# Versatile Program Model of Avatar Face

Milan Adamović, Dragan Ivetić[1], Aleksandar Kupusinac[2]

*Abstract* – **Human facial models can be abstracted and coded in electronic form that is suitable for a variety of purposes. This paper describes the process of developing a program model of the human face. The model is based on a parameterized curves and topological guards. Special attention is given to the way the human mind perceives faces, as well as differences between genders, ethnic and age groups. The produced model is then implemented and tested in a facial composite modeling application used at our faculty.**

*Keywords* – **avatar, face, program model, facial composite.**

## I. INTRODUCTION

Avatars have become commonplace in video games and on the Internet [1]. Use of avatars includes, but is not limited to identification, or, on the contrary, protection of users' identity in virtual social networks. Avatars can be made in different ways, in different styles (ranging from stylized to realistic), can represent a person's entire body, or only their face, and can be 2D or 3D. Avatars can be composed from templates and premade elements, or drawn by hand. These characteristics depend on the avatar's domain of application.

A versatile program model of the human facewas developed by students of Animation in Engineering for the purpose of coding practice. Most existing program models of avatar face use simple premade shapes as building elements, which severely limits the number of available combinations. This approach ensures consistency between the avatar and other graphical interface elements in an application. While hand drawn avatars do give the final user more creative freedom, they also require certain skills to create, and may clash with the rest of the interface. The model described in this paper is vector based, and avoids the topological guard imposed by a small number of available premade shapes, but also contains safety mechanisms to prevent deformations in the final product and not burden the user with needless information.

The paper is structured as follows: In section one we introduce the subject of avatars and look over some existing areas of application, then in section two, we describe perceptual characteristics of human facial features, by the brain and the ways in which it differs from computer modeling, then we move on with abstraction of those facts into a form more suitable for a program model as well as the differences between it and , which we follow up with a example of the model's implementation into a facial composite application in section four, and finally conclude the paper in section 5 with an evaluation of the models usability.

[1] Dragan Iveticis with the Faculty of Technical Sciences at the University of Novi Sad, Trg Dositeja Obradovica 6, Novi Sad 21000, Serbia, E-mail: ivetic@uns.ac.rs

[2] Aleksandar Kupusinac is with the Faculty of Technical Sciences at the University of Novi Sad, Ohridski Blvd, Sofia 1000, Bulgaria.

## II. PERCEPTUAL CHARACTERISTICS OF FACES

### A. Process of facial perception

The intuitive facial processing happens in the right hemisphere of the brain, while the conscious processing happens in the left [2]. First, the right hemisphere roughly evaluates the proportions of facial features, and compares them to faces the observer knows. Afterwards, the left hemisphere processes individual details, such as hair and eye color, scars etc. The mole on the face of Marilyn Monroe is a distinguishing feature, and it's presence or absence in a depiction on Fig. 1 determines whether the person is supposed to be Marilyn Monroe, or just any blond blue eyed woman [3].

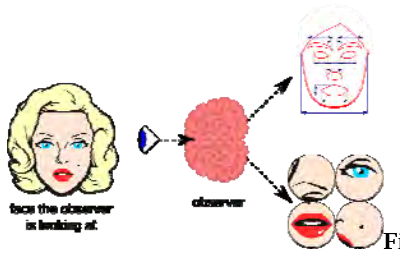



**Fig 1.** The face of Marilyn Monroe perceived by the brain hemispheres.

The facial recognition mechanism of the human brain is effective during tasks that include recognizing persons, but is not helpful when it comes to painting, modeling and similar tasks.. Certain individuals are suffering from a condition called prosopagnosia [4], that hinders the intuitive facial perception. Such persons can be trained to get their left brain hemisphere to completely take over the process of facial recognition. Training for persons with prosopagnosia involves observing and analyzing individual facial features and remembering certain parameters (e.g. typical chin shapes: square, round, spiked...) characteristic for certain groups of people (genders, races, age groups, etc. For example square shaped chins are more common on men than women) Persons who are trained to consciously analyze faces, but don't suffer from prosopagnosia achieve more accurate results when modeling human faces [8].

Such analytical approach to facial recognition makes it possible to consciously perceive and describe individual facial features, making them easier to model. The characteristics described in section 2.B describe facial features in conversational language, while section 3 deals with their abstraction into a program model.

