# Optimized Face Animation with Morph-Targets

Uwe Berner
TU Darmstadt, Interactive
Graphics Systems Group (GRIS)
Fraunhoferstrasse 5
64283 Darmstadt, Germany

uberner@gris.informatik.tu-darmstadt.de

## ABSTRACT

An ongoing research topic is to build suitable visualization architectures for anthropomorphic conversational user interfaces which will run on different devices like laptops, PDAs and smart phones. One interesting way to handle the performance restrictions on small and mobile devices are scalable avatars and animations. An idea to speed up our animation system based on morph targets is presented, to achieve this the number of animation weights for the morph targets is reduced dynamically. This method is optimized by regarding the graphical structure and dependancies of the morph targets.

## Keywords

Face Animation, Morphing, Scalability, Mobility.

## 1. INTRODUCTION

Computers are moving more and more from the desktop into our everyday life. This leads to the important question of how the user interfaces should look like for this new generation of computing. One idea is the use of human like conversational agents to interact with the user in dialogs.

Thus an important field of activity at the Interactive Graphics Systems Group (GRIS) are Conversational User Interfaces where the primary goal is to give the computer a face to talk with. Conversational User Interfaces uses natural dialog-centric communication patterns to interact with the computer. The user is migrating from the paradigm of direct manipulation to the usage of assistance functionality [Maes94]. The goal is the development of software architecture to shift complex tasks to human like assistants (avatars) which can be incorporated on different stationary and mobile devices like laptops, PDAs and mobile phones.



***Journal of WSCG, Vol.12, No.1-3, ISSN 1213-6972***
*WSCG'2004, February 2-6, 2003, Plzen, Czech Republic.*


Today's challenge is to build a suitable visualization architecture for anthropomorphic conversational user interfaces which will run on different devices like laptops, PDAs and smart phones. Concrete implementations as part of conversational interfaces are User-Interface Avatars which are anthropomorphic representatives on the base of artificial 3D characters.

Scalable animation is an interesting way to handle the performance restrictions on small and mobile devices. Our actual work deals with scalability of animation and graphical representation of avatars, in this paper we will focus on optimization strategies for the animation. At first we present an idea to speed up our animation system and in the second part we show some improvements. At the end we give some perspective of the further development of the system.

## 2. RELATED WORK

There are a lot of systems dealing with virtual humans and dialogcentric interfaces. Normally there is no adaptability of the graphical representation on the running system, if you need a less complex model you make it by loading a different one. Some examples of systems featuring virtual humans are [Ava03], [Lex03] and [Mira03]. They use advanced techniques to animate and visualise synthetic characters and a lot of effort is done by using standards like H-ANIM[Anim03]. But more and more mobile devices are emerging and people want to use interfaces which were easy to use on small screens. One solution is to use animated avatars to communicate with the user. Thus some other systems are focussing on mobile aspects like [Inca03], [Char03] and [Ant03]. Neverthenless there are a few systems dealing with the scalability of the graphical representation of the virtual humans during runtime like [Mag00], [Bre03], [Bre01] and [Ber02].

## 3. ANIMATION SYSTEM

The Animation System is devided in three components: One for the graphical representation, one for the speech synthesis and one for controlling the behavior of the avatar (see Fig. 1). All this components are driven from an external dialog control giving the commands to the avatar like "speak this sentence" or "do this gesture".

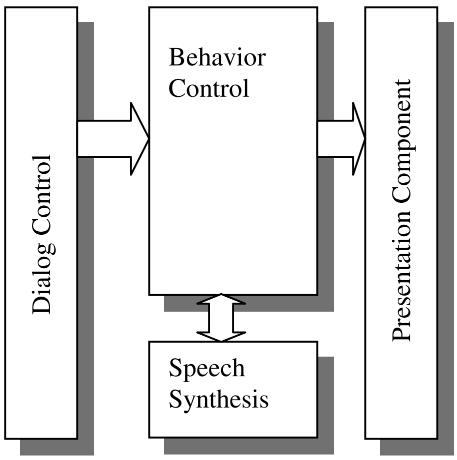



**Figure 1. Components of the Animation System.**

The detailed functions are:

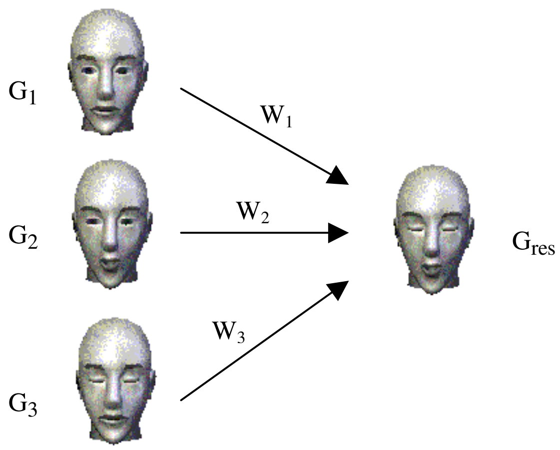
**Presentation Component:** This module provides the functionality to present animated artificial characters, to perform facial animations, and to achieve lip sync. The facial structure is based on morph targets and efficiently realized as a morph node [Alexa00]. Hereby, a wide range of facial expressions can be provided, while the parameterization of the face is kept simple and can be realized with just a few key values. Keyframes defining the state of the animated character may consist only of a small number of such values and playback is possible even over low-bandwidth networks and on small portable devices. Moreover, facial animations can easily be mapped to different geometries, thus further extending the possible choices for the user interface designer. In this way, photo-realistic avatars or comic-like characters can be displayed and animated.

**Speech Synthesis:** An important requirement for the avatar in order to achieve a realistic appearance is speech output with lip-sync. Since the dialog with the user is not known in advance, prerecorded animation sequences with lip animation cannot serve as a solution. Instead, the real-time lip sync with the audio output generated by a speech synthesis module is developed. For this, the speech synthesis system is expected to generate phoneme information, which will be mapped to the appropriated viseme. Phonemes are communicated using the international standard SAMPA [Sampa95]. The mapping to a viseme or sequences of visemes is based on timing and frequency information available in SAMPA. Heuristics are used to generate nonverbal facial expressions from phoneme information.

**Behavior Control:** While the Presentation Component allows for the control of the avatar on the fundamental geometric level, the Behavior Control provides an interface on a more abstract level. Tasks and even motivations can be specified and the corresponding actions are performed automatically. Examples for such actions are gestures and movements of the avatar, but also more complex facial expressions with temporal aspects. In addition, behavior patterns for specific motivations or moods can be activated. This corresponds to a sometimes rule-based, sometimes non-deterministic activation of behavior elements. Possible applications for this are simple avatar actions (e.g. accidental looking around or smiling) to avoid repetitive behavior, which is easily detected as artificial by the human observer. One of our objectives is to make the avatar more natural and less artificial or stiff. Even if the underlying application layer does not invoke any action the avatar presents randomly generated behaviors like eye blinking or head movement. The different animation sequences representing an explicit gesture or an unconscious behavior are synchronized via a blackboard mechanism, which also controls the appearance of the emotional state of the avatar.

## 4. ANIMATION WEIGHTS

The key idea behind the animation with morph targets is to combine different geometries corresponding to their weights (see Fig. 2). The different morph targets $G_1$:"neutral", $G_2$:"O" and $G_3$:"closedEyes" were combined regarding different weights ($w_1$, $w_2$ $w_3$) to the resulting geometry $G_{res}$. The weights $w_i$ are the output stream of the Behavior Control modeling the face animation sequence of the avatar.




$G_{res}= w_1*G_1+w_2*G_2+w_3*G_3$

**Figure 2. Combination of Morph-Targets.**

During the animation the values of the weights are changing dynamically. Some weights are more important which is represented by a higher value. The idea is to speed up the animation by ignoring weights with small values. Then the morph process computes the results with less morph targets. We set the weights under a specified value (threshold) to zero and distribute these weights over the remaining ones. This action is required in order to achieve a total sum of one for all weights corresponding to the rules of the Java3D morph node. Two strategies are used for this:

1. Add the weights to the maximum weight.
2. Add the weights to the remaining non zero weights.

The speed up for the two strategies is the same as we are using the same number of morph-targets. It is about 23% with threshold 0.2, a higher threshold leads to a better performance but also to animations of lower quality. In Figure 3 you can see the different framerates with different thresholds as the avatar is reciting a poem (using an AthlonXP 2100, GeForce 4)."Ohne Optimizer" is the german expression for "without optimizer".

