Efficient Distribution of Emotion-related Data through a Networked Virtual Environment Architecture

Peter Quax Fabian Di Fiore Wim Lamotte

Frank Van Reeth

Hasselt University - tUL - IBBT

Expertise Centre for Digital Media

Wetenschapspark 2

BE-3590 Diepenbeek

Belgium

{peter.quax, fabian.difiore, wim.lamotte, frank.vanreeth}@uhasselt.be

http://www.edm.uhasselt.be

Abstract

In this paper we describe the way in which emotion-related data can efficiently be exchanged between participants in a large-scale networked virtual environment. This type of metadata is extracted from real-time captured video streams using off-the-shelf webcams and applied onto a 2D stylized avatar; thereby improving the immersion the user experiences while navigating and communicating in the virtual world. As emotion-related data – once processed through the system – can be considered a specific type of state information, a generic networked virtual environment architecture can be used to distribute the information between participants. We have opted to extend the in-house developed ALVIC-NG architecture to be able to process the information flows. We will show that the inclusion of this new type of information does not have a detrimental effect on the scalability of the system.

Keywords: networked virtual environments, immersive communication, scalability, facial animation, emotions, avatars, MPEG-4

1 Introduction

Conveying emotional state between users in a virtual environment is an essential component in achieving total immersion in a three-dimensional world. The currently existing systems provide unsatisfactory results, as they require quite a bit of user intervention in order to keep the avatar's state in sync with the actual emotional state of the user. While this manual process suffices for a coarse impression of the emotional state of the correspondent, reality is that emotional state changes rapidly and in a non-discrete fashion. To keep up with the pace in which the parameters change over the course of a conversation, an automatic metadata extraction and visualisation system for emotion-related data is clearly required. In previous work[1], we have shown how to extract the emotion-related metadata from real-time video streams — captured through a webcam pointed towards the user's face — and apply these onto a stylised 2D representation of an avatar. The solution consists of a hybrid approach, combining benefits of facial animation and user-controlled 2D modelling and animation techniques. Facial animation is employed to extract the movement and timing of the user's facial components, lessening the need for high-bandwidth channels typically required by video-enabled applications. In the visualisation phase, we opt for a structured 2D methodology as the face to which the captured facial movements are applied can be either drawn by hand or based on real footage.

While our previous work considered the bandwidth usage of the streams generated by this type of information, the real distribution of this type of 'state' information was not discussed. In this paper, we will introduce emotion-related information in an existing networked virtual environment architecture called ALVIC-NG [2]. We will show that emotion-related information can be handled in a similar way to generic 'state' information such as positional data – which makes up the bulk of information that is already exchanged between participants. In practice, this means that the required extensions and adaptations are minimal; thereby making sure that the scalability of the architecture is not impacted in a negative way.

This paper is organised as follows. Section 2 provides a short discussion of some related work. In sections 3 and 4, an overview is presented on the required technology for the extraction of emotion-related information from real-time video sequences, as well as the design of the ALVIC-NG architecture. Subsequently, the integration of the new data streams is discussed, along with any alterations required to the architecture (section 5). In section 6, some test results will be discussed that show that the inclusion of emotional information does not impact the scalability of the architecture.

2 Related Work

As our solution consists of two main parts, we will discuss some of the related work that is specific to both.

For a detailed comparison between our approach on extracting the emotion-related metadata and the state-of-the-art, we refer to our previous work published in [1]. We will provide a short overview to point out the main differences between existing literature and the proposed solution. In 2D, either a set of parameters (e.g., co-articulation regions (CR) as discussed in [3]) or existing animations (as in [4]) can be used to generate new examples. However, mappings are difficult to automate and cartoon-style expressions are non-trivial to create. Specialized systems for facial animation are also in existence (e.g., [5]), but they do not support transformations outside the drawing plane. Automatic generation of stylized cartoon renderings is made possible through the application of (3D) NPR techniques (e.g. [6, 7]). However, it is increasingly difficult to obtain the same 'lively' results as typical 2D animation.

Regarding the distribution of immersive communication streams, a distinction should be made between systems that focus on capturing entire 3D models (such as [8]) and others that use additional data obtained from other sensors to precisely position avatars in a 3D environment (e.g., [9]). Some other solutions consist of the application of full-body video onto a 3D mesh of a virtual human figure (see [10]). The main difference between these examples and the solution we propose is that the distribution of the information is done in a very trivial way. Nearly all the proposed systems rely on the presence of a high-bandwidth link between all participants, and limit the total number of active participants in a single session.

While our previous work considered the bandwidth usage of the streams generated by this type of information, the real distribution of this type of 'state' information was not discussed. In this paper, we will introduce emotion-related information in an existing networked virtual environment architecture called ALVIC-NG. We will show that emotion-related information can be handled in a similar way to generic ‘state’ information such as positional data – which makes up the bulk of information that is already exchanged between participants. In practice, this means that the required extensions an adaptations are minimal; thereby making sure that the scalability of the architecture is not impacted in a negative way.

3 Capturing and Visualizing Emotions

The novelty of our approach lies in how we combine benefits from performance-driven facial animation and user-controlled 2D modelling and animation techniques. Performance-driven facial animation is employed to extract the movement and timing of the user’s facial components, lessening the need for high-bandwidth channels typically required by video-enabled applications. In the visualisation phase, we opt for a structured 2D methodology as the face to which the captured facial movements are applied can be either hand drawn or incorporated real footage.

