Automated Video Surveillance System based on Hierarchical Object Identification

Diana Zahorodnia, Yuriy Pigovsky,
Pavlo Bykovyy, Viktor Krylov
Ternopil National Economic University
Ternopil, Ukraine
dza@tneu.edu.ua

Anatoliy Sachenko, Agnieszka Molga
Kazimierz Pulaski University of Technology and Humanities in Radom
Radom, Poland
sachenkoa@yahoo.com

Abstract— **The structure, data models and implementation of the automated video surveillance system are proposed in this paper. The developed methods and basic information technologies for contour segmentation based on the Canny's method, and the methods of contour inflection points selection with the adjustable detailing allow us to carry out structural and statistical identification of hierarchical objects by their contour inflection points. Using the technologies proposed by the authors, the automated video surveillance system can identify objects with different levels of detailing. This allows us to increase its efficiency under the conditions of limited computing resources.**

Keywords—automated video surveillance system; contour segmentation; inflection point; hierarchical identification and classification

I. INTRODUCTION

Having analyzed the existing automated video surveillance systems [1-4], we can see that the number of cameras in them is constantly increasing and, accordingly, there is the need for increasing the number of additional operators for their maintenance and resource intensity. The use of Intelligent Video Surveillance Systems [5], enables to solve the problem with additional operators. However, such systems reduce the efficiency and require significant computing resources [5-6].

In order to increase the efficiency of the automated video surveillance systems, it is proposed to reduce the amount of data processed at the stages of object segmentation and identification, using the methods of wavelet analysis [7-8]. This enables to select video objects and details of objects of different geometric sizes according to the specified goal of processing [9-11].

In previous work authors proposed the usage of hierarchical methods and basic informational technologies of contour segmentation for inflection points (IP) selection and identification. This allowed us to increase the efficiency of the applied intelligent information systems of video surveillance with sufficiently reliable identification [12-15]. At the same time, the proposed approach requires further research within the framework of systematization and implementation, which is discussed below.

II. STRUCTURE OF THE SYSTEM

The proposed Automated Video Surveillance System (AVSS) consists of the following hardware and software components (Fig. 1):

- Television sensors (photo-, video cameras) provide an effective mechanism for creating an electronic version of the image;
- ADC (analog-to-digital converter) converts and compresses images into supported formats, and the encoding and storage unit provides recording and playback of digital video signals;
- Image pre-processing provides conditions for increasing efficiency and quality of selection and identification of the objects which are being detected or investigated;
- Selection of moving objects allows to assess automatically the change of the object visual environment that is viewed by television camera. It is used to alert to the appearance of moving objects in the protected area and is based on a comparative analysis of images of the "current" and "previous" frames stored in the frame memory;
- The selected object is localized, in this case, the object face is selected in the frame from the video stream;
- Hierarchical contour segmentation provides splitting the image into nested contours. This image segmentation results in a set of contours selected from the image. This allows to reduce significantly the amount of data which is processed and increase the efficiency of the AVSS;
- IPs are selected on the contour of each hierarchical level (the coordinates of IPs are recorded into the array);
- After that, structural and statistical identification of objects is carried out;
- Classification – the procedure for referring the object under study (face image), given by the set of observations, automatically to one of the mutually exclusive classes of people, or conclusion is drawn that the object does not belong to any of the known classes.

978-1-5386-1495-2/18/$31.00 ©2018 IEEE

- On the basis of the classification, the human operator makes the final decision (to point the camera closer at the object, to inform the security guard, to ignore).

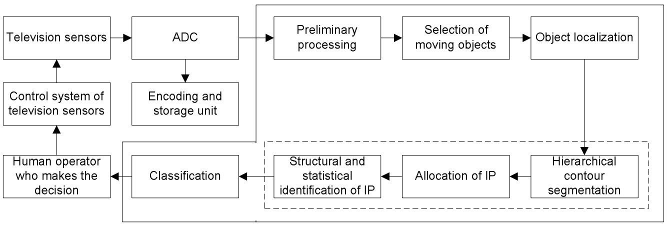

Fig. 1. Structure of the AVSS with the proposed subsystems.

The software implementation of AVSS with proposed subsystems is described below.

III. Database

As it is shown in Figure 1, the AVSS processes both current input data and stored and entered data you want to search. Such a large amount of data is usually stored in a database to provide interconnectivity, minimal redundancy, program independence, integrity, and protection from unauthorized access [16].

