Designing LBP-descriptor for reconfigurable computing environments

D V Shashev*, A A Taganov, M Mondal and M V Okunsky
National Research Tomsk State University, 36 Lenin ave., 634050, Tomsk, Russia

*E-mail: dshashev@mail.tsu.ru

Abstract. The article outlines a new approach to constructing Local Binary Patterns (LBP-descriptors) for implementation on computers with parallel-pipeline architectures. This approach can be effectively applied to pedestrian detection. The new algorithm is optimized for the original model of the reconfigurable computing environment, which allows it to be implemented in the form of combinational logic with high speed. A mathematical model for the new LBP-descriptor is also presented.

1. Introduction

Computer Vision is one of the most popular areas of science and technology, the development of which directly affects the advancements in robotics and artificially intelligent systems. A huge number of tasks are solved by creating systems that can process video to understand the relative surroundings.

One of these tasks is pedestrian detection by onboard computers of unmanned vehicles. Although research in this field has been done for at least 20 years, there is no universal solution, addressing the limitations due to computing resources, real-time systems, high-resolution video streams, as well as the security and health risks.

An alternative approach to the pedestrian detection, in which existing algorithmic solutions are adapted to the concept of a reconfigurable computing environment is shown in [1-3], which gives them the properties of hardware implementation on computing structures with parallel-pipeline architecture (especially on Field-Programmable Gate Array, FPGA) and a significant advantage in performance. In turn, these properties of algorithms allow us to overcome the limitations of their use in onboard computers of unmanned vehicles.

The research aim is to develop new image descriptors adapted for high-performance computing systems with parallel-pipeline architecture, based on the model of parallel processing [1-3].

The article considers the construction of adapted LBP-descriptor, as one of the most frequently used algorithms for the pedestrian detection.

2. Local Binary Patterns

Pedestrian detection is usually a two-step process which includes feature extraction and classification. Multiple methods have been used for detecting pedestrians. One of the possible classifiers includes [4]:

- gradient-based features;
- shape-based features;
This paper focuses on the descriptor of textual features, namely LBP-descriptors, since textual features are the basic features of any image and are used everywhere in image recognition algorithms.

The LBP-descriptor is a binary descriptor derived from texture analysis. Despite their high memory usage, they have become a modern descriptor for detecting pedestrians due to their fast computation and reliable performance \cite{5}. This type of descriptor is well established in the field of pedestrian detection due to its mathematical simplicity and high performance, but the disadvantage is a large amount of memory usage for the resulting large feature vector, which can be compensated either by increasing computing resources or using alternative computing architectures.

The first step in the LBP-descriptor is to convert the image to grayscale. After that, each pixel of the image is assigned an LBP-code using equation (1), which is calculated using the formulas below, and the image is transformed into an LBP-map.

\[
LBP_{p,R}(x,y) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p,
\]

where \( P \) is the number of considered neighbourhood points; \( R \) is the radius of the surrounding area; \( x, y \) are the coordinates of the central pixel; \( p \) is the number of the considered pixel; \( g_c \) is the brightness value of the central pixel; \( g_p \) is brightness of the considered pixel; \( s(g_p - g_c) \) is calculated using equation (2).

\[
s(g_p - g_c) = \begin{cases} 
1, & \text{if } (g_p - g_c) \geq 1; \\
0, & \text{if } (g_p - g_c) < 0. 
\end{cases}
\]

To calculate the special code of the image pixel, 8 brightness values of neighbouring pixels are used. The brightness value of the pixel in question is the threshold against which the brightness values of neighbouring pixels are compared. If the neighbouring pixel has a lower brightness value, it is assigned 0. If greater than or equal to – 1, the resulting eight-bit number characterizes the neighbourhood of the pixel in question.

Then the resulting LBP-map is divided into an arbitrary number of blocks and for each of them, a histogram is formed in accordance with the following rules:

- the number of histogram bins is 256 (from 0 to 255 in increments of 1);
- the value of each bin is equal to the sum of pixels whose LBP-code values are equal to the bin number.

The histogram of the entire image represents the desired feature vector as a result and is formed by concatenating the histograms of each block. Later, this feature vector will be used to build a classifier.

The detailed principles of LBP-descriptor can be viewed in \cite{6}.

