

Efficient Visualization for an Ensemble-based System

János Tóth¹, Róbert Tornai¹, Imre Labancz², András Hajdu¹

¹University of Debrecen, Faculty of Informatics, Kassai út 26, H-4028 Debrecen, Hungary, e-mail: {toth.janos, tornai.robert, hajdu.andras}@inf.unideb.hu

²University of Debrecen, Institute of Educational Studies and Cultural Management, Egyetem tér 1, H-4032 Debrecen, Hungary, li0003@stud.unideb.hu

Abstract: Ensemble-based systems have proved to be very efficient tools in several fields to increase decision accuracy. However, it is a more challenging task to become familiar with the operation and structure of such a system that contains several fusible components and relations. In this paper, we describe a visualization framework in connection with an ensemble-based decision support system in the domain of medical image processing. First, we formulate the operations that can be used for composing such systems. Then, we introduce general visualization techniques for the better interpretability of the components and their attributes, the possible relations of the components, and the operation of the whole system as well. Our case study assigns the general framework to image processing algorithms, fusion strategies, and voting models. Finally, we present how the implementation of the visualization framework is possible using the state-of-the-art 3D collaboration framework VirCA. The proposed methodology is suitable for both visualization and visual construction of ensembles.

Keywords: customizable content management; information visualization; application generation; collaboration arena; 3D Internet

1 Introduction

Using ensemble-based systems [1] is a rather popular approach in several application fields [2, 3], since such a system usually outperforms any of its members in terms of accuracy. An ensemble-based system is constructed by selecting and combining members that have diverse operating principles or models using an appropriate strategy in order to solve a given (machine learning) problem. In our former practice, we also successfully adopted this methodology to compose an ensemble-based system for the screening of diabetic retinopathy based on the processing of digital retinal images [4]. In our system, ensembles are created at multiple levels containing components having the same detection or classification tasks [5, 6]. The complete decision process currently includes the

execution of 38 algorithms that can be started also in a strict order, since their operations depend on each other's outputs. When our system grew large, we faced the problem how we should make it easily interpretable and configurable for interested users. Classic techniques like flowcharts or UML diagrams [7] are not appropriate in our case since they do not provide visual tools for specific operations and elements that are considered in an ensemble-based system. To address this issue, in this paper we introduce a 3D visualization framework for the better understanding and easier construction of ensemble-based systems.

The comprehension of an ensemble-based system requires the creation of mental models on several levels of abstraction. An appropriate visualization framework, which takes advantage of the strengths of human cognition but also takes the human limits, needs and behavior [8] into account, can significantly facilitate the creation of mental models and thus support the reasoning about the system.

To measure the visualization tools that are necessary for our system, first we formally define the general rules for ensemble creation. These steps will include the possible fusion of components having different functionalities, and the organization of components having the same functionality into ensembles. Besides these tasks, we need tools to visualize the components that can be also interpreted as a set of attributes. For the description of the properties of components we can use color, shape, and size features [9]; however, to support more complex parameter settings we consider attribute panels too. Components belonging to the same functionality groups are visualized by their spatial arrangement, as well.

As we propose techniques for a decision support system, we also need specific elements that are necessary to evaluate the accuracy of the system in terms of the reliability of its decision. Such evaluation can be made taking specific error (energy) functions into consideration with testing on specific databases, which elements need visualization as well. In this way, the performance of a system can be evaluated. When the aim is to compose a system that is optimal regarding a specific error function, the necessary components and decision rules can be determined by optimization algorithms, which process is called automatic application generation. For an automatic generation, the proposed visualization tools help the users discover the automatically selected components and their relations. On the other hand, the users are allowed to compose an ensemble by manually selecting its components and defining the relations between them. Thus, to support this form of interaction we also introduce visual elements and tools for the selection of components and performing operations. This type of interaction with the system is called manual application generation [10]. As a result, after selecting a database and an appropriate energy function, the users can evaluate the performance of the ensemble composed by them.

As it can be seen, our aim is not only to visualize an ensemble with an already fitted model but also to allow efficient interaction between the users and a system that can be considered an artificially cognitive one [11] due to its decision-making

and self-adjusting capabilities. The users can learn from the applications (ensemble setups) generated automatically by the system and can create applications fitting better their data processing needs using the acquired knowledge. That is, the decision making efficiency of the system can be improved based on the blending of human and artificial cognitive capabilities [12, 13].

