern Recognition. Fundamentals and Techniques”, Wiley, 537 p., 2010.

[3] D.G. Lowe, “Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision, V. 60, N2, pp. 91-110. 2004.

[4] A. Bosch, A. Zisserman, X. Munoz, “Scene classification via pLSA”, Proc. 9th European Conference on Computer Vision (ECCV'06), Springer Lecture Notes in Computer Science N. 3954, 2006, pp. 517–530.

[5] H. Bay, T. Tuytelaars, L. van Gool, Luc (). SURF: Speeded up robust features. Proc. 9th European Conference on Computer Vision (ECCV'06), Springer Lecture Notes in Computer Science, N.3951, 2006, pp. 404-417.

[6] N. Dalal, B. Triggs, “Histograms of oriented gradients for human detection”, Proc. Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, 2005, pp. 886-893.

[7] M. Calonder, V. Lepetit, C. Strecha, P. Fua, “Brief: Binary robust independent elementary features”, Computer Vision–ECCV, 2010. Springer, 2010, pp. 778-792.

[8] E. Rublee, V. Rabaud, K. Konolige, G. Bradski "ORB: an efficient alternative to SIFT or SURF", Proceedings of the International Conference on Computer Vision, ICCV’11, 2011, pp. 2564-2571.

[9] S. Leutenegger, M. Chli, R. Y. Siegwart. “BRISK: Binary Robust Invariant Scalable Keypoints”, Proceedings of the International Conference on Computer Vision, ICCV’11, 2011, pp.2548-2562.

[10] A. Alahi, R. Ortiz, P. Vandergheynst, “FREAK: Fast Retina Keypoint”, Proceedings of the International Conference on Computer Vision, ICCV’12, 2012, pp. 510-517.

[11] J. J. McCann, E. H. Land, “Lightness and retinex theory”, Journal of the Optical Society of America, V. 61, N. 1, 1971, pp. 1 – 11.

[12] S. Paris, P. Kornprobst, J. Tumblin, F. Durand, “Bilateral filtering: theory and applications”, Foundations and Trends in Computer Graphics and Vision, V. 4, N. 1, 2008, pp. 1–73.

[13] M. Hassaballah, A. A. Abdelmgeid, H. A. Alshazly, “Image features detection, description and matching”, in: Image Feature Detectors and Descriptors. Foundations and Applications. A. I. Awad, M. Hassaballah, Eds. Springer, 2016, pp. 11 – 46.

[14] M. Nagao, and T. Matsuyama, “Edge preserving smoothing”, Computer Graphics and Image Processing, vol. 9, No. 4, 1979, pp. 394–407.

[15] R. O. Duda, P. E. Hart, “Use of the Hough Transformation to Detect Lines and Curves in Pictures”, Comm. ACM, V. 15, No. 1. 1972. pp. 11-15.

[16] T. Ai, M. A. Ali, G. Steffan, K. Ovtcharov, S. Zulfiqar, and S. Mann, “Real-time HDR video imaging on FPGA with compressed comparametric lookup tables”, 27th IEEE Canadian Conference on Electrical and Computer Engineering, CCECCE’14, 4–7 May, 2014.

[17] P.-J. Lapray, B. Heyrman, D. Ginhac, “HDR-ARtiSt: An adaptive real-time sSmart camera for high dynamic range imaging”, Journal of Real-Time Image Processing, Springer Verlag, 2014, pp.1-16.

[18] A. Sergiyenko, P. Serhiienko, Ju. Zorin, “High Dynamic Range Video Camera with Elements of the Pattern Recognition“, IEEE 38th International Conference on Electronics and Nanotechnology, ELNANO’18, Kyiv, April 2018. pp. 435-438.

[19] O. Molchanov, A. Sergiyenko, M. Orlova, "Software/Hardware Co-design of the Microprocessor for the Serial Port Communications", in Advances in Computer Science for Engineering and Education II. Springer. Advances in Intelligent Systems and Computing, V. 938, 2019, pp. 238-246.

[20] A. Sergiyenko, A. Serhiyenko, A. Simonenko, “A method for synchronous dataflow retiming”, 1-st IEEE Ukraine Conference on Electrical and Computer Engineering (UKRCON), pp. 1015-1018, 2017.