The subject domain of the subsystem of the AVSS for structural hierarchical identification can be presented in a tabular form. Thus, it is advisable to use a relational database model, supporting database management systems, in particular SQLite [17], which can provide the maximum efficiency and speed of data processing.

For designing the infological model of the subject domain of the subsystem for hierarchical identification, the graphic language of ER-modeling [16] is used (Fig. 2).

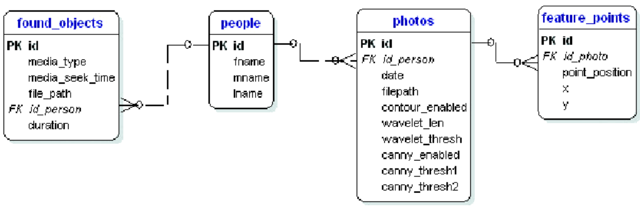

Fig. 2. Data scheme of the subsystem for structural hierarchical identification.

The "people" table shows information about the objects included into the system; the "photos" table contains files of object images to be searched for, length settings and threshold size of wavelet functions, the size of the thresholds of the proposed method based on the Canny's method; the "feature_points" table shows information about IP contours of the object images to be searched for; the "found_objects" table contains information about the found objects: the types of the processed files, the time and date of the objects having been found in the table "people".

This structure is implemented on the basis of the DBMS SQLite. This provides flexible access and data management while using the AVSS.

IV. Data Models, Components and Methods of the Designed System

Software is implemented using object-oriented approach, design patterns, and methods for minimizing complexity of components. In an object-oriented program, the essence of the subject domain is presented by classes. The level of abstraction depends on the specific problem and helps to avoid insignificant details while solving this problem.

Design patterns are standard solutions to typical programming tasks. Several design patterns have been used while developing software, including: singleton, dependency injection, model-view-presenter [18-19].

The data models of software implementation of the AVSS database entities are described by the following components: Person, Photo, FeaturePoint, DetectedPerson. The UML diagram of their classes is shown in Fig. 3 [20]. The "Person" model describes the person who will be identified by the AVSS. This model contains all the properties of the corresponding database table and additional methods that allow you to display information about the person in a readable form and create new objects of this model.

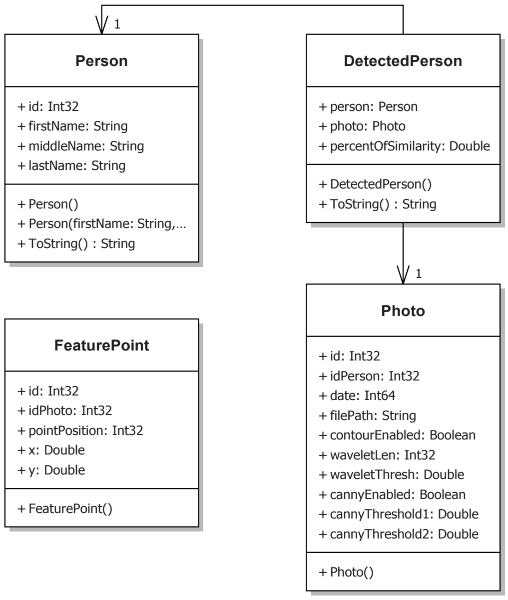

Fig. 3. UML – diagram of the AVSS data models.

The "Photo" model contains settings for contour segmentation methods, the path to the video file of the detected face and the date of entry to the database. The "FeaturePoint" model represents the IP table. The "DetectedPerson" model describes the degree of similarity between a set of IPs of a new image and a particular person. This model is related to the "Person" and "Photo" models in such a way that the field percentOfSimilarity describes the percentage of similarity between the IPs and a certain photo of a person.

Data access is arranged using Data Access Objects (DAO). This arrangement allows us to move easily from one DBMS to another and simplifies the unit software testing. AVSS Data Access Objects are shown in Fig. 4.

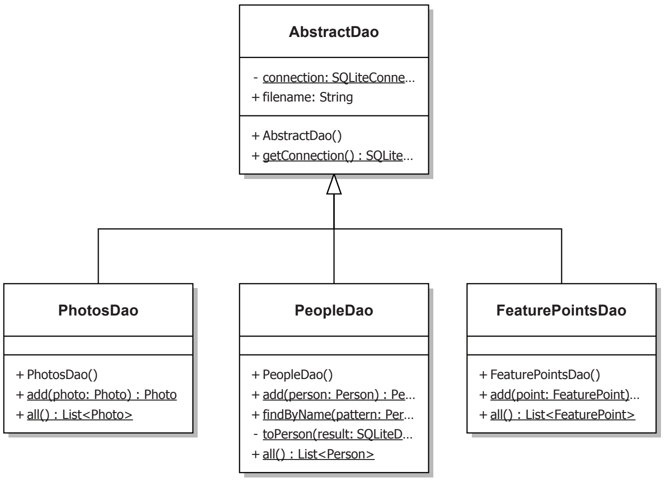

Fig. 4. UML – diagram of Data Access Objects classes.