In our case study dedicated to diabetic retinopathy screening based on digital images, the components are image processing algorithms. These algorithms belong to specific functionality groups, e.g. based on whether their aims are image preprocessing, the detection of anatomical parts or lesions, etc. Algorithms having the same functionality can be organized into ensembles to raise the accuracy of that given functionality. Moreover, it is also possible to fuse algorithms that have different functionalities (e.g. a preprocessor can be fused with a detector to gain a new detector algorithm). The proposed general visualization framework will be explained on this specific system. As for the implementation of the visual framework supporting both automatic and manual application generation, we have selected the state-of-the-art 3D collaboration framework VirCA [14, 15]. We present how this VIRtual Collaboration Arena is capable of meeting the visualization and interaction requirements of our methodology.

The rest of the paper is organized as follows. In Section 2, we introduce our formal description for the composition of ensemble-based systems being investigated, and summarize the requirements for the visualization of the elements. In Section 3, we discuss on how cognitive biases affecting the perception of a visual scene are addressed in our approach for the visualization of the system. Section 4 contains our case study together with the proposed visualization techniques and a description of its elements represented by an XML schema. In Section 5, we present how our approach can be implemented in the VirCA system. Finally, some conclusions are drawn in Section 6.

2 Formal Description for Ensemble-based Systems

In this section, we give a general formal description for the ensemble-based systems discussed. The formalization covers all such types of members, and operations between them that can be used to compose a complete system. Then, using this general model, we will be able to list all the operators and operands (components) and also the results of such operations that need to be visualized. In later sections, we will give a concrete realization of the proposed formalism regarding the operations, and also an application with concrete components.

We start with defining possible functionalities F_1, F_2, \dots, F_N assets containing components $C_{1,1}, \dots, C_{1,M_1}, C_{2,1}, \dots, C_{2,M_2}, C_{N,1}, \dots, C_{N,M_N}$ having the corresponding functionality:

$$F_i = \bigcup_{j=1}^{M_j} C_{i,j}, \quad i = 1, \dots, N. \quad (1)$$

The cardinality $|F_i| = M_i$ can be arbitrarily large, that is, the number of components having the same functionality can be extended freely. In our interpretation, a component will be a concrete algorithm having a specific functionality.

Since we let the components interact in our system, we go on with defining possible operations between components. For this aim, note that operations are needed between components having both the same and different functionalities. In case of same functionalities, some components can be grouped together to form an ensemble at functionality level. Since these ensembles can be considered as new components having the same functionality, formally we define this element as a function instead of a simple relation. Thus, for the functionality F_i we define the following function to set up ensembles from the components $C_{i,1}, \dots, C_{i,M_i}$:

$$ENS_i : F \subseteq F_i \times F_i \rightarrow F_i, \quad i = 1, \dots, N. \quad (2)$$

Note that with definition (2) we let the creation of ensembles that have only two members; however, larger ensembles can be easily generated by applying ENS_i multiple times.

Besides creating ensembles, we also allow the creation of new components by merging components having possibly different functionalities. The new components must belong to an existing functionality, which may be different from any of the ancestor components. Thus, we introduce the following fusion operation between components $C_{i,j}, C_{i',j'}$ with $i, i' \in \{1, \dots, N\}$, $i \neq i'$:

$$FUS_i : (C_{i,j}, C_{i',j'}) \in F_i \times F_{i'} \rightarrow F_k, \quad k \in \{1, \dots, N\}. \quad (3)$$

Regarding the possible number of basic components that can be fused, we can make the same comment as for (2). That is, by applying the fusion operator FUS more than once, several algorithms can be merged. In our practice, a merged component will have the functionality of either of its ancestor components; however, we do not need to apply this restriction in our formalization.

Besides the above operations to set up an ensemble-based system, we need some other special elements regarding evaluation and optimization purposes. We need databases DB_i for two reasons: First, in the case of manual selection of ensemble components, the created system can be evaluated on a given database. Second, in the case of automatic generation of the system, the database can be used during the optimization process to find the components of the system by data mining algorithms. Besides databases, energy functions EF_i should be considered for the same two reasons. That is, for manual generation, the user can see the accuracy of the system regarding a given energy function. Moreover, in the case of automatic generation, the optimization is carried out using the energy as the objective function.

For a more detailed presentation of the components, the visualization of their attribute values is also necessary. Such attributes can be the name, accuracy, speed, and controlling parameters of the component. That is, a component $C_{i,j} \in F_i$ formally can be split further into a collection of attributes:

$$C_{i,j} = (A_{i,j,1}, A_{i,j,2}, \dots, A_{i,j,T}), \quad (3)$$

where the number of attributes T can be component-specific, so in general we do not restrict its value, just leave as an arbitrarily large integer. However, in practice, components that have the same functionality should have the same number of attributes.