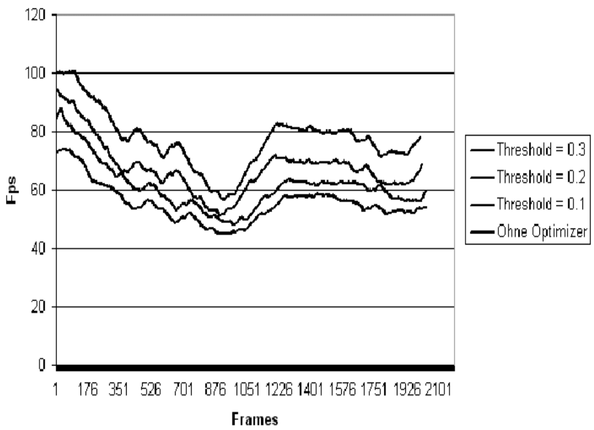



**Figure 3. Optimized Weights.**

## 5. IMPROVED ANIMATION

As seen in Figure 3 we have a satisfying speed up of our animation. But the visual appearance could be better, with the usage of bigger thresholds the animation becomes more and more choppy.

To solve this problem we found the following solution: The differences between the neutral morph target and the others were incorporated in the computation of the optimized weights.

**The first step** is to compare each morph target with the neutral one. We computed the euclidean distance between the corresponding vertices from the two morph targets and add all distances to a sum. The distance ($dist_{morph}$) between the neutral morph target ($T_n$) and for example the morph target of the vowel a ($T_a$) is computed over the euclidean distance ($dist_{eucl}$) of the corresponding vertices $v_i(T_n)$ and $v_i(T_a)$.

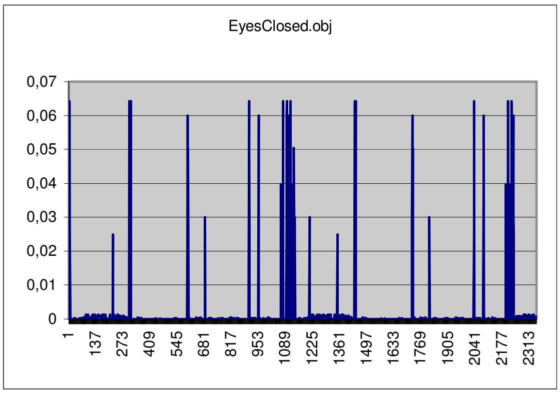
$$dist_{morph}(T_n,T_a)=\sum dist_{eucl}(v_i(T_n),v_i(T_a))$$

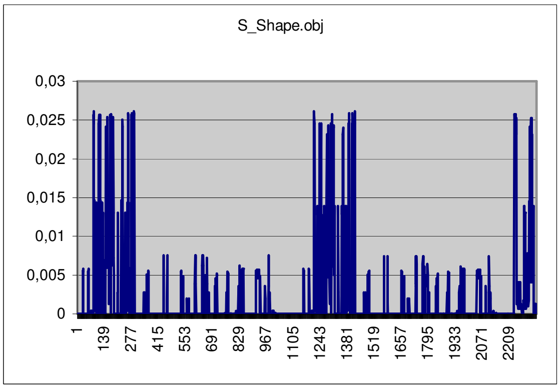
For a better comparison the distances of the morph targets are normalized with the maximum of the $dist_{morph}$. You can see a list of normalized distances in Table 1. Of course the distance of the neutral target is 0, some targets with not much animated regions of the face like "S" or "eyes closed" have small values, a lot of moved vertices in comparison with the neutral target are in "mouth smile" and "O". Thus the most important morph target for the animation in this example is "O".

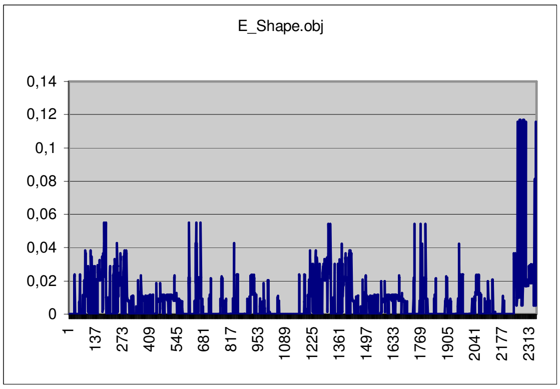
| Morph-Target | Normalized Distance |
|---|---|
| Neutral | 0 |
| EyesClosed | 0.11 |
| BrowsUp | 0.12 |
| S | 0.16 |
| F | 0.20 |
| BrowsFright | 0.31 |
| BrowsAngry | 0.39 |
| L | 0.51 |
| E | 0.62 |
| A | 0.66 |
| MouthSmile | 0.67 |
| MouthSad | 0.90 |
| O | 1 |

**Table 1. Normalized Distances of Morph-Targets**

For a better understanding of the different morph targets we present some images of the distances to the neutral morph target. You can see in Figure 4 the distances for every point of a morph target.