3.1 Avatar Creation

Instead of drawing a ‘complete’ face at once, every individual part (face outlines, mouth, nose, left eye, right eyebrow, . . .) of the face can be drawn independent of the others. These facial components (also denoted facial channels) are arranged in a hierarchical manner, defined as Hierarchical Display Model (HDM).

In order to achieve convincing 3D-like animations, several view-dependent versions of the HDM (each depicting the same face but as seen from a different viewpoint) can be drawn in order to cover out-of-the-plane animation [11]. Considering facial animation from an artistic point of view, realistic behaviour is not always desired but there's a need for fake, yet very impressive or dramatic effects; especially when applied to avatars [12, 13, 14]. Hence, in addition several 'expressive' versions of each facial channel can be modelled covering the range of expressiveness held in the user's mind. So, for each expression type, all channels have a separate version. Figure 2 shows three extreme poses of a drawn animation character illustrating the discussed concepts: (a) is composed of 15 facial channels which all depict the same expressive version $e_{neutral}$ whereas (b) and (c) are made up of the same facial channels illustrating expressive versions $e_{emotional}$ and $e_{exaggerated}$. Typically, in total 18 to 27 extreme poses are more than sufficient to cover a wide range of views and expressions (9 depicting the several views, multiplied by 2 or 3 expressive versions).

Besides freely drawing extreme poses/frames starting from a blank canvas, our system also includes the possibility to create extreme frames by incorporating scanned drawings or real images depicting extreme poses (see Figure 3). Starting from incorporating a real image, depicting one extreme pose, the user can define layered mesh structures over certain image parts that contain interesting information. First, one or more initial meshes are created (using subdivision surfaces) for only one image, corresponding to one extreme frame. Then, other extreme frames are created by incorporating new images (each depicting another extreme pose) for which the user only has to modify a copied instance of the initial meshes. As a result multiple HDMs can easily be created, where each HDM again corresponds to a specific view.

3.2 Facial Motion Data Capture and Extraction

Facial motion data is directly captured from the user's movements using off-the-shelf hardware such as low-cost webcams and digital cameras. Unlike the rigid demands posed on the frame grabbing process by real-time video (i.e. a minimum of 10 fps for fluent motion), which is required for full-frame avatar reconstruction, our solution demands only a few frames to be grabbed in each time frame to achieve adequate results. To substantiate this claim, we remind the reader that the latter technique is much more computationally expensive (i.e. to reconstruct in-between frames in a video sequence) than to interpolate between a very limited set of (feature) coordinates representing the facial expressions on a 3D model.

After the raw video frames are captured, we use the face detection algorithms present in the OpenCV library, which in effect uses techniques based on Haar-like features. Because the existing Haar classifiers are complementary, we combined the results into a set of (possibly) overlapping rectangles, the union of which is calculated in a subsequent step. This step of the process ends with the creation of a set of rectangles representing the detected faces. Elementary image processing algorithms are applied to the detected regions in order to determine the location of the important features — which, for emotion recognition, are mainly the shape of the mouth and the eye/eyebrow combinations. We also exploit some well-known anatomical facts that help to speed up the processing, such as the assumption that there is a minimal distance between the features and their relative position with regards to one another. What we end up with is a black and white mask, for which it is easy to extract the required feature parameters (see Figure 4).

More in detail, data describing the movement of facial components is extracted [15] and made available on a multi-level basis according to the MPEG-4 characterisation [16]: (i, low level) movement of individual feature point positions relative to a set of facial invariant points according to the MPEG-4 Facial Feature Points Location; (ii, medium level) movement of defined areas of the face described in terms of MPEG-4 Facial Animation Parameters (FAPs); and (iii, high level) motion in terms of MPEG-4 Facial Expressions.

3.3 Animation System

After the extreme poses of the facial parts are modelled, the extracted facial motion data can be applied to animate the avatar. As the extracted motion of the facial components is made available on a multi-level basis, various mappings can be defined between the modelled facial channels and the extracted motion data.

At the lowest level, for each facial channel any arbitrarily control point or user-selected part of the channel can be enforced to inherit the motion of one of the captured MPEG-4 Facial Feature Points. At a medium level, each facial channel can be driven by one or more of the captured MPEG-4 Facial Animation Parameters (FAPs). At the highest level, ‘expressive’ versions of facial channels can be grouped together on the basis of expressing the same emotion (e.g., joy or sadness). At the medium level, for instance, user defined facial parts can be driven by one or more of the captured MPEG-4 Facial Animation Parameters (FAPs). This happens in an easy and interactive way during the modelling stage and requires only a reasonable amount of manual input. For each facial part, the animator only has to define regions (FAP regions) using a lasso tool and attribute each of them to a desired FAP. Figure 5 depicts a FAP region defined by the animator.

Once the desired mappings have been made, the extracted facial motion data is loaded into the animation system and all keyframes are automatically set, hence, driving the animation. As mentioned in previous section, our solution demands only a few frames to be grabbed in each time frame to achieve adequate results. Missing (i.e. non-captured) frames are reconstructed on-the-fly either by extrapolating or by interpolating between surrounding frames (after delaying the stream a few frames). For the case of drawn facial parts, the underlying curves making up the drawings are interpolated to create in-between drawings whereas for real footage all corresponding meshes imposed on the keyframes are warped to each other automatically across intermediate frames. We refer the reader to [17] for an in-depth explanation of the animation system.