As a summary, in Table 1 we collect all those elements from the above formalization that needs visual representation. Note that, in case of manual application generation, selectability should be supported, as well.

Table 1
Elements that need visualization for an efficient presentation of the whole system

F_i	Visualizing different functionalities
$C_{i,j}$	Visualizing/selecting components belonging to different functionalities
$A_{i,j,1}, A_{i,j,2}, \dots, A_{i,j,T}$	Visualizing attributes of the components
F	Visualizing the subset for ensemble creation, showing selectable components
ENS_i	Visualizing the resulted ensemble
DB_i	Visualizing/selecting databases for testing/evaluation
EF_i	Visualizing/selecting energy functions for testing/evaluation
$(C_{i,j}, C_{i',j'})$	Visualizing a pair of components for fusing, showing fusible components
FUS_i	Visualizing the fused component

3 Cognitive Aspects and Biases

The comprehension of the operation principles of an ensemble-based system requires the creation of mental models at several levels of abstraction, taking into consideration the operation of the individual components, the possible component fusions, the ways of ensemble creation, and also the system as a whole. An appropriate visualization framework that assigns easy-to-recognize visual elements to the concepts and the components can significantly facilitate this process, and thus, support the reasoning about the system.

During the construction of the corresponding visualization, we have to take into account the possible cognitive biases of the users as well. Cognitive biases are patterns of deviation in judgment that occurs in particular situations, and the fundamental attribute of these is that they manifest unconsciously. According to recent psychological studies [16, 17], cognitive biases are heuristics selected by evolutionary pressure. Therefore, they are not flaws but features of human cognition, which emerged in order to aid rapid decisions.

From our perspective, the most important question is how these biases affect the perception of a visual scene. Based on their past experiences and everyday interactions with objects, humans continuously develop their concepts about how certain things should appear and behave, and where they should be located in various situations. As humans tend to seek for evidence that confirms what they accept as true, the visual representation of the system will also be viewed in this way. That is, the users will perceive and try to match the visualization principles and elements of the system to their own concepts.

The relative position and appearance of the visual elements are also very important from a human comprehension perspective. The basic principles of the Gestalt psychology [18] (e.g. law of proximity, similarity, symmetry, "common fate", and closure) describe how humans perceive visual objects. These principles have to be considered in a visualization method to compose logical groupings and visual hierarchies.

If the appearance, arrangement or the expected behavior of the elements of a visualization technique opposes the cognitive heuristics and the most common concepts, it highly reduces its usability and efficiency. This is particularly true in the case of those visual elements that are critical or frequently used in a user's work-flows. Moreover, the consistency of the functionalities assigned to the visual elements has to be maintained as well, as humans expect similar elements to behave similarly.

An efficient visualization technique has to be able to display information that is semantically related in the given context and to allow the users to freely explore them. In our case, it affects, e.g. the visualization of the possible fusions and ensembles, and component properties. Displaying information within context is primarily preferred, but the visual scene has to be created in that way that it diminishes the cognitive overload, as well. The visualization framework also has to avoid employing menus or other elements that do not fit the visualization logic, in order to keep up the focus and attention of the user [19]. Our efforts to translate the manipulation of elements to physical, real-world-like interactions have been motivated also by this issue to avoid menu-based manipulation.

Our approach for visualization takes the features of the process of understanding and reasoning into account to suppress the negative effects of cognitive biases besides taking advantage of the strengths of human cognition.

4 Case Study

In this section, we introduce general visualization techniques for the elements described in section 2, to facilitate the interpretability and efficient construction of ensembles. These techniques are presented through a medical decision support system [3] that aims to detect the signs of diabetic retinopathy (DR) on digital retinal images.

As the manual DR grading of retinal images is a slow and resource-intensive task, automated software systems which can distinguish healthy retinas from pathological ones are welcome, in order to perform triage and to pre-screen patients prior to further medical examinations. The users of such a system have to be allowed to customize its operation according to their purposes, therefore a suitable visual representation of the system components and their attributes, and the concepts corresponding to the ensemble creation and component fusion are necessary.

The visualization framework that we introduce employs manipulable objects arranged in the 3D space to represent the different elements of the system. In this space, the point-of-view can be freely moved by the user to interact with a specific element. Compared to conventional user interfaces, this direct manipulation-like behavior has benefits, mainly during the learning phase [20]. As the human visual system is able to quickly recognize different scenes [21], the time needed to get an overview of the system components and their relations can be reduced as well through showing them in a more natural, spatial arrangement.

4.1 Visual Representation of the System

4.1.1 System Functionalities

To realize its goal, our system contains a number of different image processing algorithms that belong to specific functionalities (F_i). The list of these functionalities is given in Table 2.

