Modeling and prototyping of biometric systems using dataflow programming

N Minakova, I Petrov

Altai State University, 61 Lenin av., Barnaul, 656049, Russia
E-mail: PetrovIV90@gmail.com

Abstract. The development of biometric systems is one of the labor-intensive processes. Therefore, the creation and analysis of approaches and techniques is an urgent task at present. This article presents a technique of modeling and prototyping biometric systems based on dataflow programming. The technique includes three main stages: the development of functional blocks, the creation of a dataflow graph and the generation of a prototype. A specially developed software modeling environment that implements this technique is described. As an example of the use of this technique, an example of the implementation of the iris localization subsystem is demonstrated. A variant of modification of dataflow programming is suggested to solve the problem related to the undefined order of block activation. The main advantage of the presented technique is the ability to visually display and design the model of the biometric system, the rapid creation of a working prototype and the reuse of the previously developed functional blocks.

1. Introduction

We can see rapid growth and development of biometrics systems in recent years. Biometrics technologies got width spreading in various areas of human being. The last version of smartphones from the several producers has fingerprint readers and even iris scanners. Despite on rapid growth and width spreading there are many problems related to biometrics such as [1-3]:

- Possibility to fake biometrics features. Biometrics features can be spoofing (for example, rubber pads on the fingers). Many producers and developers of biometric devices react to this by adding additional technologies for detecting spoof. However, intruders find the new ways to counterfeits.
- Compromising. Unlike the password, biometric characteristic can’t change in case of compromising. Compromised biometric characteristic cannot used for identification person anymore.
- Bandwidth of biometric systems. The developers constantly strive to accelerate recognition using more optimal algorithms and more advanced scanners.
- Confidence. Biometric systems can prove a personality only with some probability. The basic characteristics of any biometrics systems are false acceptance and false rejection rates (FAR and FRR respectively).
- Convenience. Some biometrics systems demand high level of cooperation, which reduces the convenience of using this system.
There are no specific methodologies of biometrics systems development. Assessments of performance and reliability based on empirical data. The basic characteristics of biometrics systems such as FAR and FRR are calculated from results of testing by biometric samples. A public databases of biometrics samples are used for reproducibility purposes during the technological tests. Developers can preliminary reconfigure his recognition algorithms under the specific sampling during the tests on such databases [3]. So, early making of a working prototype helps to reveal many different problems and fix them.

The technique of modeling and prototyping of biometric systems based on dataflow programming is proposed in this work.

2. Dataflow programming

A dataflow program is a directed graph with the functions in the nodes which transform input data from the input arcs to output data which is transmitting then over the outputs arcs to other nodes [4-6]. Arcs connect output and input ports of blocks. Data performing occur as result of block activation after their input ports receipt the data. Important components of a biometric system are a feature extraction subsystem and a feature comparison subsystem. These subsystems represent a pattern recognition system. The internal structure of such systems is in good agreement with the structure of the data flow program. The data in this case are raw, intermediate and processed biometric samples, and nodes are processing functions, filters, classifiers, etc. This approach has many advantages in developing of systems oriented on data processing:

- Visualization. Dataflow graphs can easily visualize as dataflow diagrams.
- Modularity. Concept of nodes is in good agreement with modularity which is allowing reuse of the earlier developed components.
- Extension. The modules have simple and uniform interface which facilitates the expansion of the system.
- Parallelization. Ease to parallelize a program.

The commercial systems have much more complex structure. Beside the primary functionality, it is necessary to solve many other tasks including integrity protection of system, data encryption, storage the big data, integration with external systems, etc. Implementation of all of this with the dataflow programming is not expedient. However, dataflow programming allows significantly reduce developing time and reveal previously hidden problems and features when creating prototypes of biometric systems, or individual components of biometric systems that perform basic functions. Also, dataflow architecture leads to natural parallelization, which can has positively affect to system performance in multi-core architectures.

The modeling environment based on dataflow programming has been developed in frames of this work. Functional logic of individual blocks is placed in dynamic plugged modules. Special open interfaces for integration of various third-party algorithms in one system are provided.

Proposed technique of developing includes three stages:

1. Functional blocks developing;
2. Biometric system’s model creating;
3. Prototype generating.

3. Functional blocks developing

Header files are provided with the abstract Block class, which contains virtual functions that provide the basic functionality:

1. InternalPerform is a function which processing data from the inputs ports and transmitting results directly to the outputs ports.
2. Init is the function which is necessary to create ports which will be used in performing function.

To extend the functionality of the system it is necessary to inherit from the base class Block.

An example code with the implementation of the block’s functions for this system shown in figure 1.
4. Biometric system’s model creation
The second stage is the process of modeling. Using the functional blocks created in the first stage, a data flow graph is created with adding these blocks to the special scene and creating links between the input and output ports of the added blocks. A distinctive feature of this approach is the possibility of using composite blocks. Such blocks allow to store other functional blocks inside it, including composite ones. This makes it possible to simplify the system by grouping elementary functional blocks into containers that serve as separate subsystems.

Figure 2 shows the screenshots of the program at the modeling stage. This screenshot shows an example of a localization subsystem of iris image loaded from a file. Small blocks on the left represent source blocks and allow users input data into the program through the graphical interface. Small blocks on the right are receivers and allow users output data to the graphical interface. Blocks can be placed in special container (or composite) blocks, which allows to combine a group of functional blocks into a separate subsystem. Large blocks perform specials algorithms. This iris localization subsystem unites several different algorithms such as low-pass filtering [7], rank filtration [8], Hough transforms (to search for the center of the pupil) [9], an algorithm for pre-localization of the iris [10], Daugman's integro-differential operators for searching for internal and the outer boundaries of the iris [11] and the eyelid detector. The result of this subsystem is the attributes of the localized boundary of the iris. The following steps in the operation of the iris recognition system should include the normalization of the iris and the obtaining of the code (as an option using the methods of texture...
analysis used to study heterogeneous structures [12]). An example of a graphical user interface is shown in Figure 3.