4 The ALVIC-NG Architecture

The ALVIC-NG (Architecture for Large-Scale Virtual Interactive Communities) was developed with scalability and real-life deployment as primary features. This is achieved by layering the communication flow. An overview of the various layers is provided in figure 6. Contrary to traditional client/server architectures, ALVIC-NG provides an additional layer, which increases the scalability and facilitates deployment on generic access networks. More specifically, the Proxy layer shields the end-user application from direct contact with the Logic Servers, which maintain the state information for the entire virtual world. At the same time, the fact that clients are assigned to proxies based on a set of metrics, including the network delay and proxy processing load, provides better results than arbitrarily assigning the clients to a set of world servers (which is the classic approach in a sharded world).

Connections between the Proxies and the Logic Servers are created and broken down in a dynamic fashion. These connections are tightly coupled to the spatial subdivision scheme that is integrated in the architecture. In essence, each Logic server is responsible for managing the state of a specific part of the virtual world. While in theory any type of scheme can be used, for testing purposes a quad-tree-based spatial subdivision scheme is chosen. In practice, this means that the entire virtual world is divided into a set of 'cells', each of them containing a number of participants. These cells each correspond to a (virtual) geographic region in the virtual world. Once the amount of users in a specific region exceeds a pre-determined threshold, the region is split into 4 new cells, and each of them is assigned to a Logic Server. In fact, the number of active clients is only a simple example of a metric used to determine when to split the region; in general this can be adapted dynamically at run-time; as the optimal set of parameters may depend on the type of application that is deployed on top of the architecture.

The choice for a client/server based architecture is primarily based on our previous experiences with the development of a peer-to-peer based system (the original ALVIC architecture [18]). In practice, it turns out these systems exhibit a lot of problems when deploying them on real-life networks; and not just from a technical point-of-view. Peer-to-peer systems are notoriously difficult to integrate with the abundance of NAT routers and gateways that are being used by home users (precisely the main target group for these types of applications). Besides this, managing a pure peer-to-peer system is very difficult: issues such as moderation of content, cheat prevention, distribution of patches etc are much more complicated. From an application/service provider's point of view, these issues are crucial; they are also the main reason why pure peer-to-peer based architectures are not commercially being used today.

5 Distribution of Emotion-related Data through ALVIC-NG

The bulk of information being exchanged between participants consists of generic state information. In practice, this mainly consists of positional data (which in turn encompasses position, orientation and information on avatar limbs etc). Several optimization techniques exist (such as the well-known dead-reckoning optimization) to limit the amount of state updates that need to be sent over the network in order for all participants to have a consistent view of the virtual world. The state information in ALVIC-NG is not exchanged directly between clients, but rather forwarded (through the proxy layer) to the appropriate Logic Server, which in turn uses this information to calculate a new global state. We should point out here that the server may perform additional checks on the state updates it receives, before forwarding them to the other participants which are present in the region the server is responsible for. Collision checks, boundary crossings and validity of movement can all be checked by the Logic server. At the same time, the logic server is responsible for the persistency of the data; which in practice means that the global state should not only be kept in main memory, but occasionally stored in a database or other location.

It is of vital importance to note that there are various degrees of importance that can be attributed to state information. Positional and other types of data (e.g., monetary transactions,...) can be tagged with a higher importance than emotion-related data. While the degree of immersion will drop when emotion-related data is no longer available in a session (even temporarily), it is much more disturbing when the avatar would completely disappear from the screen or would remain stationary due to the lack of positional information. The persistent storage module integrated in ALVIC-NG provides the ability to disable the permanent storage of some types of information. As the emotion-related data is transient, it is highly likely that there is no need for it to be stored in a database and can be kept in main memory at all times. The overall load on the logic server is therefore only slightly increased, as operations in main memory are much more efficient than disk I/O.

As we stated above, the positional data does not need to be updated for every frame that is to be rendered, as optimizations exist that allow each client to determine positions based on previous state updates, coupled with directional and speed information. The same is true for emotion data, as there can be an interpolation between keyframes for the animation of the 2D rendering. In practice, positional updates are sent at a rate of approximately 3 updates per second, which yield visually pleasing results [19]. The same or a lower update rate can be chosen for the emotion-related information, which allows for piggybacking of the emotion-related data onto the packets already sent containing the positional information.

6 Test Results

As the proposed solution consists of both visual aspects and the distribution of information over the network, we will discuss these topics separately.

6.1 Scalability

In order to test the scalability of the proposed solution, a test setup is created on a dedicated low-cost PC hardware cluster. A specialized version of the client software is developed that does not perform rendering of scene information, but does output the appropriate information to the network layer. Several instances of this software are controlled by a governing application called the Bot Server. This setup allows us to test thousands of instances of the client software, without the need for large-scale beta test user groups. The software can be instructed (through a scripting interface) to perform actions that correspond to those of a human user. At the same time, it is possible for several instances to communicate with each other, and change their behavior according to a set of parameters (e.g., perform movement in group, predator-prey-like interactions,...).