Table 2
Functionalities of our image-analyzing system

F_1	Region of interest detection
F_2	Vascular system detection
F_3	Image preprocessing
F_4	Optic disc detection
F_5	Macula detection
F_6	Exudate detection
F_7	Microaneurysm detection

Different system functionalities are visualized as sets of the corresponding components, having common appearance and grouped by their spatial proximity. The components are enclosed by a borderline to contribute to the easier distinction of groups. The label of the group is affixed to the top of this borderline (see Fig 1).

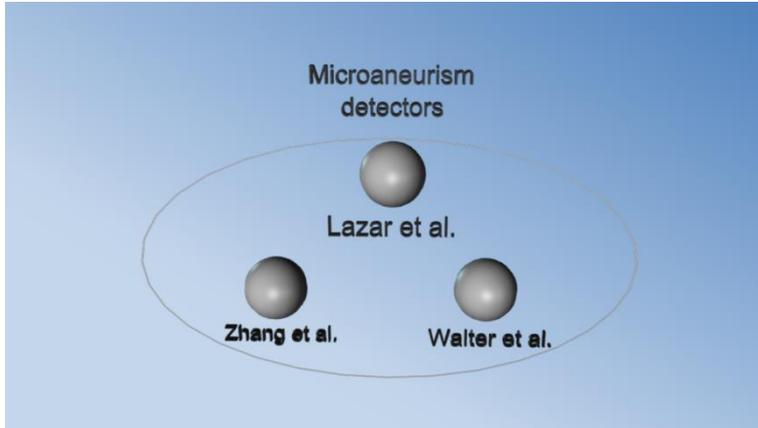


Figure 1

Visual representation of a functionality group

4.1.2 System Components and Their Attributes

In our system, we consider the different image processing algorithms as components $C_{i,j} \in F_i$. For example, the components of the F_7 -functionality are given in Table 3.

Table 3

List of the microaneurysm detector components of our system

$C_{7,1}$	Lazar et al. - rotating cross-section based microaneurysm detector
$C_{7,2}$	Walter et al. - bounding box closing based microaneurysm detector
$C_{7,3}$	Zhang et al. - 5 Gaussian filter based microaneurysm detector

Each component has a specific number of attributes, whereof three are common: the state, the name, and the description attributes. In our system, the components have two possible states (selected and not selected) what is indicated by the generally used colors green and gray, respectively. The names of the components are displayed as simple text labels. For example, the attributes of the component $C_{7,1}$ is given in Table 4.

The state of a component can be toggled with a point-and-select gesture. On a selected component, the user is enabled to perform the following manipulations using its icon menu [15]:

- displaying the attribute panel (gear icon) on which the user is allowed to set the parameters of the component with standard user interface elements, like sliders, spinners, and input boxes, etc. rendered in the 3D space (see Fig. 2);
- initiating ensemble creation (voting hand icon) and show the selectable components within a functionality group to form an ensemble with;
- initiating algorithm fusion (zipper icon) and show the selectable components in other functionality groups for algorithm fusion;
- displaying information about the component (info icon), including description of the method, explanation of its parameters, and its accuracy measured on different databases, if applicable.

Table 4

Attributes of the Lazar et al. microaneurysm detector algorithm

	<i>Attribute</i>	<i>Type</i>
$A_{7,1,1}$	State	Boolean value
$A_{7,1,2}$	Name	String
$A_{7,1,3}$	Description	String
$A_{7,1,4}$	2D smoothing parameter	Boolean value
$A_{7,1,5}$	Smoothing radius parameter	integer value
$A_{7,1,6}$	Smoothing sigma parameter	real value
$A_{7,1,7}$	Levels parameter	integer value
$A_{7,1,8}$	Threshold parameter	integer value
$A_{7,1,9}$	Accuracy	real value

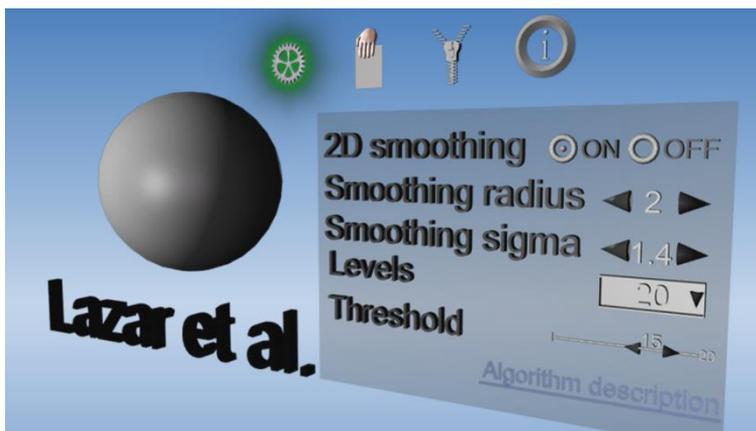


Figure 2

A system component with its attribute panel

4.1.3 Ensemble Creation and Algorithm Fusion

Components having the same functionality can be organized into ensembles in order to raise the accuracy of the given functionality. As for ensemble creation, we consider majority and weighted majority voting models, depending on the member components.