![Figure 3. Screenshot of the program. The graphical user interface of modeling environment.](image)

In the screenshot placed the display of various intermediate data (visualization of the results of the localization of the iris, the result of the work of the rank filter for eliminating the eyelashes, the result of the transformation of Hough for the search for the center of the pupil).

5. Communication with biometric sensors

Biometric sensors are an external data source for a data flow program. To communicate with them, special blocks of data sources are implemented. Inside these blocks there is an interaction with the device with using drivers.

6. Prototype generation

At the third stage, the prototype is automatically generated from the model. The prototype is a file containing the structure of a dataflow graph and dynamically plugged binary modules with the logic of function blocks. The prototype is executed in a special interpreter, which reads the structure of the graph from the file, loads the necessary modules, and starts the execution of the program.

7. Modification of the approach. Use priorities for block activation

Block activation occurs after data is set on all input ports in dataflow programs. Two (or more) blocks which is ready for activation can be executed at the same time in different threads [7]. In some cases, this approach may be unacceptable. For example, when it is necessary to turn on the biometric sensor, read the data from it and turn it off. If the data is read in the program at once, it is possible to include the reading of biometric data and switching on/off the sensor into one functional unit. However, in the case when sensor reading occurs repeatedly during the execution of the program of data streams, the
permanent switching-off of the sensor is inadmissible. In such a case, it is necessary to create the three blocks for each operation of the sensor: the block for starting, stopping and reading. These blocks in the program will be executed at a time as they are sources, but it is necessary that the switching unit works first, then the read block (repeatedly) and at the end the block for closing. In this paper, the introduction of such a block parameter as the execution priority is proposed. In this case, blocks ready for activation at one time will be ranked by this priority and activated according to it.

8. Comparison with analogues
A literature review of known solutions based on data flow programming was conducted in this work. Most of the considered solutions were specialized for a specific application, for example, vvvv [13] was developed for prototyping interfaces, KNIME [14] was developed for working with large data and analytics. Such solutions as Lucid [15] are a separate non-visual programming language that requires the mastering of specific language constructs. The RiDE method [16] is also considered. The main disadvantage of RiDE can be attributed to the fact that individual computing nodes are supposed to be implemented as separate programs. Such an approach can affect performance, as it requires the use of interprocess communication tools when data is being transferring between computing nodes.

9. Conclusion
Thus, in this paper, a method for developing biometric systems based on modeling and prototyping was presented. The developed model can be used to generate prototypes, or integrated into the biometric system as a separate component for the parametrization and comparison of biometric samples.

The main stages of prototype development were described. A test system was developed to evaluate this approach. The advantages include:

- Rapid prototype development, testing in conditions close to real. Rapid development is achieved using a graphical development method.
- Allows to reveal new aspects of the developed biometric system. Due to the graphical way of representing the system, one can consider in general and analyze the developed model.
- Reusing the designed units. Blocks are stored in separate dynamically connected modules.

References
[1] Jain A K, Flynn P and Ross A A 2008 Handbook of Biometrics (Springer-Verlag London)
[2] Boulgouris N V and Konstantinos N P 2009 Biometrics: Theory, Methods, and Applications (Wiley-IEEE Press)
[3] Petrovska-Delacrétaz D and Chollet G 2009 Guide to Biometric Reference Systems and Performance Evaluation (Springer-Verlag London)
[4] Sousa T B 2012 Dataflow Programming Concept, Languages and Applications Dr. Symp. Informatics Eng 7 p 13
[5] Carkci M 2014 Dataflow and Reactive Programming Systems (Leanpub book)
[6] Hils D D 1992 Visual languages and computing survey: Data flow visual programming languages J. Vis. Lang. Comput 3(1) pp 69–101
[7] Nixon M S and Alberto S A 2008 Feature Extraction and Image Processing (Academic Press) p 88
[8] He Z Tan T Sun Z and Qiu X 2008 Robust eyelid, eyelash and shadow localization for iris recognition Proc. - Int. Conf. Image Process. ICIP pp 265–268
[9] Hilal A Daya B and Beauseroy P 2012 Hough Transform and Active Contour for Enhanced Iris Segmentation IJCSI Int. J. Comput. Sci. Issues 9(6) pp 1–10
[10] Minacova N N and Petrov I V 2015 Method of preliminary localization of the iris in biometric access control systems IOP Conf. Ser. Mater. Sci. Eng. 93(1) p 12056
[11] Daugman J 2004 How iris recognition works Circuits Syst. Video Technol. IEEE Trans. 14(1) pp 21–30
[12] Minakova N N Karpov S A and Ushakov V Ya 2002 Texture analysis of the disperse structure of composite elastomers with a modified carbon filler *Russian Physics Journal* **45**(2) pp 185-189

[13] vvvv - a multipurpose toolkit. https://vvvv.org/

[14] KNIME, https://www.knime.org/

[15] Ashcroft E and Wadge E 1985 *Lucid, the dataflow programming language* (APIC Stud. data Process)

[16] Bakhterev M Kazantzev A Vasev P and Albrekht I 2011 Dataflow-Based Distributed Computing System Proceedings of the Euromicro PDP pp 6-7