*B. Genders, races and age groups*

An important factor that contributes to easier differentiation between genders, races and age groups is neoteny [5], which is the ability of living creatures that determines what amount of youthful features adult individuals get to keep (Fig. 2, far left). Species with higher degree of neoteny tend to look younger and more feminine, while species with little to no neotenous features tend to look rugged and threatening. Most people fall into one of three basic ethnic groups (Fig. 2, center and right). Human beings are very neotenous compared to other species. Persons with neither neotenous, or non-neotenous faces are called androgynous. Gender of androgynous persons is difficult to tell from just quickly glancing at their faces.

Fig 2. Genders (far left, no color fill), three biggest ethnic groups depicted using the program model (center is Asian, followed by Caucasian and African), as well as signs of aging (far right) also depicted using the program model

Wrinkles can be easily added to the model as simple curves, as well as gray hair, while the droop of the facial muscles can be added with a topological guard (Fig. 2, far right).

he shapes described in conversational language in this chapter are abstracted and coded into a program model described in chapter 3. Differences (e.g. a round chin versus an angular chin) are broken up into sets of individual that affect certain topological areas of the face (e.g. chin width, chin height, etc.) and given descriptive names to ease the implementation for first time users.

## III. FACIAL FEATURES PARAMETRIZATION

The more control the user is given over the model, the more complex the model will have to be. Achieving a usable compromise between these two characteristics in a program model is the end goal [6]. Data in the model can be sorted into two categories:

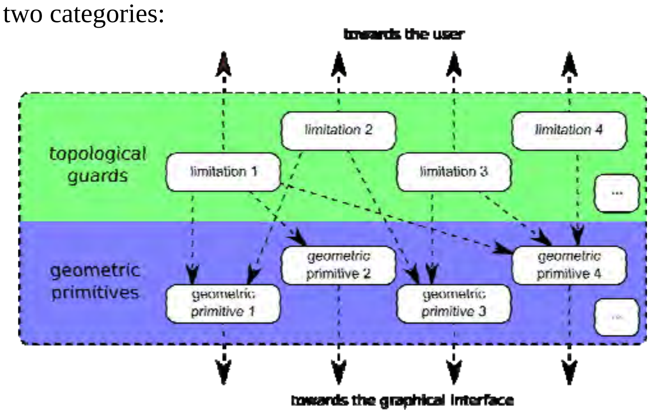
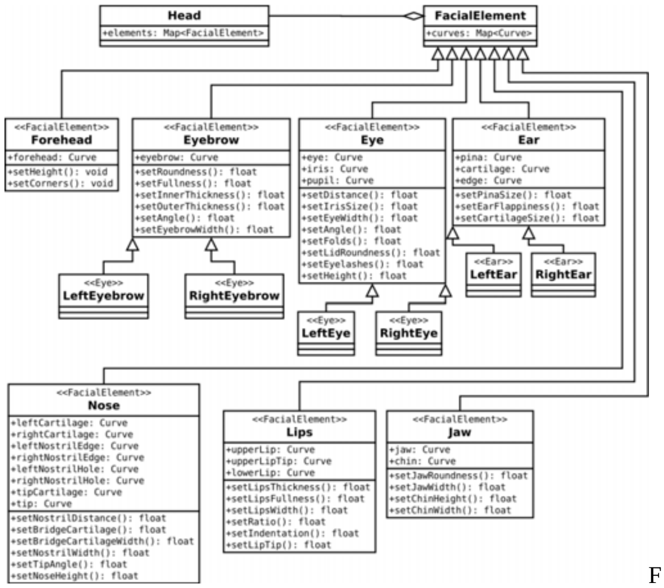



Fig 3. Hierarchy of data in the model

1. Geometric primitives that are used to model the face (2D vertex, lines, polygons, etc.) One of the goals of the model is to make manual manipulation over the geometric primitives as low as possible.

2. Topological guards that provide a mechanism for the controlled composition of avatar's face. Topological guards themselves are decimal numbers normalized to a range from 0 to 1. Topological guards make the model robust by preventing input of wrong parameters that can cause deformation of avatars' faces [7].

The graphical primitives in the model are visualized by Bezier's curves (figure 5). Green points are endpoints, while dark purple crosses and bright orange triangles are control




F
ig. 4 .Class diagram of the facial elements hierarchy

points. Curves themselves are encapsulated in classes that represent individual facial features.Common parameters are position, rotation and scale.