As we mentioned before, the emotional state information is not stored in the database by the logic servers. It is therefore clear that the major impact of the addition of this type of information should be located on the Proxy servers. Figure 7 presents results that are obtained by averaging out the figures of 5 consecutive test runs of a simulation. Shown are the number of clients active in the simulation (on the X-axis) versus the delay that a client notices when receiving updates from the Logic server (which are being routed through the proxy server as described above). It should be clear that the RTT delay will increase once the Proxy server gets overloaded, as it handles a much higher number of connections than the Logic server. For purposes of this test, the interactivity threshold is put at 50ms (RTT), which brings it in line with the positional data streams (important as the data may or may not be piggybacked onto existing packets). The chart shows clearly that about 625 clients can be supported using a single instance of a proxy server. Going beyond this amount, the numbers increase in a nearly linear fashion - thereby illustrating that the scalability is not impacted in a negative way.

6.2 Capturing and Visualization

Figure 8 depicts some stills of a generic scene consisting of 3 billboards where a user is, at the same time, visualised using the entire video stream (first billboard) requiring large amounts of bandwidth, and two avatar forms including the emotional information (requiring minimal additional bandwidth). The second billboard depicts the emotion conveyed by the first billboard but retargeted to an expressive drawn man. For this avatar 18 extreme frames were used consisting of 9 versions which are used to cover different views multiplied by 2 emotional versions (i.e. either neutral or happy) which have been drawn for each view-dependent one. The model itself consists out of 15 facial parts and in total 33 FAP regions were defined. The third billboard shows some snapshots of the same emotions retargeted to a very expressive (i.e. always happy) drawn frog. In this example, only 4 extreme frames were used to drive the animation. The frog's face is composed of 14 facial channels. The last image shows two users communicating through their avatars.

7 Conclusions

In this paper we have presented a means of capturing and processing emotional state information, starting from real-time webcam streams. This information is subsequently used to animate a (stylised) avatar representation of the user. Technically, this is achieved by representing the coordinates as a set of MPEG-4 facial feature points and combining them with MPEG-4 Facial Animation Parameters. An in-betweening process is used to interpolate the keyframe information that is sent between the communicating parties. As emotion-related data can be considered a special type of 'state' information, we have shown that the in-house developed ALVIC-NG architecture for networked virtual environments can easily be adapted to support the distribution of this novel type of information. Test results have shown that there is no negative impact on the scalability of the architecture.

Acknowledgements

Part of the research at EDM is funded by the ERDF (European Regional Development Fund) and the Flemish government. The authors would like to thank Panagiotis Issaris, Cedric Vanaken and Jeroen Dierckx for their help.

References

[1] Fabian Di Fiore, Peter Quax, Cedric Vanaken, Wim Lamotte, and Frank Van Reeth. Conveying Emotions through Facially Animated Avatars in Networked Virtual Environments. *Motion in Games 2008 (MIG08), Lecture Notes in Computer Science (LNCS)*, 2008.

[2] P. Quax, J. Dierckx, B. Cornelissen, and W. Lamotte. ALVIC versus the Internet : Redesigning a Networked Virtual Environment Architecture. *International Journal on Computer Games Technology*, 2008(Article 594313), 2008.

[3] Douglas Fidaleo and Ulrich Neumann. CoArt: Co-articulation Region Analysis for Control of 2D Characters. In *Proceedings of Computer Animation (CA2002)*, pages 17–22, June 2002.

[4] Christoph Bregler, Lorie Loeb, Erika Chuang, and Hishi Deshpande. Turning to the masters: Motion capturing cartoons. In *Proceedings of SIGGRAPH*, volume 21(3), pages 399–407. ACM, July 2002.

[5] Kristinn R. Thórisson. Toonface: A system for creating and animating interactive cartoon faces. Technical report, MIT Media Laboratory, Learning and Common Sense 96–01, April 1996.

[6] Bruce Gooch and Amy Ashurst Gooch. *Non-Photorealistic Rendering*. A. K. Peters Ltd., ISBN: 1568811330, 2001.

[7] Thomas Strothotte and Stefan Schlechtweg. *Non-Photorealistic Computer Graphics. Modeling, Rendering, and Animation*. Morgan Kaufmann Publishers, ISBN: 1-55860-787-0, 2002.

[8] Joseph Insley, Daniel Sandin, and Thomas DeFanti. Using video to create avatars in virtual reality. In *Visual Proceedings of the 1997 SIGGRAPH Conference*, page 128, Los Angeles CA, 1997.

[9] Tetsuro Ogi, Toshio Yamada, Ken Tamagawa, and Michitaka Hirose. Video avatar communication in a networked virtual environment. In *Proceedings of the 10th Annual Internet Society Conference*, volume Electronic edition, 2000.

[10] S. Yura, T. Usaka, and K. Sakamura. Video avatar: Embedded video for collaborative virtual environment. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, volume 2, page 433, 1999.

[11] Fabian Di Fiore, Philip Schaeken, Koen Elens, and Frank Van Reeth. Automatic inbetweening in computer assisted animation by exploiting 2.5D modelling techniques. In *Proceedings of Computer Animation (CA2001)*, pages 192–200, November 2001.

[12] Preston Blair. *Cartoon Animation*. Walter Foster Publishing Inc., ISBN: 1-56010-084-2, 1994.