The "AbstractDao" component includes the common properties of all Data Access Objects, namely: the getConnection method, which performs a Lazy initialization of the "connection" field using an encapsulated tape connection to the database. The value of this tape allows you to change the type of DBMS. Since, SQLite of DBMS, which is used in this study, stores data in the file system then, the "filename" field specifies the path to the database.

The "PeopleDao" component allows to create, search and retrieve collections of all objects of the "Person" class. The "add" method is used to create a new object. This method receives a pattern of an object describing a new person at the entrance, and returns the same object at the exit, completing it with a unique identifier (id) of the newly created record in the database.

The search for a person object with the use of the "findByName" method is based on his name, second name and surname. The method implementation results in the person object with a unique identifier when there is a corresponding record in the database, otherwise, the method will return the reference to an empty object.

The result of the "all" method implementation is the collection of all person objects that were recorded into the database. Moreover, this collection may be empty.

The private method "toPerson" is intended to transform the untyped entity of the database into a strictly typed "Person" object.

The "PhotosDao" and "FeaturePointsDao" components perform function, which is similar to the function of "PeopleDao" component, but with the objects of the "Photo" and "FeaturePoint" classes. That is, their "add" methods create an object in the database and complement it with a unique identifier and the "all" methods read out this data.

The model-view-presenter (MVP) paradigm is chosen for software implementation that allows us to minimize the complexity of the software and facilitate the writing of the unit tests. Data processing logic, which is called business logic, is included into the presenters. Components that implement the view abstraction display data in the user interface (UI). The data, which are exchanged between the presenters and display components, are called models.

Fig. 5 shows a diagram of the MVP classes for adding new objects in the AVSS. The "AddFaceView" component provides indirect access to a set of settings of contour selection methods that are derived from the user interface, including the one implemented by the "AddFaceForm" component. Indirect access allows you to add easily and unmistakably new types of the user interface without any changes to the "AddFacePresenter" component of video data processing.

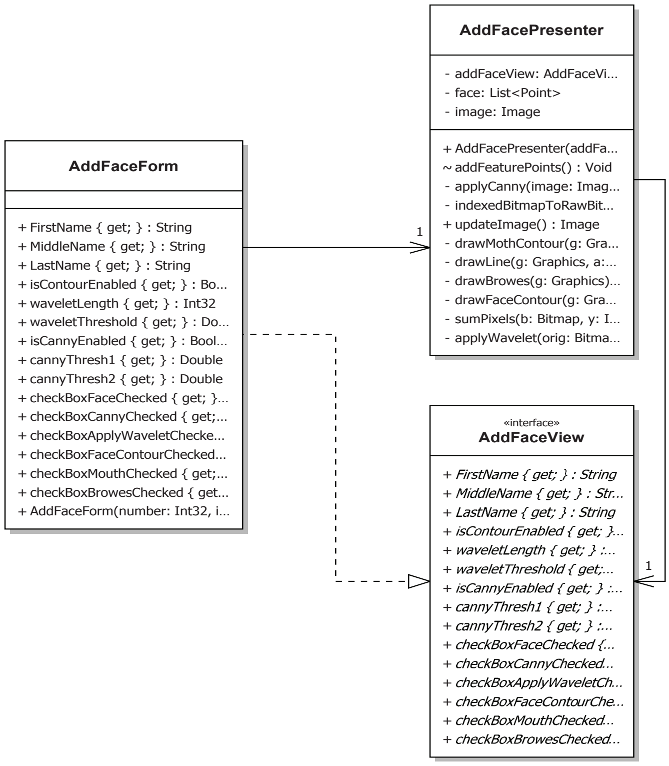

Fig. 5. MVP class diagram for adding new objects.

The "AddFacePresenter" component contains methods for displaying contours of video data at different levels of detailing using different threshold values, according to the settings of the "AddFaceView" component.