The most important property of the LBP-descriptor in the real-time application is its resistance to monotonous changes in grayscale, for example, by changes in lighting. Another important feature is its computational simplicity, which allows you to analyze images in real-time.

Returning to the arguments about the construction of the algorithm for pedestrian detection, it is worth noting that for the effective operation of the algorithm, several descriptors are most often used to form a common feature vector.

3. Reconfigurable computing environments (RCE)
RCE is a discrete mathematical model of a high-performance computing system consisting of identical and equally connected elementary universal elements (elementary calculators, ECs), programmatically tuned to perform any function from a complete set of logical functions, memory and any connection with its neighbours [1-3]. ECs works only based on logical functions AND-OR-NOT. According to the computer model proposed by russian scientists [3], based on which the RCE is built, the ECs in RCE are interconnected as shown in Figure 1. Each EC in RCE is bidirectionally connected to neighbouring ECs.

Figure 1. The structure of the interconnections of ECs in RCE: \(y_1, y_2, ..., y_8\) and \(f_1^M, f_2^M, ..., f_8^M\) are respectively information inputs and outputs between ECs; \(A_{ij}, B_{ij}, ..., H_{ij}\) are the name of the links between ECs; \(x_{ij}\) is the main input, which receives the pixel value of the original image; \(f_{ij}\) is the main output from which the value of the corresponding pixel of the resulting image is taken; \(z_{ij}\) is the tuning input to which the EC tuning code is input.

The basis for building the RCE are the following principles:

- homogeneity (due to the fact that all ECs are identical and are the same type connected);
- short-range action (due to the fact that all ECs are connected only to the nearest ECs, signal transmission between remote ECs is carried out via intermediate ECs);
- universality (due to the fact that each EC realizes a set of logical functions);
- software configuration (due to the fact that each EC can be configured to perform one function using external setting signals and continue to save the tuning state until the next tuning signal arrives).

Due to this, RCEs have the following unique properties:

- high speed (due to parallel computing);
- high reliability (due to interchangeability of ECs);
- high manufacturability (due to the uniformity of ECs and connections between them);
multifunctionality and adaptability (due to the possibility of changing the architecture for a specific task);  

- independence to element basis, i.e., independence from the manufacturing technology of computing environment.

4. Problem statement

The previous sections show that the LBP-descriptor works in spatial processing of the source image, and each pixel can be processed in the same way and parallel. From this, we can conclude that the fastest way to implement the LBP-descriptor is hardware, where each pixel of the image is processed in parallel.

The approach proposed by the authors allows us to adapt the algorithm of the LBP-descriptor to the model of an RCE, where the descriptor itself will be executed hardware. To do this, one needs to upgrade the LBP-descriptor as follows:

- each pixel of the source image must be processed by one EC of RCE;
- all calculations must be performed using Boolean algebra.

Fulfilling the first requirement will allow you to achieve maximum parallelism in the LBP-descriptor, and the second the ability to implement the developed algorithms on parallel-pipeline computing architectures. Moreover, the work of the descriptor can only be described by combinational logic, which in turn means almost instantaneous execution of the work of an EC in 1 clock cycle.

5. LBP-descriptor model for RCE

Given the features of building the RCE model, as well as the algorithmic features of the LBP-descriptor presented in sections 2-3, it is necessary to make some changes to the characteristics of the input and output data of the EC (Figure 1):

- all inputs $x_{ij}, y_1, y_2, ..., y_8$ are converted in $X$ and $Y_j$ are the 8-bit binary numbers,
- all outputs $f_{ij}, f_1^M, f_2^M, ..., f_8^M$ are converted in $F$ and $F_j$ are the 8-bit binary numbers,
- inputs $z_{ij}$ are converted in $Z$ are the 4-bit binary numbers.

Here, $X$ is the brightness of the pixel that the LBP-descriptor is applied to, $Y_j$ are the brightness of neighbouring pixels, $F$ is the generated LBP-code for the pixel, $F_j$ are the equal to $X$ and are passed to neighbouring EC for processing, $Z$ is the code for setting up an EC.