[13] Ronen Barzel. Faking dynamics of ropes and springs. *IEEE Computer Graphics and Applications*, 17:31–39, 1997.

[14] Richard Williams. *The Animator's Survival Kit.* Faber and Faber Limited, ISBN: 0-571-20228-4, 3 Queen Square London WC1N 3AU, 2001.

[15] Donald Mac Vicar, Stuart Ford, Ewan Borland, Robert Rixon, John Patterson, and Paul Cockshott. 3D performance capture for facial animation. In *Proceedings of 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 42–49, 2004.

[16] Igor S. Pandzic and Robert Forchheimer. *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. John Wiley & Sons, ISBN: 0-470-84465-5, 2002.

[17] Fabian Di Fiore and Frank Van Reeth. Multi-level performance-driven stylised facial animation. In *Proceedings of Computer Animation and Social Agents (CASA2005)*, pages 73–78, October 2005.

[18] Peter Quax. *An Architecture for Large-Scale Virtual Interactive Communities*. PhD thesis, Transnationale Universiteit Limburg, 2007.

[19] Fengyun Lu, Simon Parkin, and Graham Morgan. Load balancing for massively multiplayer online games. In *NetGames '06: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, page 1, New York, NY, USA, 2006. ACM.

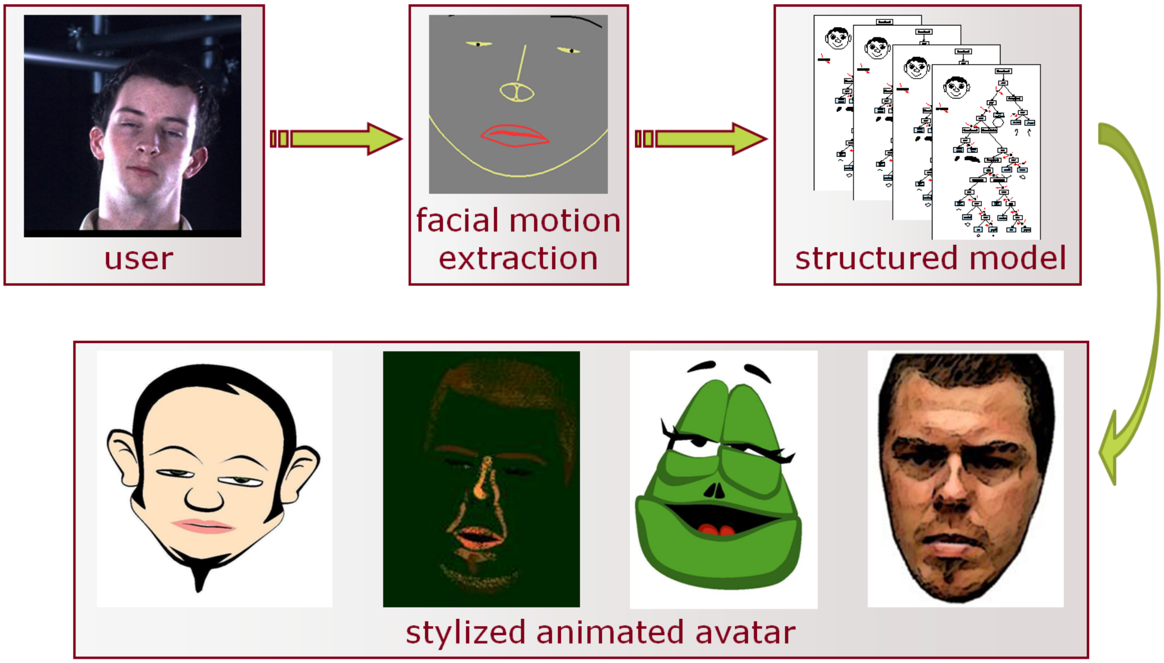

Figure 1: Overview of the different components of creating and animating an avatar.

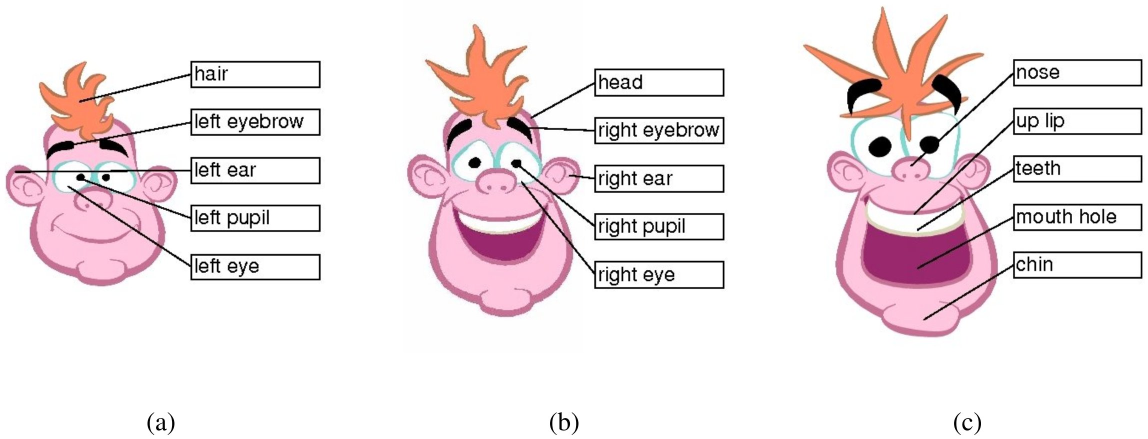

Figure 2: Some extreme poses of a drawn animation character composed of only 15 facial channels depicting three expressive versions: (a) $e_{neutral}$, (b) $e_{emotional}$, and (c) $e_{exaggerated}$.