The "applyCanny" method filters input video data due to the Canny's method, using "isCannyEnabled", "CannyThresh1" and "CannyThresh2" parameters. Before the processing, it stores a copy of the original video data that allows you to experiment easily with the parameter values and cancel previous transformations.

The "applyWavelet" method performs wavelet transformation of video data using "checkBoxApplyWaveletChecked", "waveletLength" and "waveletThreshold" parameters. This method uses the auxiliary function of "sumPixels", which performs addition of values of the adjacent pixels brightness in a line of video data.

The "updateImage" method, based on the two above-mentioned methods, creates a resulting image and allows us to overlay different characteristic features, detected at various levels of detailing with the use of the auxiliary functions such as "drawFaceContour", "drawMounthContour", and "drawBrowes". These functions, in turn, use the "drawLine" procedure, which shows graphic primitives in the resulting pattern of video data.

Fig. 6 shows a diagram of the MVP components for selecting contours in video files. The "FeaturePointComparationType" component describes the possible methods for calculating the distances between the IP vectors. You can select the appropriate method using the "Settings" menu item in the main window of the AVSS.

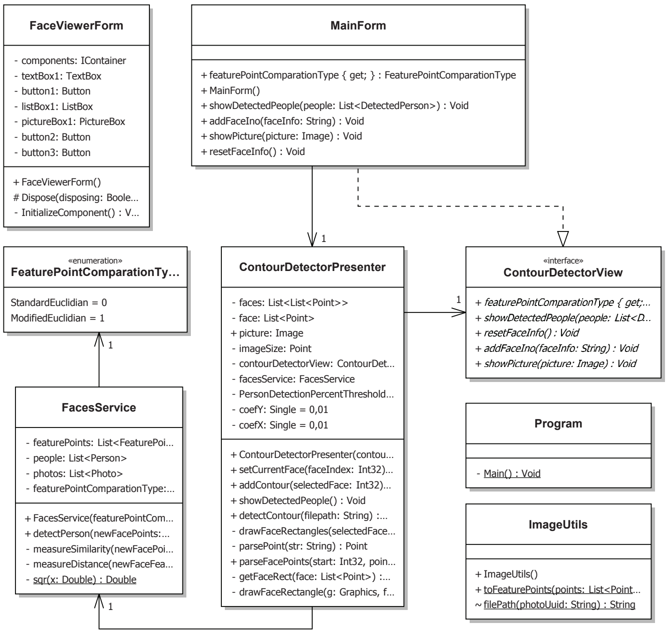

Fig. 6. MVP components diagram for selecting contours in video files.

The "FacesService" component performs calculation of the degree of similarity between the IP vector of a new pattern of video data and the IPs of known faces in the cluster base.

The "ImageUtils" component performs the normalization of IP vectors.

The "ContourDetectorPresenter", "ContourDetectorView" and "MainForm" components are intended to select a set of faces in video file and provide an interactive viewing of the results of their classification using cluster analysis.

The "detectContour" method of the "ContourDetectorPresenter" component receives at the entrance the path to the video file and creates a list of IPs of all faces detected in video using the "parseFacePoints" auxiliary function. The "parseFacePoints" function is implemented in a recursive form by processing certain IPs due to the "parseFacePoint" function.

The "drawFaceRectangles" method with the use of "drawFaceRectangle" and "getFaceRect" functions visually selects the faces detected in video, highlighting them with a rectangular frame with a serial number.

The "showDetectedPeople" method initiates the process of displaying the results of a cluster analysis for the classification of face objects detected in video. It is used at each change in the type of classification methods and their parameters.

The "Program" component launches the main window of the AVSS and prepares the resources necessary for its work.

The "FaceViewerForm" component is intended for viewing, responding, and removing the IP vectors of faces detected in a database.

V. User Interface of the AVSS

The described components were implemented as AVSS of object identification in the Visual Studio 2015 environment for the .NET platform in C# programming language using OpenCV libraries. The main units of AVSS allows to download and analyze video/images, select contours of the faces and the contour IPs, build identification vectors on the basis of which the classification of objects is carried out. It consists of a window of video data input, a window of new objects input and a window of view/edit the existing objects.

A window of object identification (Fig. 7) allows to open a video file, in which the face detection unit highlights the face objects as rectangles and shows their number.

