Elastic Bone Transformation for Realistic Facial Animation in WebGL

Andoni Mujika, Nagore Barrena, Sara García, David Oyarzun

Vicomtech-ik4, Mikeletegi Pasealekua, 57, Donostia – San Sebastian, 20009, Spain

Abstract

This chapter describes the mathematical model that will be used to animate a virtual face in a project called SPEEP, a project that makes use of this virtual face to teach the pronunciation of foreign languages to students. The mathematical model is based on the well-known Skeleton Subspace Deformation, but an elastic layer is inserted in the generation of bone's transformations. Besides, the whole process that will be followed in the project is described, from the definition of the skeleton structure and the training of the parameters of the model to its application in a WebGL environment.

1 Introduction

The utilization of avatars, 3D virtual characters, in films like Toy Story or Shrek is widely widespread and it is the same for computer and console games. That is why the animation of 3D characters is a very mature technology. Specifically, the facial animation of virtual characters is increasingly realistic due to its importance in 3D applications, especially in those where the avatar talks to the user. Indeed, a study made by Hodgins et al. [1] proved that an audiovisual content with virtual characters looses emotional value if the facial animation is not of high quality.

Although historically the intervention of modelers has been needed for obtaining realistic results, nowadays their work could be restricted to a final stage, since actual facial motion capture systems and new 3D engines give a considerably realistic animation. The modelers only have to correct the errors of the system. Nevertheless, in real-time applications modelers cannot touch the animation and realism must be obtained directly from the capture system. In the case of Visual Speech Synthesis, where the movements and the voice of the character are synthesized from plain text, the motion capture and the corrections must be done before the application is run. Then, the animation is generated automatically in real-time.

In SPEEP, the project for foreign language pronunciation learning partly funded by Basque Government we are presenting in this chapter, both real-time problems described above have to be solved, since the users pronunciation will be reproduced in a virtual face and the correct pronunciation will be rendered from plain text.

Besides, the application will be run in a web browser. For that, WebGL has been selected to render the virtual model via web. WebGL [2] is a powerful Application Programming Interface (API) to present 3D graphics in a web page. It is based on OpenGL, which is a very widely used open source 3D graphics standard. Moreover, WebGL is compatible with different and most common browsers such as, Google Chrome, Mozilla Firefox, or Safari. So, WebGL allows using web technologies, which are an easy way to access to contents for non expert users.

In the following, we make a short review of the state of the art in the second section; we present the modules of the facial animation system and the mathematical model used for the facial animation in the next two sections, to finally conclude the chapter in the fifth section.

2 Related Work

Radovan and Pretorius [3] classify the different methods for facial animation in three groups: based in geometry, based in images and based in real movements.

In the first group, we can find the most primitive works, where transformations of the vertices in the facial mesh are computed one by one. The most relevant method in this group is Parke's [4], since it is considered the first work in facial animation. On the other hand, more recent techniques can also be classified in geometry-based animation. Works that make use of muscles or pseudo-muscles [5] or directly simulate the physical behavior of the muscles and the skin [6] exist nowadays.

Other methods, usually focused in the film industry, are based in images instead of geometry: morphing techniques [7], where different key faces are interpolated to obtain the animation; texture manipulation methods [8], where the changes in the texture of the facial mesh create the animation and works based on blend-shape interpolation [9], where the intermediate frames between two modeled faces are computed.

Nevertheless, recently performance-driven techniques have replaced others because of their realism. Lots of works that capture the movements of an actor and translate them to a virtual face can be found. Beeler et al. [10] analyze the captured data to recognize predefined motions and launch these predefined movements in the virtual face. In the work presented by Arghinenti [11] the captured data is only a layer of the facial animation engine. Other layers cope with expressions, phonemes and muscles. Deng et al. [12] combine a motion capture system with a face scanning system in order to obtain a realistic animation with low-level details, such as wrinkles. Once they obtain the minimum number of key expressions, blend-shape interpolation is used to generate the final animation.

However, as stated before, the motion capture system cannot be used always. And in these cases, another type of facial animation has emerged recently, skeleton-driven facial animation [13-15]. As in corporal animation, the face of the virtual character has a structure of joints and bones attached. The vertices of the facial mesh are transformed in concordance with the skeleton, since they are associated to certain bones. The project SPEEP can be classified in this group.

Unlike in corporal animation, the skeletons differ widely in skeleton-driven facial animation methods. Some have elastic bones, others have rigid bones; some use segments as bones, others use curves as bones; some have tree structure with its node in the neck, others have disjointed groups of bones.