In our visual representation, ensemble creation can be initiated using a components icon menu. For clarity, if necessary the components are spatially rearranged before the subset F of components available for ensemble creation is visualized with arrows pointing at them from the selected one (see Fig. 3). The user can select any of these components and finish the operation using the icon menu again. The components of the result of the ensemble creation (ENS_i) are represented through green color and spatial grouping.

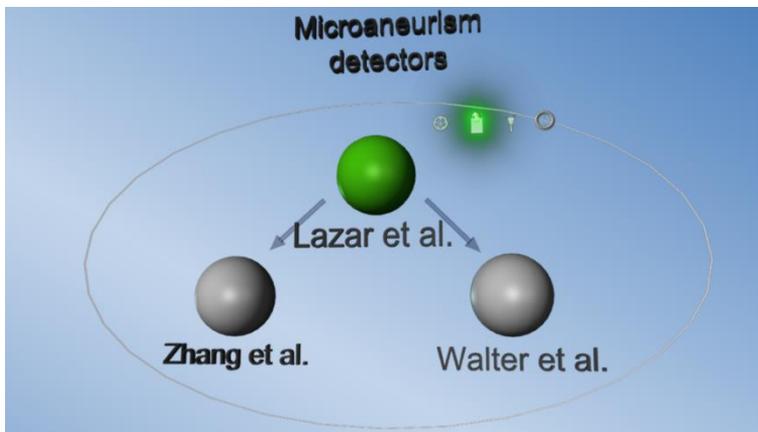


Figure 3

Selectable components for ensemble creation

Our system also contains algorithms that provide functionalities that can be composed in order to obtain a new component with different or improved functionality. For example, the fusion of an image preprocessor and a microaneurysm detector algorithm together can form an improved microaneurysm detector component.

In our visual representation, algorithm fusion can be performed in a similar way as ensemble creation. The fusible pairs $(C_{i,j}, C_{i',j'})$ consisting of the selected component and specific components in other functionality groups are visualized with connecting arrows; however, the user can select only one of these components at once (see Fig. 4).

The result of the algorithm fusion is a new component FUS_i that is represented with different color and the icon of algorithm fusion (see Fig. 5).