Facial elements the model describes are Forehead, Eyebrows, Eyes, Ears, Nose, Lips, Jaw, Wrinkles, Scars, Scalp hair and Facial hair. Despite the fact that accessories (glasses, jewelry, etc.) are not a part of the face, the model includes them, as well.

The basis of the model is shown on figure 3. All facial elements inherit FacialElement that contains a collection of curves. Using a collection of curves rather than individual curves is to ensure that certain elements contain several separate lines (e.g. nostrils).

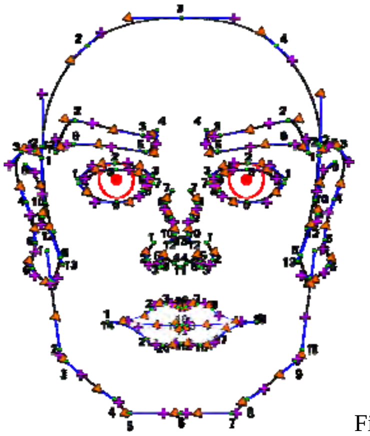


Fig 5. Face shown as a collection of Bezier curves.

Graphically, the face depicted through the use of Bezier curves (figure 4). The figure shows the face without the use of fill colors to keep focus on the shapes. Elements such as hair and accessories are not included here to avoid overcrowding. They were not coded using topological guards, but as collage elements that can be swapped in and out and replaced as needed.

Common parameters (topological guards) for all elements are: position, rotation and scale. More specific parameters can be derived from the basic three, but they only affect specific groups of nodes (Fig 6). Position, angle and scale of nodes can be affected using different mathematical equations that depend on topological guards. Figure 6. shows the process of a curve alteration when the value of a topological guard is changed from 0 to 1 (all topological guards are normalized to 0-1 range). The dark line shows the look of the curve when the limitation is set to 0, while the lightest line shows the curve when the limitation value is set to 1.

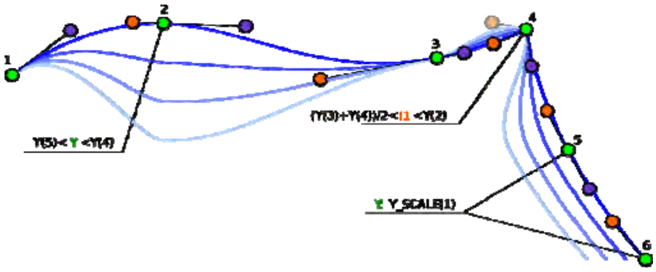



Fig 6. Example of a topological guard affecting a curve.

Colored letters can be green (nodes), orange or purple (control points). The formulas show the way the limitation affects the nodes and control points.Each topological guard represents a certain editable aspect of a facial feature. For example, adding a topological guard “chin width” would affect nodes 4, 5, 6, and 8 of the chin curse, as well as control points of nodes 3 and 9. At value zero, the chin will be the narrowest, and the nodes will be tightly packed together. As the value starts changing, the nodes will disperse horizontally, and the chin will widen. When the value of “chin width” reaches one, the chin will be at it’s widest. The highest and lowest possible values of a topological guard depends on a multitude of factors, including other topological guards. Chin width would certainly depends on overall head width, and possibly on jaw width, depending on the rest of the parameters. When a topological guards changes value, values of other topological guards have to be recalculated to make sure they fit the 0 to 1 range when the change is applied.

Each topological guard represents a certain editable aspect of a facial feature. For example, adding a topological guard “chin width” would affect nodes 4, 5, 6, and 8 of the chin curse, as well as control points of nodes 3 and 9. At value zero, the chin will be the narrowest, and the nodes will be tightly packed together. As the value starts changing, the nodes will disperse horizontally, and the chin will widen. When the value of “chin width” reaches one, the chin will be at it’s widest. The highest and lowest possible values of a topological guard depends on a multitude of factors, including other topological guards. Chin width would certainly depends on overall head width, and possibly on jaw width, depending on the rest of the parameters. When a topological guards changes value, values of other topological guards have to be recalculated to make sure they fit the 0 to 1 range when the change is applied.

Topological guards can affect groups of other topological guards. For example, facial characteristics that are typical of men and women can be roughly quantified into a single parameter. Topological guard such as a chin width, jawline angle, eye size, eyebrows shape etc. can be connected to a single topological guard called “gender”. Individual features of a person can not be accurately modeled using only one parameter. It’s primary purpose would be to get the rough shape of a person’s face, and then fine tune the features using more specific topological guards.