Apart from the facial animation technique, in the project SPEEP we have to decide how the virtual face will pronounce the needed sentences. For example, when pronouncing the phonemes *p* and *a* in different order (*ap* and *pa*) the behavior of the human mouth is not the same, i.e. the "interpolation" between *a* and *p*, and *p* and *a* is not the same. This is due to the so called co-articulation.

One of the first works in this field and one of the main references is the one presented by Cohen and Massaro [16]. They create exponential functions for facial parameters that rise until the phoneme's exact time and fall afterwards. The coarticulation is obtainted by the combination of different phonemes and the functions for their associated facial parameters.

Since the first Works in co-articulation were published, several methods have been presented that take speech unities and convert them in a fluent and realistic facial animation: rule-based, Markov models [17], neural networks [18]…

To finish this review of the state of the art, it is important to say that not many works in facial animation for WebGL can be found. The work by Benin et al. [19] is an implementation of a WebGL talking head that works with MPEG-4 Facial Animation Parameters (FAPs) standard. The human facial movements are generated by different lips models that are built using 3d kinematics dates. This work is only developed for Apple iOS mobile devices.

There are some other methods for web-based MPEG-4 facial animation that can be more appropriate like FATs [20]. The Face Animation Tables (FAT) allows the precise specification of the movements of the vertices assigned to each FAP.

In conclusion, there exist ingredients for a realistic facial animation, but very few have been used with the emerging WebGL technologies. Our work goes in this direction.

3 Animation Pipeline

The application developed in the project SPEEP will be used for foreign language pronunciation teaching. On the one hand, the images and the speech of the user will be used for rendering his pronunciation in a virtual character. On the other hand, Visual Speech Synthesis, that is a virtual character pronouncing correctly the text, will be used to correct the errors of the user.

The key feature of the project is the utilization of web browser for the rendering of the virtual face. To generate animations in any device, the object has to be drawn many times in a second, drawing it in a different position each time. With this goal, the position of the model should be calculated every time. This means that the animation rate dependents on the speed of the rendering cycle. A slower rendering would produce choppy animations and a too fast rendering would create the illusion of objects jumping from one side to the other. So the renderer time is crucial to generate a good animation [21].

For this reason the computational cost when the object is rendering in a browser is an important point in our project. WebGL technology is slower than native OpenGL because it uses JavaScript for execution. But it is still nearly seven times faster than Flash owing to GPU acceleration. For this reason this project is developed using this technology.

In spite of the good computational cost of WebGL in general, this kind of methods for facial animation can be slower than expected. The speed of the construction of the FATs [20] for example has to be improved to achieve better results.

Thus, our work has been focused on optimizing each step of the facial animation engine, in order to obtain an efficient system. As shown in the figure 1, the facial animation system of the project SPEEP is divided in four different parts: the Viseme/Facial Animation Parameter (FAP) Converter, the Co-articulation Module, the FAP/Bone Converter and the Facial Engine. Each part has been or will be studied to reduce the computational cost.

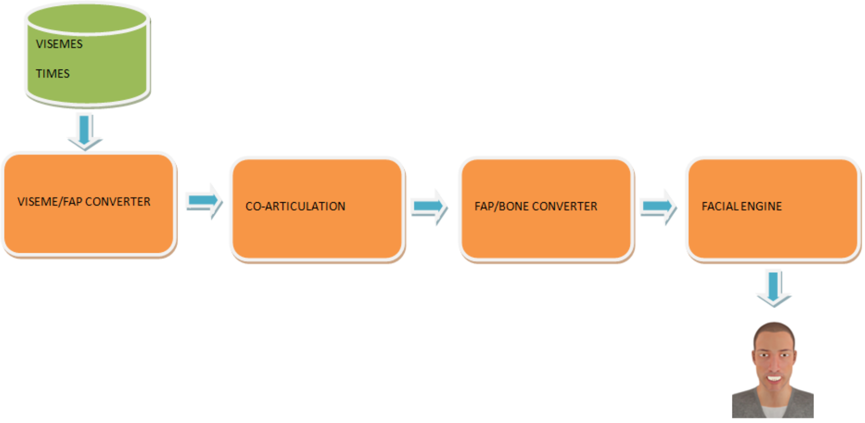

Fig. 1. Overview of the SPEEP system.