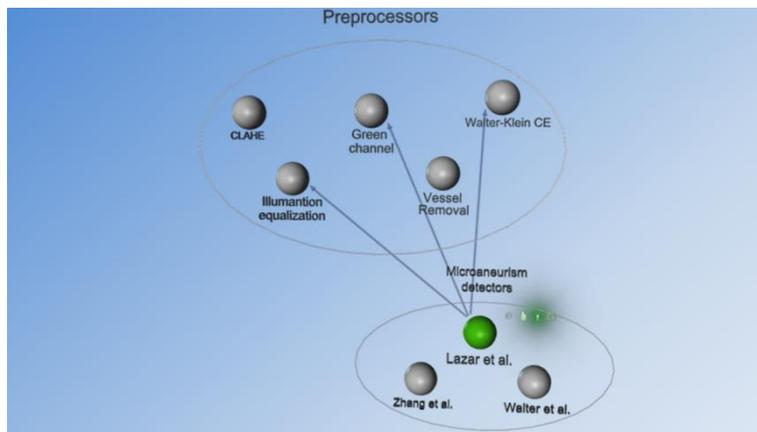


Figure 4

Visualization of the fusible pairs between a functionality group and a component

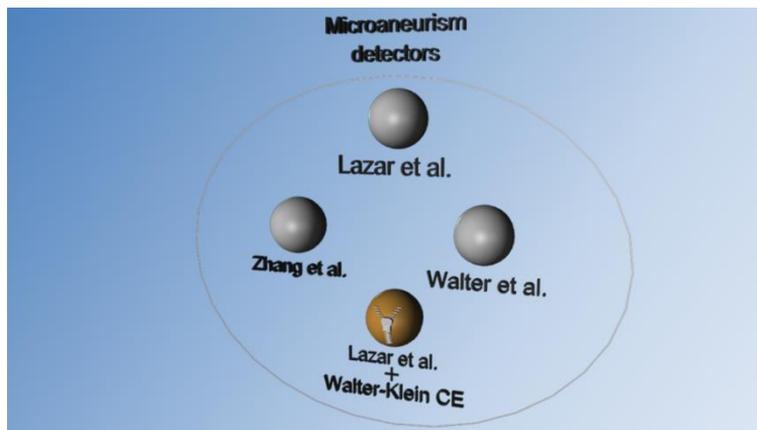


Figure 5

Result of the algorithm fusion: a new component is created

4.1.4 Databases and Energy Functions

The different databases DB_i and error (energy) functions EF_i are involved in application generation. In the case of automatic application generation, the aim is to compile a system that is optimal regarding a given energy function on the selected database(s) without user intervention. In our system, we consider two energy functions: optimization for accuracy and optimization for computational time. These energy functions are represented by icons that refer to the target of optimization (see Fig. 6).

In case of manual application generation, databases are used to evaluate the performance of the system constructed by the user, and to obtain information

about the accuracy of different components, in order to assist the selection of the best ones fitting the requirements of the user.



Figure 6

Icons for the energy functions in our system

Databases are represented by the usual database icon in our visualization (see Fig. 7). The user can also display information about a database and statistics about its content using its component icon menu.

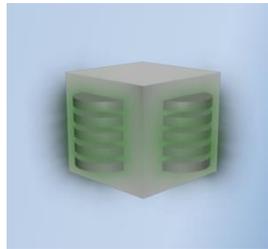


Figure 7

Database icon in selected state

4.2 Metadata Description

We defined an XML schema to be able to describe the elements of our decision support system in a uniform way. It is practical to provide descriptions in this manner, as this schema can also be considered as an easily extendable communication interface between the visualization platform and the application server of the system for the implementation.

Next, we briefly present the main sections of the XML schema. Namely, these are:

- Algorithms
 - Detectors (for functionalities F_4 , F_5 , F_6 and F_7 (see Table 2))
 - Preprocessors (for functionalities F_1 , F_2 , and F_3)
- Energy functions
- Databases
- Definition of the applicable voting models for ensemble creation.

In the schema, each component has a globally unique identifier attribute for the ease of reference. Each preprocessor is defined with the *Algorithm* complex type, which describes a general individual algorithm, having the following attributes:

state, name, and algorithm description, and an arbitrary number of controlling parameters required for the algorithm.

The detectors are defined by the *DetectorAlgorithm* complex type that extends *Algorithm* with a set of accuracy attributes that are the measured accuracies of the given algorithm on different databases. The metadata description of the *Algorithms* section of the schema is shown on Fig. 8.

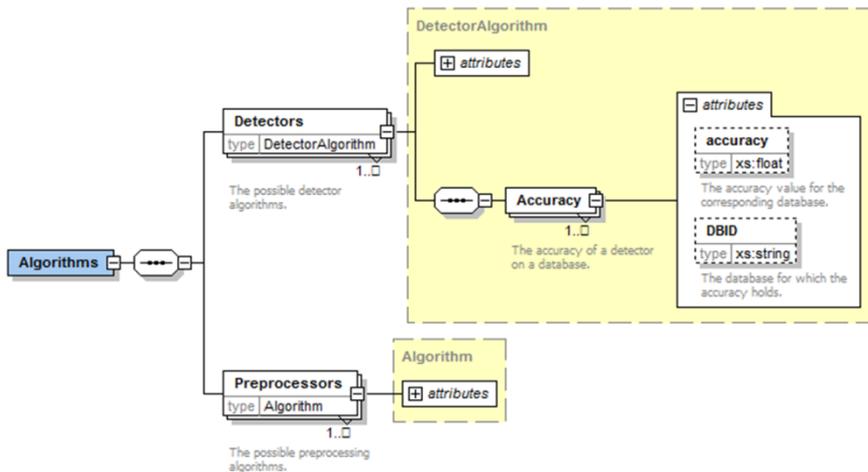


Figure 8

Metadata description for the algorithms/components of the system

In this XML schema, databases are defined to have an attribute that describes their content, and energy functions are defined to have attributes describing formally the target of optimization. The applicable voting models are defined by the attributes of the possible voting schemes and the formal description of the method used for combining the outputs of the members.

5 Implementation in the VirCA System

We have studied various 3D graphical systems, VRML worlds and other frameworks for the visual representation of our general formal description for ensemble-based systems. However, most of them have limitations in the number of the parallel handled users, in the interaction capabilities, or in the level of collaboration.