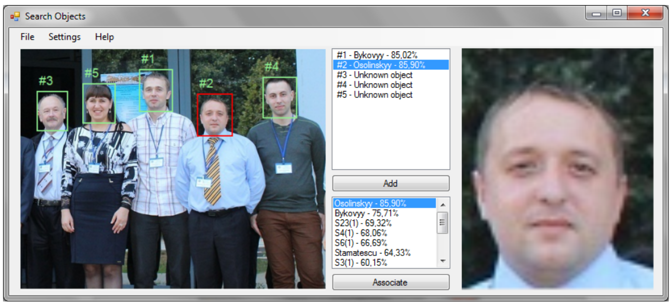

Fig. 7. A window of the AVSS object identification with the information about detected faces.

The contour segmentation method, based on the Canny's method for searching contours in order to obtain a sequence of image skeletons, is applied to the selected objects; for each contour the IPs are selected using the wavelet analysis method of the curvature function, an identification vector is built on the basis of which the classification is performed at each level of detailing, and at the following level of classification only the objects that have been classified at the previous level are compared. At the last level, the surname and name of the object with the highest match percentage are displayed. If the level of matching is low (less than the threshold setting), the object is marked as "#number - Unknown object".

The window allows you additionally to view the percentage of similarity between the faces which are in video and those, which are in the database of objects, arranged according to descending the percentage of similarity. Object selection allows you to view its image in the form on the right.

Selection of the option "Settings->View Objects" in the menu shows the viewing and editing form of video clusters. Each cluster of video data is described by the name of the person that corresponds to it. Selection of a surname causes downloading of the list of video data associated with it.

Selection of the "Associate" button allows you to refer the undetected object to an existing cluster in the database.

According to the system settings, after the unknown object has been selected and the "Add" button has been clicked, the form for new objects input can be opened (Fig. 8). Surname, name, patronymic name can be written in this window; the length of the wavelet, the size of the threshold, the contour segmentation thresholds can be determined; the selected contours can be viewed and the corresponding object can be added to the system.

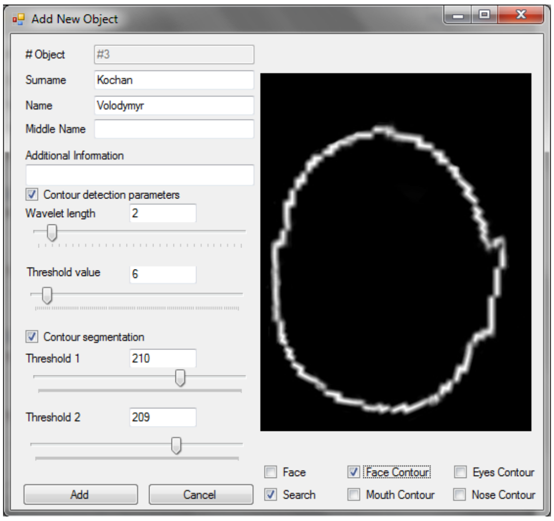

Fig. 8. A window of new object input into the AVSS database.

VI. Experimental Results

In order to evaluate the time complexity of AVSS, it was investigated the face processing time of different frames using procedures of structural and statistical identification of hierarchical object identification (Table I).

TABLE I. Comparison of Frame Processing Time with Different Number of Objects (Faces)

Number of objects in the frame	Frame processing time using (Yes) and without use (No) of hierarchical method, sec		Reduced duration using the method, times
	Yes	*No*	
1	0,05	0,75	15
3	0,08	0,96	12
5	0,1	1	10
10	0,3	2,4	8

The accuracy of classification results is shown in Table II, where in two columns Yes/No were implemented/not implemented the procedures of structural and statistical identification of hierarchical object identification.

TABLE II. Comparison of the Object Classification Accuracy for Frame with Different Number of Objects

Number of objects in the frame	Frame processing accuracy using (Yes) and without use (No) of hierarchical method, %		Difference in processing accuracy, %
	No	*Yes*	
1	100	95	5
3	95	88	7
5	93	85	8
10	90	80	10

As we can see from the tables above, after the increasing of number of people in the frame by 8-10 times, proposed AVSS make decisions quickly (Table I), but they are not always reliable (Table II). In case, of not reliable or clear results the classification task is assigned to TV operator with makes the final decision.

VII. Conclusions

The existing video surveillance systems have been analyzed and it has been found that their efficiency can be increased. It is proposed to apply a hierarchical approach to the segmentation and identification procedures of geometric objects in order to increase the efficiency of the automated video surveillance systems. To implement the hierarchical approach, the methods of wavelet analysis was selected, also the structure of automated video surveillance system was proposed.

According to proposed structure, the database, data models, components and methods for structural and statistical identification of hierarchical objects by the contour inflection points was implemented.