3.1 Viseme/FAP Converter

On the one hand, a viseme is a generic facial image that can be used to describe a particular sound, i.e. it is the equivalent of a phoneme in the field of facial positions. For example, the equivalent of the phoneme *a* is the face that pronounces the phoneme. On the other hand, Facial Animation Parameters (FAPs) were presented in the standard MPEG-4 [22] to define the reproduction of emotions, expressions and speech pronunciation.

The system of the project receives a set of ordered visemes that have to be reproduced and the exact times when they have to be reproduced. So, the first step in the system will be the conversion of these visemes to values of FAPs. The correlation between visemes and FAPs is defined and stored before the real-time performance and when rendering the virtual face, the system only takes the needed values from the stored data and weights them with the input. For example, if the converter receives the viseme *a* with the weight 0.6, the output will be the stored corresponding FAP values weighted with 0.6.

3.2 Co-articulation Module

The second module of the system receives the values of the FAPs at the moment when the visemes have to be rendered and has to compute the values at any moment. Nowadays, the model of co-articulation is based on Cohen and Massaro's [16] work. Nevertheless, since it is a key aspect for the realism, this is a clear field for our future work.

3.3. FAP/Bone Converter

Once we have the values of FAPs at any time, they must be translated to the facial engine, i.e. they have to be translated to the skeleton that moves the virtual face. A rule-system method is used to convert the FAPs into transformations of the corresponding bones.

For that, an skeleton based on the one defined in the MPEG-4 standard [22] has been defined. In our case, as the system has to work in WebGL, the number of bones has been lowered in order to overcome system's low power. Based on the work by Contreras [23], some sets of bones defined in the standard were replaced by a unique bone, mainly in the lips, ears, nose and jaw. The reduction of the number of bones has been done so that the main areas for the facial animation are not too affected. In the figure 2, final bone positions can be seen.

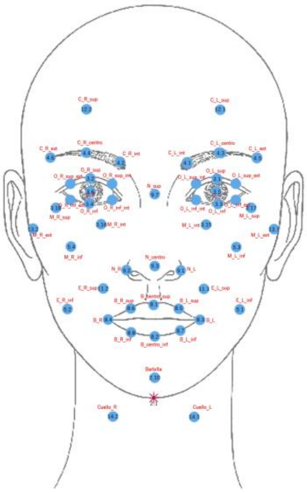

Fig. 2. Positions of the bones defined for project SPEEP.

3.4. Facial Engine

Finally, the facial engine will receive the transformations that must be applied to the bones of the skeleton and will perform the facial animation. The engine moves the skeleton and the vertices of the facial mesh are transformed accordingly. In the following section the details of the mathematical model used for the facial animation are shown.

4. Animation Model

During the animation of the virtual face of the project SPEEP, the transformations of the vertices that compound the virtual face are defined by Skeleton Subspace Deformation [24], a widely-used and efficient animation technique. The position of the vertex is determined by the transformation of the bones of a skeleton:

$$v = \sum_i w_i T_i \hat{T}_i^{-1} \hat{v} \tag{1}$$

Where v is the new position of the vertex, T_i is the transformation of the i th bone in the current pose, $\hat{T}_i^{-1}$ is the inverse of the transformation of the i th bone in the initial pose, $\hat{v}$ is the position of the vertex in the initial pose and w_i is the weight assigned for each bone. The weights w_i must fulfill the following condition:

$$\sum_i w_i = 1 \qquad (2)$$

This way, the new position of the vertex is a linear combination of the positions obtained transforming the vertex with the movements of the neighbour bones. For example, the new position of a vertex located in the biceps is obtained by transforming its position with the transformations of the shoulder and the elbow and combining them with their weights.

Nevertheless, the movements obtained by Skeleton Subspace Deformation are very rigid, i.e. they are really appropriate for corporal animations, but too rigid for facial animation. For example, when moving the mouth, only the vertices around it are moved and in order to obtain a high level of realism, it is important that the cheeks and the temples also move.

Thus, in the animation engine, before transforming the vertices, we apply an elastic layer to the bones that allows the system to animate the key part of the animation (the mouth) and automatically animates the other parts (cheeks and temples). A key characteristic of the new system must be its efficiency, since the system works in WebGL.

Therefore, first the directions of the translations that will be applied to the bone are computed, i.e. we compute the effects of the bones in the mouth will apply to the bones in the cheek:

$$u = (\hat{p}_i - \bar{p}_i) + \delta_i(\bar{p} - \bar{p}_i) \qquad (3)$$

Where u is the direction of the effect, $\hat{p}_i$ is the position after the first transformation, $\bar{p}_i$ is the initial position, i is the index of the neighbour bone and δ_i is the sign of the scalar product of the two vectors in the summation. This way, with the combination of the segment between two bones and the segment between the actual position of the first bone and its initial position, the first bone pulls or pushes the other. The sign of the scalar product is inserted in the summation in order the effect in the bone to be in the right direction, as shown in the following figure.