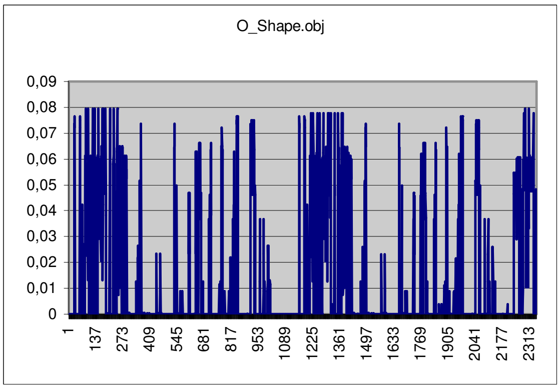




**Figure 4. Distances of Morph-Targets.**

One additional point is very interesting: The regions with high distances are potentially most animated, in addition to the sum of distances this regions can be used for other optimizations of the animation.

**The second step** is to set the weights to zero regarding the threshold and the normalized distances of each morph target. The decision to set a weight to zero results from the following equation. The threshold from the old method is called $th_{fix}$ , the normalized distance of the morph target $dist_{norm}$ and the adjusted threshold $th_{adj}$ .This leads to:

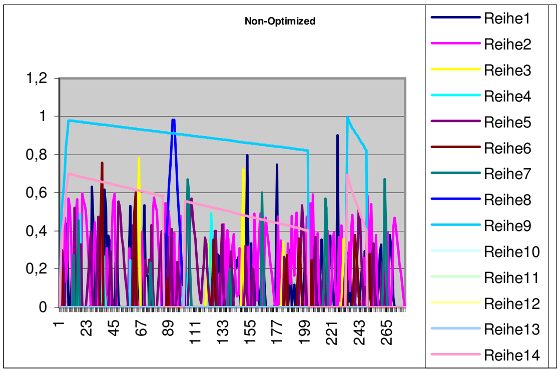
$$th_{adj} = th_{fix} * (1-dist_{norm})$$

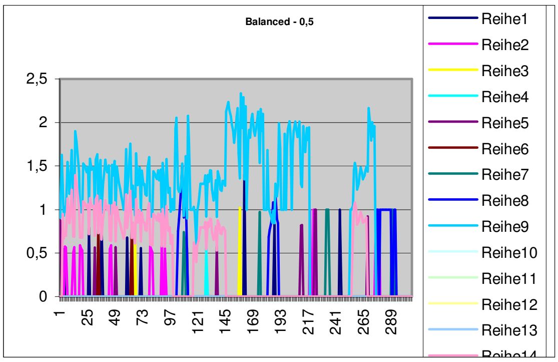
The substraction is used to give the more important morph targets a lower threshold. This implies a longer remaining in the animations.

To quantify the performance of the new method we made some new measurements of the frame per seconds output rate. We used a 700 Mhz Intel Processor with ATI-Rage-Mobilty 128. The avatar was speaking a longer text and we recorded the fps-rate from a part of the animation.

With the old optimization method we get a speed up of 23 %, the new one is nearly in the same dimension.

Above all the quality of the graphical output is much better than before. There is only a slight difference between the new method and the non optimized version. The best way to see this is a video, but we made also some mesurements of the used weights during the animation. The structure of the animation weights over the time resulting from the new method is more like the non optimized version, which is a hint of the graphical output quality. In the “balanced adjusted” diagram more weights are used and there are less holes over the time. You can see this in Figure 5 the first diagram is the non optimized version, the second is the old optimization method (balanced) and the last is the improved method (balanced adjusted); the weights are called “Reihe 1” and so on for the rows of an excel sheet.