We have found that the high-level requirements that are needed for the realization of such a complex system can be fulfilled only by a visual collaboration platform having even a physical simulation subsystem. For this reason, we can recommend VirCA as a good environment for implementing these compound graphical user

interfaces. VirCA is a highly customizable 3D collaboration framework [22, 23] that is able to handle several users and their interactions with the objects in the visualized scene in real time [24]. It has a versatile viewpoint system having freely portable predefined cameras with the capability of zooming to interact with the visual elements that can be manipulated using a multilevel command system. Furthermore, the network communication is implemented using the ZeroC ICE (Internet Communications Engine) [25] object-oriented platform, through which the visualization interface of the system can interact with the underlying application server.

5.1 Visual Elements and Handling

The described visual representation can be accomplished in VirCA by using spatial elements as spheres, ellipsoids, cubes, cones, pyramids, etc. to represent components that belong to the different functionalities. The control elements (e.g. sliders, spinners, etc.) of the attribute panel and the description box of a component can be implemented using the platform independent Qt [26] widgets.

To interact with the elements of the interface, we can use for example traditional or spatial mouse, camera, motion, eye-gaze, and hand gesture sensors, or even a Microsoft Kinect game controller. Each interaction gives a clear visual feedback, and audio feedback also can be set up through the text-to-speech function of the system. The physical subsystem of VirCA is able to handle only a few thousand elements. However, this is not a limitation for our purposes since the number of visual elements required for the visualization of the systems we consider is expected to be much lower than this limit.

5.2 Performance and Implementation Issues

Techniques to dynamically create and modify objects in real time in contrast of using statically created and stored models and objects can make the implementation of the visual interface more efficient. Currently, even a simple color change in the visualization needs the same model to be stored in multiple instances according to the number of colors used. In the forthcoming versions of VirCA, it will be possible to generate objects dynamically using OpenGL function calls. Using dynamic generation, the meshes and materials do not have to be stored in files in advance, but they can be created or modified during operation. In this way, when it comes to the modification of an object, calling a delete and construct procedure pair is not necessary, which yields to performance gain. By building the complex objects programmatically, it is easier to create joint points and make a skeleton to move as required, thus the animation of the model will be more natural by this approach.

Conclusions

In this paper, we have described a 3D visualization framework that assists the comprehension of and interaction with an ensemble-based system by emphasizing the human factors and natural communication in its design. We have given a general formal description for all the elements and operations that can be used to compose such systems. As a case study, we have considered an ensemble-based decision support system for diabetic retinopathy screening to assign the concepts of the visualization framework to a real-world application. Accordingly, we have introduced visualization techniques to facilitate interpretability of the system components and functionalities and the reasoning about their relations. The framework we have proposed supports the better understanding of the applications automatically generated by the system, and allows the users to modify these ensembles or to create new ones that fit better their requirements. In this way, the decision making process of the system visualized can evolve based on the blending of human and artificial cognitive capabilities. Furthermore, implementing the proposed visualization in a 3D collaborative framework like VirCA allows multiple users to simultaneously explore, gain knowledge of [27] and modify the ensemble setups, which promotes both intra-cognitive and inter-cognitive information transfer [11] about the decision process of such a system.

Acknowledgement

This work was supported in part by the projects EFOP-3.6.2-16-2017-00015 and VKSZ 14-1-2015-0072, SCOPIA: Development of diagnostic tools based on endoscope technology supported by the European Union, co-financed by the European Social Fund.

References

- [1] R. Polikar: Ensemble based systems in decision making, IEEE Circuits and Systems Magazine, Vol. 6, No. 3, pp. 21-45, 2006
- [2] C. Zhang and Y. Ma (eds.): Ensemble Machine Learning: Methods and Applications, Springer-Verlag New York, 2012
- [3] H. B. Mitchell: Image Fusion: Theories, Techniques and Applications, Springer-Verlag Berlin Heidelberg, 2010
- [4] DRSCREEN - Diabetic Retinopathy Screening Project. [Online, accessed 2018-05-25] <http://drscreen.eu/>
- [5] B. Antal, I. Lazar and A. Hajdu: An Ensemble Approach to Improve Microaneurysm Candidate Extraction, ICETE 2010, Communications in Computer and Information Science, Vol. 222, pp. 378-394, Springer-Verlag Berlin Heidelberg, 2011
- [6] B. Harangi, R. J. Qureshi, A. Csutak, T. Peto and A. Hajdu: Automatic detection of the optic disc using majority voting in a collection of optic disc