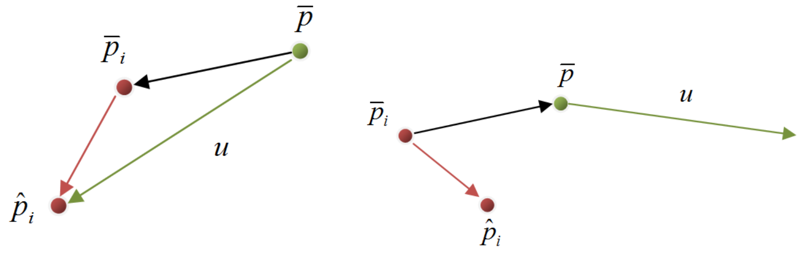

Fig. 3. In the left, the movement of the i th bone (red arrow) pulls the neighbour bone (green arrow). In the right, the movement of the i th bone (red arrow) pushes the neighbour bone (green arrow).

Once the directions of the new transformations are obtained, we sum them weighted with the distance between the initial position and the position after the regular transformation of the bones, $\|\hat{p}_i - \bar{p}_i\|$, and the weight assigned to each bone, w_i . Then, we obtain the last position of the bone, p .

$$p = \hat{p} + \sum_i w_i \frac{u}{\|u\|} \|\hat{p}_i - \bar{p}_i\| \tag{4}$$

This way, a bone is pulled by the movements of neighbour bones, but it is weighted by the assigned weight and the magnitude of the transformation of the neighbour bones. The bigger movement the bone suffers the bigger effect in the neighbour bone and the further the neighbour is the less effect will suffer.

In this moment, the project is focused on the training of this mathematical model of the animation. A facial motion capture system based on the work of Dornaika and Davoine [25] will be used to set the weights of the equation regarding the position of the bones and the predefined rules that relate the values of the FAPs and the rotations of the skeleton.

The first results are very promising, as can be seen in figure 4. As the difference between the animations with and without the elastic layer is not noticeable in one frame, the weights of the elastic layer have been increased to show the difference (note the difference in the cheeks).

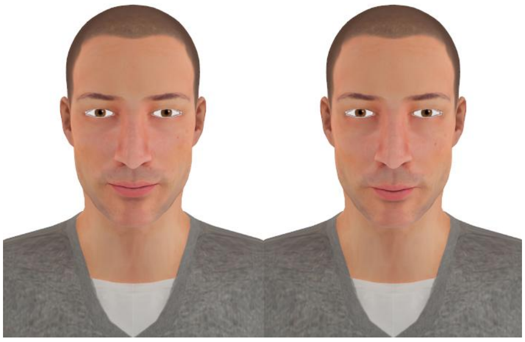

Fig. 4. In the left, a frame of an animation without the elastic layer and in the right, the same frame with the elastic layer.

5. Conclusion

We have presented the project SPEEP and its facial engine's pipeline. As the project will render a realistic virtual character in real-time by WebGL technologies, it is very important that the animation process is completely optimized.

For that purpose, we have defined a skeleton with a lower number of bones than the MPEG-4 standard without losing the main information of the movements of the face. We have also defined a mathematical model that makes use of the well-known and efficient Skeleton Subspace Deformation and adds a simple elastic layer to obtain a realistic facial animation with a low loss of efficiency in the system.

Although we are in a preliminary phase of the project, the first results have been very promising. Now, we are in the process of training the mathematical model to get more realistic model possible.

Finally, as stated before, the behaviour of the bones, the simulation of facial muscles, is very important for realism, but the co-articulation, the interpolation between visemes, is even more crucial, since it is useless to have a realistic behaviour in the face if the mouth does not move correctly. So, the main goal in the rest of the project will be the development of an efficient co-articulation algorithm that makes the virtual character realistic enough to read its lips.