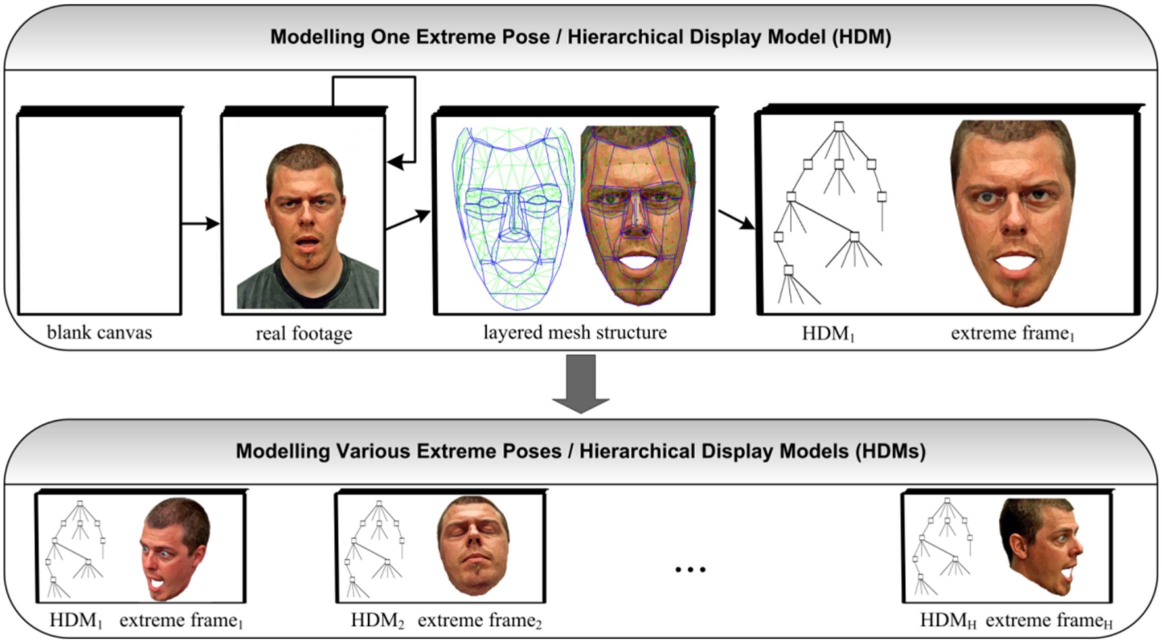
Modelling One Extreme Pose / Hierarchical Display Model (HDM)
blank canvas
real footage
layered mesh structure
HDM$_1$
extreme frame$_1$
Modelling Various Extreme Poses / Hierarchical Display Models (HDMs)
HDM$_1$ extreme frame$_1$
HDM$_2$ extreme frame$_2$
...
HDM$_H$ extreme frame$_H$

Figure 3: Overview of the different components of creating an avatar by incorporating scanned drawings or real images depicting extreme poses.

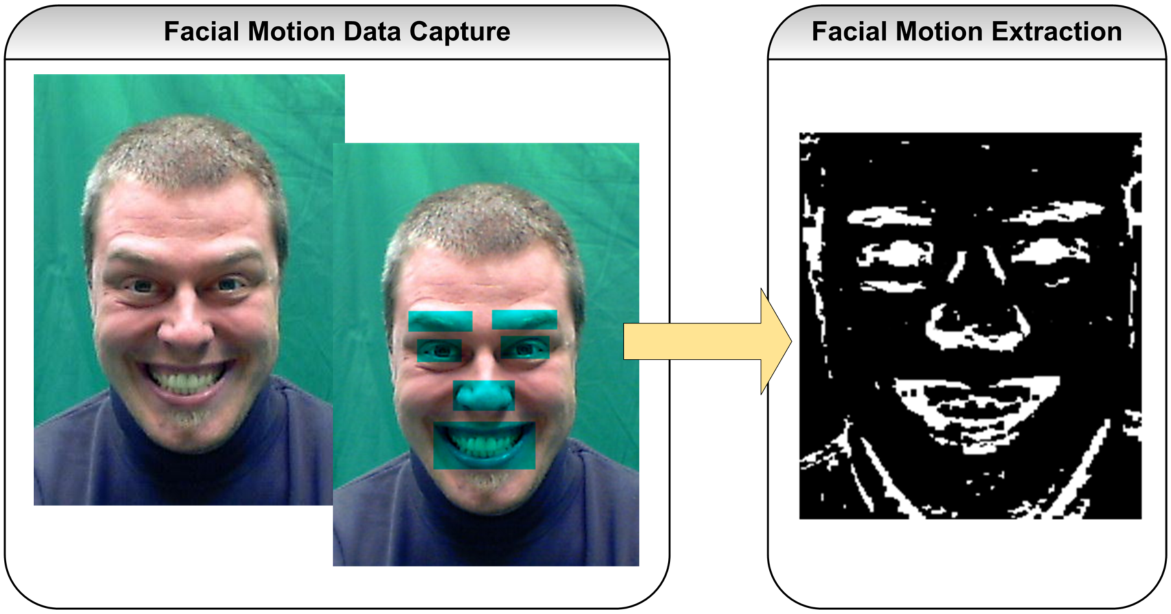

Figure 4: Overview of the process of facial motion data capture and extraction.