In order for the LBP-descriptor to work properly in RCE, an EC must perform the following nine functions, as well as be reconfigured to perform them using a programmable signal $Z = z_1 z_2 z_3 z_4$:

- processing the upper-left corner pixel of the source image and transmitting the brightness value to neighbouring elementary calculators, setting code $Z = 0000$;
- processing the upper-right corner pixel of the source image and transmitting the brightness value to neighbouring elementary calculators, setting code $Z = 0010$;
- processing the lower-right corner pixel of the source image and transmitting the brightness value to neighbouring elementary calculators, setting code $Z = 0111$;
- processing the lower-left corner pixel of the source image and transmitting the brightness value to neighbouring elementary calculators, setting code $Z = 0101$;
- processing the upper horizontal row of pixels in the source image and transmitting the brightness value to neighbouring elementary calculators, setting code $Z = 0001$;
- processing the right vertical row of pixels of the source image and transmitting the brightness value to neighbouring elementary calculators, setting code $Z = 0100$;
• processing the lower horizontal row of pixels in the source image and transmitting the brightness value to neighbouring elementary calculators, setting code \( Z = 0110 \);
• processing the left vertical row of pixels of the source image and transmitting the brightness value to neighbouring elementary calculators, setting code \( Z = 0011 \);
• processing all other pixels in the source image \( Z = 1000 \).

The first eight described functions are necessary to account for the operation of the LBP-descriptor with the edge pixels of the source image, which are not taken into account when compiling the LBP-map but are necessary for transmitting brightness values to neighbouring rows. The last function of the EC calculates LBP-codes for all image pixels except for the edge ones.

For EC operation with the basis of logical functions "AND-OR-NOT", the fulfilment of the above functional consists of the following system of equations (3).

\[
\begin{align*}
F_1 &= X (z_j z_2 \bar{z}_4 \lor z_j \bar{z}_2 \bar{z}_4 \lor z_j \bar{z}_4 \bar{z}_2), \\
F_2 &= X (z_j \bar{z}_4 \lor z_j z_2 \bar{z}_4 \lor z_j \bar{z}_2 \bar{z}_4), \\
F_3 &= X (z_j \bar{z}_2 z_4 \lor z_j \bar{z}_2 \bar{z}_4 \lor z_j \bar{z}_4 \bar{z}_2 \lor z_j z_4 \bar{z}_2), \\
F_4 &= X (z_j \bar{z}_2 \bar{z}_4 \lor z_j \bar{z}_2 \bar{z}_4 \lor z_j \bar{z}_4 \bar{z}_2 \lor z_j z_4 \bar{z}_2), \\
F_5 &= X (z_j \bar{z}_2 \bar{z}_4 \lor z_j \bar{z}_2 \bar{z}_4 \lor z_j \bar{z}_4 \bar{z}_2 \lor z_j z_4 \bar{z}_2), \\
F_6 &= X (\bar{z}_j \bar{z}_2 \bar{z}_4 \lor \bar{z}_j \bar{z}_2 \bar{z}_4 \lor \bar{z}_j \bar{z}_4 \bar{z}_2 \lor \bar{z}_j \bar{z}_4 \bar{z}_2), \\
F_7 &= X (\bar{z}_j \bar{z}_2 \bar{z}_4 \lor \bar{z}_j \bar{z}_2 \bar{z}_4 \lor \bar{z}_j \bar{z}_4 \bar{z}_2 \lor \bar{z}_j \bar{z}_4 \bar{z}_2), \\
F_8 &= X (\bar{z}_j \bar{z}_2 \bar{z}_4 \lor \bar{z}_j \bar{z}_2 \bar{z}_4 \lor \bar{z}_j \bar{z}_4 \bar{z}_2 \lor \bar{z}_j \bar{z}_4 \bar{z}_2), \\
F' &= M(Y_j, X) \cdot z_j \bar{z}_2 \bar{z}_4, \quad \text{where} \ j = \overline{1,0}.
\end{align*}
\]

Here \( M(A, B) \) is the auxiliary function that returns a boolean 1 if an 8-bit binary number \( A \) less than an 8-bit binary number \( B \), otherwise, a boolean 0 is returned. When \( n = 8 \), equation (4) is used.

\[
M(A, B) = a^8 b^8 \lor (a^{n-1} b^{n-1} \cdot (a^n b^n \lor a^8 b^n)) \lor (a^{n-2} b^{n-2} \cdot (a^n b^n \lor a^8 b^n)) \cdot (a^{n-1} b^{n-1} \lor a^{n-2} b^{n-2}) \cdot (a^8 b^8 \lor a^2 b^2) \cdot (a^2 b^2 \lor a^8 b^2).
\]

According to the principles of building the RCE model, the elementary computer, shown in Figure 1, as well as the LBP-descriptor mathematical model based on a system of logical functions, it is possible to create a simulation model of RCE in Matlab Simulink (ver. R2020a) in the form of combinational logic.