- detectors, 7th IEEE International Symposium on Biomedical Imaging (ISBI2010), pp. 1329-1332, April 14-17, 2010, Rotterdam, The Netherlands
- [7] D. Moody and J. Hillegersberg: Evaluating the Visual Syntax of UML: An Analysis of the Cognitive Effectiveness of the UML Family of Diagrams, Software Language Engineering, Lecture Notes in Computer Science, Vol. 5452, pp. 16-34, Springer-Verlag Berlin Heidelberg, 2009
- [8] A. Torok: From human-computer interaction to cognitive infocommunications: a cognitive science perspective, 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2016), pp. 433-438, October 16-18, 2016, Wroclaw, Poland
- [9] H. Wright: Introduction to Scientific Visualization, Springer-Verlag London, 2007
- [10] A. Hajdu, J. Toth, Z. Pistar, B. Domokos and Zs. Torok: An ensemble-based collaborative framework to support customized user needs, 3rd IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2012), pp. 285-290, December 2-5, 2012, Kosice, Slovakia
- [11] P. Baranyi and A. Csapo: Definition and Synergies of Cognitive Infocommunications, Acta Polytechnica Hungarica, Vol. 9, No. 1, 2012
- [12] P. Baranyi, A. Csapo and G. Sallai: Cognitive Infocommunications (CogInfoCom), Springer International Publishing, 2015
- [13] L. I. Komlosi and P. Waldbuesser: The cognitive entity generation: Emergent properties in social cognition, 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2015), pp. 439-442, October 19-21, 2015, Győr, Hungary
- [14] P. Galambos, I. M. Fulop and P. Baranyi: Virtual Collaboration Arena, Platform for Research, Development and Education, Acta Technica Jaurinensis, Vol. 4, No. 1, pp. 145-155, 2011
- [15] I. M. Fulop: Semantic services in the VirCA system, 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), pp. 331-334, May 19-21, 2011, Timisoara, Romania
- [16] Confer *et al.*: Evolutionary Psychology: Controversies, Questions, Prospects, and Limitations, American Psychologist, Vol. 65, No. 2, pp. 110-126, 2010
- [17] M. G. Haselton, G. A. Bryant, A. Wilke, D. A. Frederick, A. Galperin, W. E. Frankenhuis and T. Moore: Adaptive Rationality: An Evolutionary Perspective on Cognitive Bias, Social Cognition, Vol. 27, No. 5, pp. 733-763, 2009
- [18] D. Chang, L. Dooley, J. E. Tuovinen: Gestalt Theory in Visual Screen Design – A New Look at an Old Subject, Computers in Education 2001: Australian Topics - Selected Papers from the Seventh World Conference on

- Computers in Education (WCCE2001 Australian Topics), CRPIT, Vol. 8., ACS, Copenhagen, Denmark, 2002
- [19] T. M. Green, W. Ribarsky and B. Fisher: Visual Analytics for Complex Concepts Using a Human Cognition Model, IEEE Symposium on Visual Analytics Science and Technology (VAST '08), October 19-24, 2008, Columbus, OH, USA
- [20] E. Hutchins, J. Hollan and D. Norman: Direct Manipulation Interfaces, Human Computer Interaction, Vol. 1, No. 4, pp. 331-338, 1985
- [21] S. Thorpe, D. Fize and C. Marlot: Speed of Processing in the Human Visual System, Nature, Vol. 381, pp. 520-522, 1996
- [22] P. Galambos, C. Weidig, P. Baranyi, J. C. Aurich, B. Hamann and O. Kreylos: VirCA NET: A Case Study for Collaboration in Shared Virtual Space, 3rd IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2012), pp. 273-277, December 2-5, 2012, Kosice, Slovakia
- [23] P. Galambos, A. Csapo, P. Zentay, I. M. Fulop, T. Haidegger, P. Baranyi and I. J. Rudas: Design, programming and orchestration of heterogeneous manufacturing systems through VR-powered remote collaboration, Robotics and Computer-Integrated Manufacturing, Vol. 33, pp. 68-77, 2015
- [24] P. Galambos and P. Baranyi: VirCA as Virtual Intelligent Space for RT-Middleware, 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 140-145, July 3-7, 2011, Budapest, Hungary
- [25] ZeroC ICE: Comprehensive RPC Framework. [Online, accessed: 2018-05-25], <https://zeroc.com/>
- [26] Qt: Cross-platform software development for embedded & desktop. [Online, accessed: 2018-05-25], <https://qt.io/>
- [27] I. Horvath: Innovative engineering education in the cooperative VR environment, 7th IEEE Conference on Cognitive Infocommunications (CogInfoCom 2016), pp. 359-364, October 16-18, 2016, Wroclaw, Poland