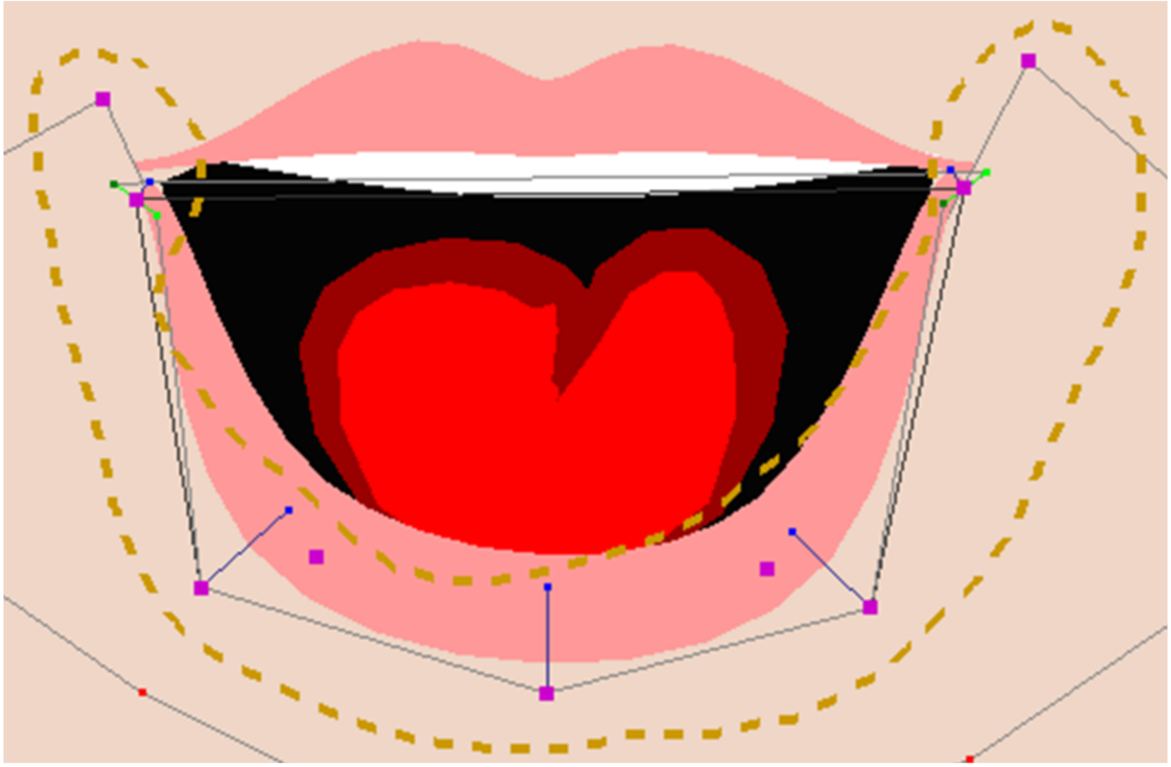

Figure 5: Example of a user defined FAP region.