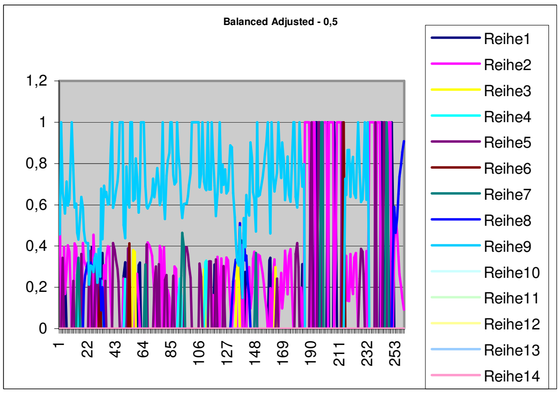




**Figure 5. Different Weights during Animation.**

A screenshot of the running system could be seen on Figure 6. You can see the avatar, the upper window to select a sentence to speak, a middle window to place the avatar and the lower one to select the optimization method and to display the frames per second rate.

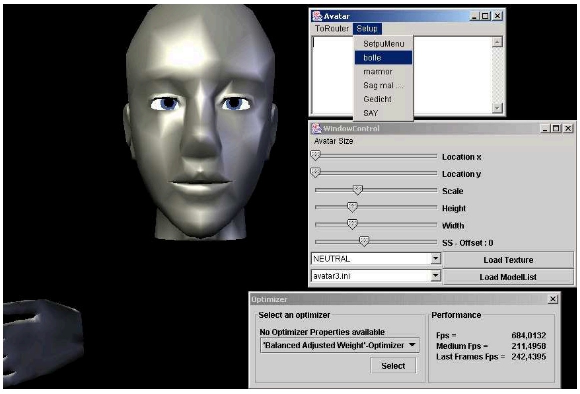



**Figure 6. Screenshot of the running system.**

# 6. FUTURE WORK

For the future we want to quantify the graphical advantage of the new optimization method better than the diagrams in Figure 5. One idea is to record the position of all vertices during the animation. Then we can compare the distances of the optimized face output with the normal animation for some specified frames during the animation. This will illustrate the graphical differences between all three animations over the time. Also we can experiment more with the function to get the new threshold leaded by the diagrams of Figure 5 or the total number of non zero weights. Even it could be interesting to give the morph targets used for the speech animation a higher threshold.

Another interesting thing are the animated regions which can be extracted from the images in Figure 4. This can lead to a modified mesh reduction of the avatar were highly animated regions are less reduced than other ones.

## 7. CONCLUSION

We have presented a system to animate a virtual human which is based on the usage of morph targets. Some control components are producing an animation of the virtual character by creating a stream of weights. We are working on the scalability of the avatar to make the system available on smaller devices. To achieve this an optimization method for the animation was developed based on the reduction of the number of non zero weights. To enhance the quality of the graphical output the structure of the morph targets is included in the optimization algorithm. At last some ideas to quantify the quality of the animation were given and concepts for other scalability methods like mesh reduction are mentioned.

## 8. ACKNOWLEDGEMENTS

Our thanks to Oliver Plohmann, who made the measurements, some programming and the graphics. This work was partially funded by the German "Bundesministerium für Bildung und Forschung", through the Research Project Virtual Human, see [VH03].

## 9. REFERENCES

[Alexa00] Alexa, M., Behr, J., Müller, W. (2000). The Morph Node. Proc. Web3d/VRML 2000, Monterey, CA., 2000

[Anim03] www.hanim.org

[Ant03] www.anthropics.com

[Ava03] www.zgdv.de/avalon

[Ber02] U Berner and T. Rieger: A Scalable Avatar for Conversational User Interfaces, User-Interfaces for All, Paris 2002.

[Bre01] G. Breton, C. Bouville and D. Pelé, FaceEngine : A 3D Facial Animation Engine for Real Time Applications, Web 3D Symposium, Paderborn, Germany, 2001.

[Bre03] G.Breton: Scalable Facial Animation, 3D Modelling 2003, Paris, France 2003.

[Char03] www.charamel.de

[Inca03] www.cse.unsw.edu.au/~inca

[Lex03] www.lexicle.com

[Maes94] P. Maes, Agents that Reduce Work and Information Overload. Comm. of the ACM, 37 (7), 1994

[Mag00] H. Seo and Magnenat-Thalmann, LOD Management on Animating Face Models, Proceedings of IEEE on Virtual Reality 2000, 2000.

[Mira03] www.miralab.ch

[Sampa95] SAMPA - Computer readable phonetic alphabet, www.phon.ucl.ac.uk/home/sampa/home.htm, 1995

[VH03] www.virtualhuman.de