The RCE simulation model is a matrix environment whose dimensions coincide with the dimensions of the processed image because each EC processes one pixel of the image in parallel. Due to the bulkiness of the model, Figure 2 shows its general appearance with the dimension of \( 5 \times 5 \) EC.

For ease of modelling, connections between EC are organized using Simulink blocks "Go to". The combinational logic in each EC model is also quite cumbersome to display, so only part of it is shown in Figure 3.
Figure 2. General view of the RCE simulation model with the dimension of $5 \times 5$ EC.

Figure 3. Part of the combinational logic of an EC.
The simulation was carried out to check the built mathematical model of the LBP-descriptor for the reliability of its operation. The comparison was carried out by analyzing the obtained LBP-maps of the constructed model and the classical implementation of the LBP-descriptor available in Matlab.

Figure 4 shows the progress of the classic LBP-descriptor, adapted to the model of a reconfigurable computing environment with identical visualized LBP-maps. This confirms the correctness of the new LBP-descriptor.

![Figure 4](image)

**Figure 4.** Comparison of the results of classical LBP and LBP for RCE.

A comparison of the speed of the Classic LBP and LBP for RCE is presented in table 1. Simulation of the Classic LBP was performed in Matlab R2020a using the Intel i9-9880H processor. Combinational logic of the LBP for RCE was simulated in Quartus Prime Lite Edition ver. 20.1 using the Intel Cyclone V FPGA SX.

Analysis of table 1 shows that the ability to execute LBP as a combinational logic on the FPGA increases performance by more than $10^6$ times, regardless of the image size. The size of the processed image will be limited only by the characteristics of the FPGA.
Table 1. A comparison of the speed of the Classic LBP and LBP for RCE

| Image size (pixels) | Classic LBP | LBP for RCE |
|---------------------|-------------|-------------|
| 640 × 480           | ~ 39 · 10^6 | ~ 6         |
| 1280 × 720          | ~ 100 · 10^6| ~ 6         |

6. Conclusion

This paper presents the advantages of constructing a new mathematical model of the LBP-descriptor for execution in a model of an RCE. This alternative approach allows you to create specialized software for computers with parallel-pipeline architectures. The algorithm of the classical LBP-descriptor was modified, and a new mathematical model based on boolean algebra logical functions was developed. The advantage of the new LBP-descriptor is the ability to represent the algorithm in the form of combinational logic, which can be easily implemented in an FPGA or Application-Specific Integrated Circuit (ASIC) with very high performance since combinational schemes are executed virtually instantly in such computing structures. Table 1 shows the improvement in speed by more than 10^6 times for new implementation of the LBP-descriptor. The development of this algorithm, as well as other algorithms in the field of computer vision systems, will be in demand for the use in mobile robotics and the construction of intelligent onboard systems for unmanned vehicles. Further research will be devoted to the implementation of the developed algorithm in the FPGA and the analysis of features and requirements for the computational structure.

Acknowledgements

The reported study was funded by RFBR, project number 19-29-06078.

References

[1] Shashev D and Shidlovskiy S 2015 Opt. Instr. and Data Processing 51 19
[2] Shidlovskiy S V 2016 MATEC Web of Conf. 79 01014
[3] Khoroshevsky V G 2004 Int. Conf. on Parallel Computing in Electrical Engineering (Dresden, Germany: IEEE)
[4] Bali S and Tyagi S S 2018 Int. J. of Adv. St. of Sci. Res. 3 100
[5] Trichet R and Bremond F 2018 IEEE Winter Conf. on Appl. of Comp. Vision (Lake Tahoe, NV, USA: IEEE) pp 1066-1074
[6] Karis M S, Razif N R A, Ali N M, Rosli M A, Aras M S M and Ghazaly M M 2016 IEEE 12th Int. Coll. On Signal Proc. & its Appl. (Malacca City, Malaysia: IEEE) pp 221–226