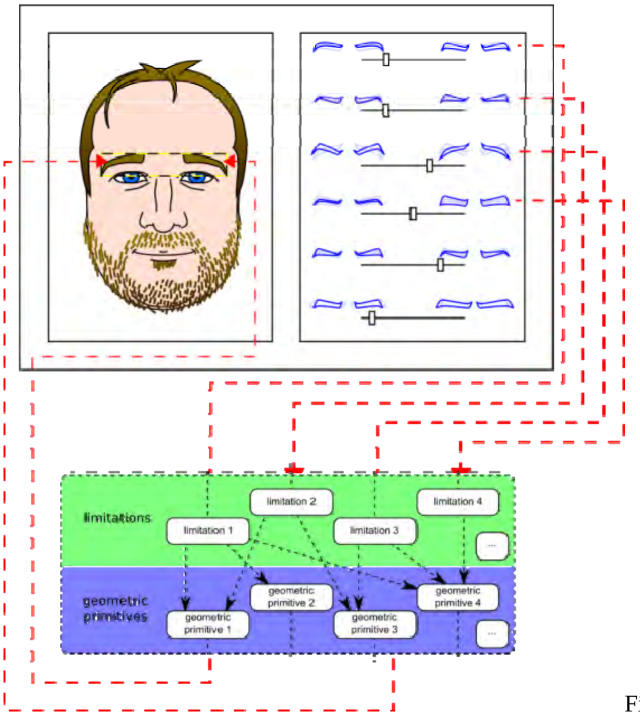
The guards used in the model were made using data for treatment of people with prosopagnosia, as that data readily described typical shapes of facial features in a form easily translatable into a program model.

Different geometric primitives can be used to quickly adjust the appearance of an avatar to the environment it could be used in, ranging from a realistic 3d textured environment, to an exaggerated cartoon environment. The look used in this paper is functionally realistic, as it retains realistic proportions, while also using bright cartoon-like outlines and coloring to make the shapes easier to distinguish at a low LOD.

## IV. AN EXAMPLE OF FACIAL AVATAR PROGRAM MODEL USE

Hand made sketches of faces that faithfully represent suspects require years to train specialized staff, while training regular staff to use software specialized for modeling facial composites takes hours. While some automated facial composite applications do exist, they mostly rely on raster picture processing and achieve usable results about 40% to 60% of the time [1]. Figure 5 shows a mockup of facial composite software that implements the vector program model described in this paper.

As values are adjusted, the changes are calculated and displayed in real time.




Fi g 7. Implementation of the model in facial composite. The mockup interface shows eyebrow topological guards.

Using the avatar facial model described here would also make searching a database of existing faces far easier. Faces would be parametrized far more precisely than they are when bitmap analysis is used, but a new database would also have to be created, first.CONCLUSION

## V. CONCLUSION

The use of procedural generation to generate graphical representations of facial features provides a practically infinite combination of different shapes, with a significant reduction of memory use, demonstrated on the facial composite application. Lip animation using the same basis as the one in model would demonstrate that it is applicable in areas it was not originally meant for. The model is expandable and can be made to support modeling of non human, but still humanoid models, as well as special cases of unusual facial structures, such as not having certain facial features. The model is promising, and could be developed to serve more purposes, rather than just having animation students practice their coding on it.

## REFERENCES

[1] [1] Chi-Hyoung Rhee and Chang Ha Lee, “Cartoon-like Avatar Generation Using Facial Component Matching”, School of Computer Science and Engineering, Seoul, South Korea

[2] [2] E. De Renzia, P. Faglioni, H. Spinnler, The Performance of Patients with Unilateral Brain Damage on Face Recognition Tasks, 1st ed. 1968, pp. 17-34.

[3] [3] Vicki Bruce, Andy Young, Understanding face recognition, 3rd ed., Vol. 77, 1986,

[4] [4] Dr Sarah Bate, „Information about prosopagnosia“, http://prosopagnosiaresearch.org/index/information

[5] [5] Science Daily team, “Neoteny”, http://www.sciencedaily.com/articles/n/neoteny.htm

[6] [6] Norman H. Mackworth, Anthony J. Morand, The gaze selects informative details within pictures, 1st ed, Harvard Publications, 1967, pp. 547-550

[7] [7] “Make Human Documentation”,. http://www.makehuman.org/documentation

[8] [8] Vicki Bruce, Andy Young, Understanding face recognition, 3rd ed., Vol. 77, 1986, pp. 305-327.