References

1. Hodgins, J.; Jörg, S.; O'Sullivan, C.; Park, S. I. & Mahler, M. The saliency of anomalies in animated human characters. ACM Trans. Appl. Percept., ACM, 2010, 7, 22:1-22:14
2. Leung, C., Salga, A., 2010. Enabling WebGL. Proceedings of the 19th international conference on World wide web, pp. 1369-1370.
3. Radovan, Mauricio and Pretorius, Laurette. Facial animation in a nutshell: past, present and future. Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries. 2006
4. Parke, Frederick I. Computer generated animation of faces. Proceedings of the ACM annual conference - Volume 1. 1972
5. DeRose, T.; Kass, M. & Truong, T. Subdivision surfaces in character animation. Proceedings of the 25th annual conference on Computer graphics and interactive techniques, ACM, 1998, 85-94
6. Yarimizu, H.; Ishibashi, Y.; Kubo, H.; Maejima, A. & Morishima, S. Muscle-based facial animation considering fat layer structure captured by MRI SIGGRAPH '09: Posters, ACM, 2009, 9:1-9:1
7. Su, M.-C. & Liu, I.-C. Application of the Self-Organizing Feature Map Algorithm in Facial Image Morphing. Neural Process. Lett., Kluwer Academic Publishers, 2001, 14, 35-47
8. Fei, K. Expressive textures. Proceedings of the 1st international conference on Computer graphics, virtual reality and visualisation, ACM, 2001, 137-141
9. Huang, H.; Chai, J.; Tong, X. & Wu, H.-T. Leveraging motion capture and 3D scanning for high-fidelity facial performance acquisition. ACM Trans. Graph., ACM, 2011, 30, 74:1-74:10
10. Beeler, T.; Hahn, F.; Bradley, D.; Bickel, B.; Beardsley, P.; Gotsman, C.; Sumner, R. W. & Gross, M. High-quality passive facial performance capture using anchor frames. ACM Trans. Graph., ACM, 2011, 30, 75:1-75:10
11. Arghinenti, A. Animation workflow in KILLZONE3: a fast facial retargeting system for game characters. ACM SIGGRAPH 2011 Talks, ACM, 2011, 37:1-37:1
12. Deng, Z.; Chiang, P.-Y.; Fox, P. & Neumann, U. Animating blendshape faces by cross-mapping motion capture data. Proceedings of the 2006 symposium on Interactive 3D graphics and games, ACM, 2006, 43-48
13. Fathom Studios for Deldo. http://www.creativecrash.com/maya/tutorials/character/c/facial-animation-rig-for-delgo
14. Talking Heads.
http://www.gamasutra.com/view/feature/3089/talking_heads_facial_animation_in_.php
15. Andy Van Straten. http://andy-van-straten.com/
16. Cohen, M. M. & Massaro, D. W. Modeling Coarticulation in Synthetic Visual Speech. Models and Techniques in Computer Animation, Springer-Verlag, 1993, 139-156
17. Yamamoto, E.; Nakamura, S. & Shikano, K. Lip movement synthesis from speech based on hidden Markov models. Speech Commun., Elsevier Science Publishers B. V., 1998, 26, 105-115
18. Massaro, D. W.; Beskow, J.; Cohen, M. M.; Fry, C. L. & Rodriguez, T. Picture My Voice: Audio to Visual Speech Synthesis using Artificial Neural Networks. In D. W. Massaro (Ed.), Proceedings of AVSP'99: International Conference on Auditory-Visual Speech Processing, 133138, 1999, 133-138
19. Benin A, Leone G.R, Cosi P, Web3D'2012, "A 3D talking head for mobile devices based on unofficial iOS WebGl Support", Proceedings of the 17th International Conference on 3D Web Technology, pp.117-120
20. Gachery, S. e Magnenat-Thalmann, N. (2001) "Designing MPEG-4 Facial Animation Tables for Web Applications", Multimedia Modeling Conference, Amsterdam, Holanda, pp.39-56

21. Cantor D, Jones Br, “WebGL Beginner’s Guide”
22. Lavagetto, F. and Pockaj, R., 1999. The Facial Animation Engine: towards a high-level interface for the design of MPEG-4 compliant animated faces. IEEE Trans. on Circuits and Systems for Video Technology, 9(2), pp.277-289.
23. Contreras, V., 2005. Artnatomy. <http://www.artnatomia.net/uk/artnatomiaIng.html> [Accessed 27 November 2012].
24. Lewis, J.P., Cordner, M. and Fong, N., 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. Proceedings of the 27th annual conference on Computer Graphics and interactive techniques, pp. 165-172.
25. Dornaika, F. and Davoine, F, 2006. On Appearance Based Face and Facial Action Tracking. IEEE Transactions on Circuits and Systems for Video Technology, 16(9), pp. 1107-1124.