The software implementation of automated video surveillance system was presented. The studies showed that the use of proposed system allowed us to reduce the time of frame processing in 8-15 times (according to the content of the frame and the number of objects in it) with a slight decrease in reliability (5-10%).

References

[1] C. Norris, M. McCahill, D. Wood, “The growth of CCTV: a global perspective on the international diffusion of video surveillance in publicly accessible space,” Surveillance and Society, vol. 2, no. 2/3, 2004, pp. 110-135.

[2] S. Fleck, W. Strasser, “Smart camera based monitoring system and its application to assisted living,” Proceedings of the IEEE, vol. 96, no. 10, 2008, pp. 1698-1714.

[3] M. Brezovan, C. Badica, “A review on vision surveillance techniques in smart home environments,” Proceedings of the 19th International Conference on Control Systems and Computer Science, 2013, pp. 471-478.

[4] P.V. Kale, S.D. Sharma, “Review of securing home using video surveillance,” in International journal of science and research (IJSR), vol. 3. issue 5, May 2014, pp. 1150-1154.

[5] K. Alok, K. Singh, S. Rajeev, “Performance evaluation of various moving object segmentation techniques for intelligent video surveillance system,” Int. Conf. on Signal Processing and Integrated Networks (SPIN), 2014, pp. 196-201.

[6] Dramatically Reduce the Cost and Complexity of Video Surveillance Storage. Oracle white paper. February 2015. – 10 p. [Online]. Available: http://www.oracle.com/us/ solutions/oos/tiered-storage-infrastructure/ wp-video-surveillance-storage-2288409.pdf.

[7] M. Polyakova, V. Krylov, “Classification of methods of the signal semantic wavelet transform for image contour segmentation,” in Int. Journal of Computing, 2008, vol. 7, issue 1, pp. 51-57.

[8] A. Koval, B. Rusyn, “Method for construction of biorthogonal wavelets based on the theory of poles,” in Journal of Automation and Information Sciences, vol. 33, no. (9-12), pp. 29-39, 2001.

[9] V. Mashtalir, D. Kinoshenko, E. Yegorova, “Clustering method for fast content-based image retrieval,” in Computational imaging and vision / M.A. Viergever, ed. – Dordrecht: Springer, 2006, vol. 32, pp. 946-952.

[10] Jun Liang, Yanning Zhang, Steve Maybank, Xiuwei Zhang, “Salient feature point detection for image matching,” IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP), 2014, pp. 485-489.

[11] R. C. Gonzalez, R.E. Woods, Digital Image Processing, 3rd Ed. Pearson, p. 976, 2007.

[12] I. Paliy, A. Sachenko, Y. Kurylyak, O. Boumbarov, S. Sokolov, “Combined approach to face detection for biometric identification systems,” Proceedings of the 5th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2009), September 21-23, 2009, Rende (Italy), pp. 434-439.

[13] I. Paliy, V. Dovgan, O. Boumbarov, S. Panev, A. Sachenko, Y. Kurylyak, D. Zagorodnya, “Fast and robust face detection and tracking framework,” Proceedings of the IEEE 6th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS’2011), Prague (Czech Republic), September 15-17, 2011, vol.1, pp. 430-433.

[14] D. Zahorodnia, Y. Pigovsky, P. Bykovyy, V. Krylov, I. Paliy, I. Dobrotvor, “Structural statistic method identifying facial images by countour characteristic points,” Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS’2015), Warsaw (Poland), September 24-26, 2015, pp. 293-297.

[15] D. Zahorodnia, Y.Pigovsky, P. Bykovyy, “Canny-based method of image contour segmentation,’ in International Journal of Computing. Ternopil: TNEU, 2016, vol. 15, issue 3, pp. 200-205.

[16] T.M. Connolly, C.E. Begg, Database Systems: A Practical Approach to Design, Implementation and Management. 5th Ed. Pearson, p. 1400, 2009.

[17] J.A. Kreibich, Using SQLite. 1st Ed. O'Reilly, p. 530, 2010.

[18] E. Gamma, R. Helm, R. Johnson, J. Vlissides, G. Booch, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional; 1st ed., p. 395, 1994.

[19] S. McConnell, Code Complete: A Practical Handbook of Software Construction: A Practical Handbook of Software Costruction Microsoft Press; 2 ed edition, p. 960 , 2004.

[20] C. Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice Hall; 3 ed.. p. 736, 2004.