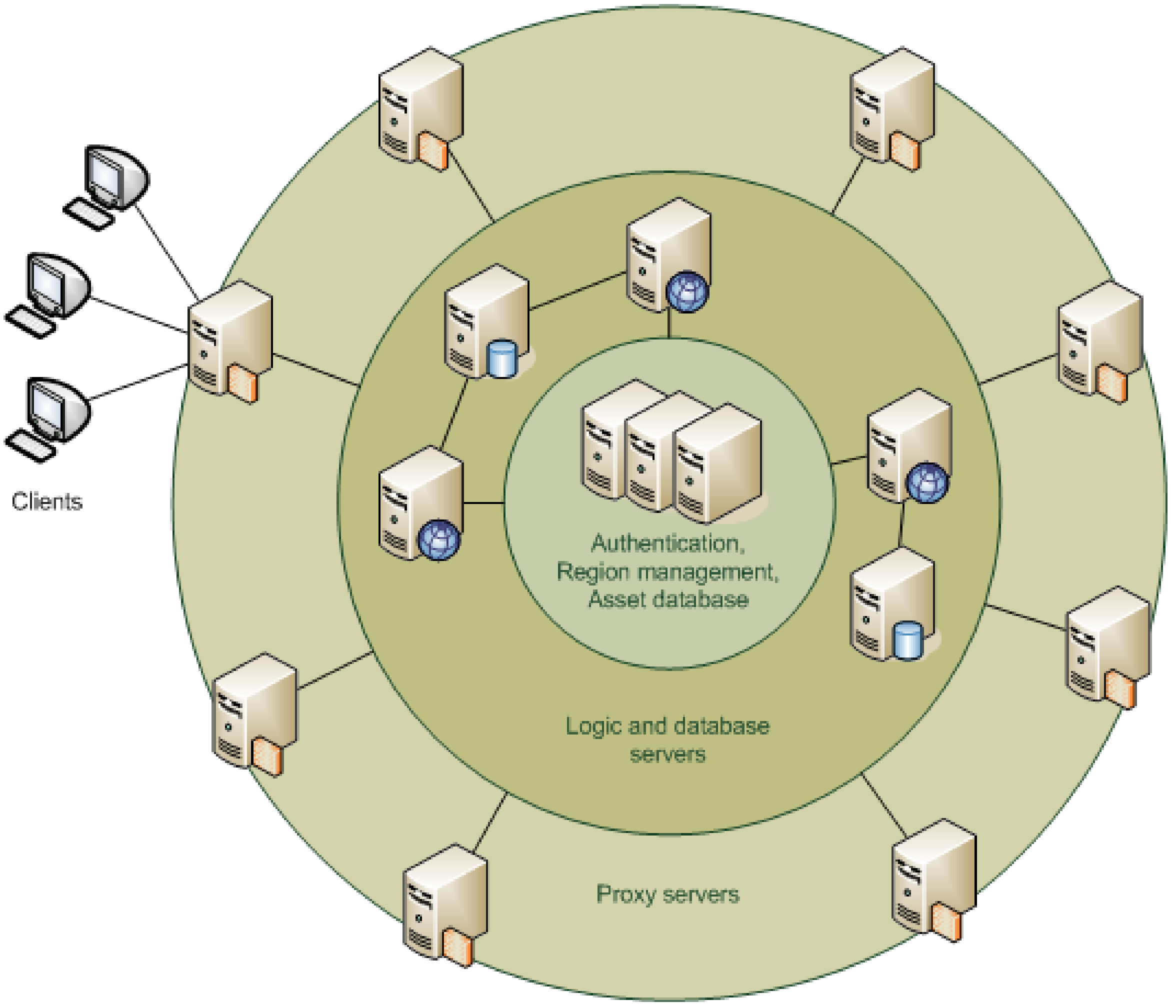

Figure 6: Components of the ALVIC-NG architecture

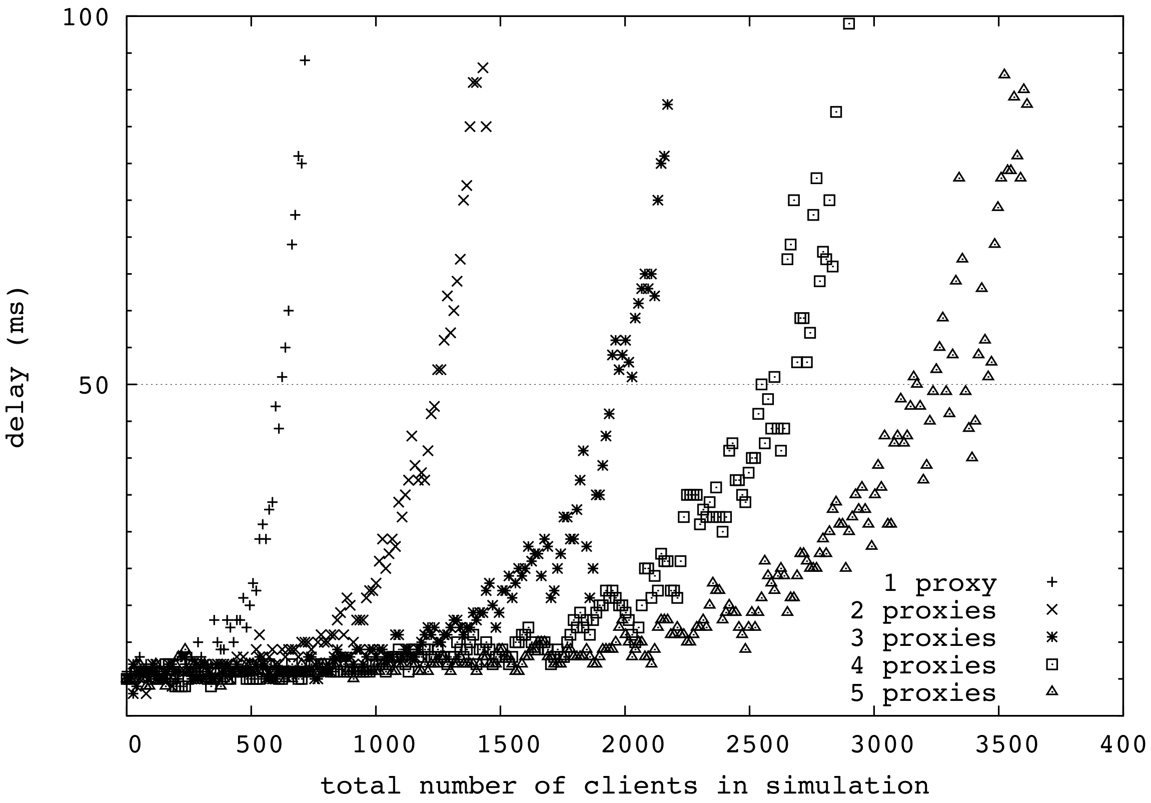

Figure 7: Scalability test results (averaged over 5 simulation runs)

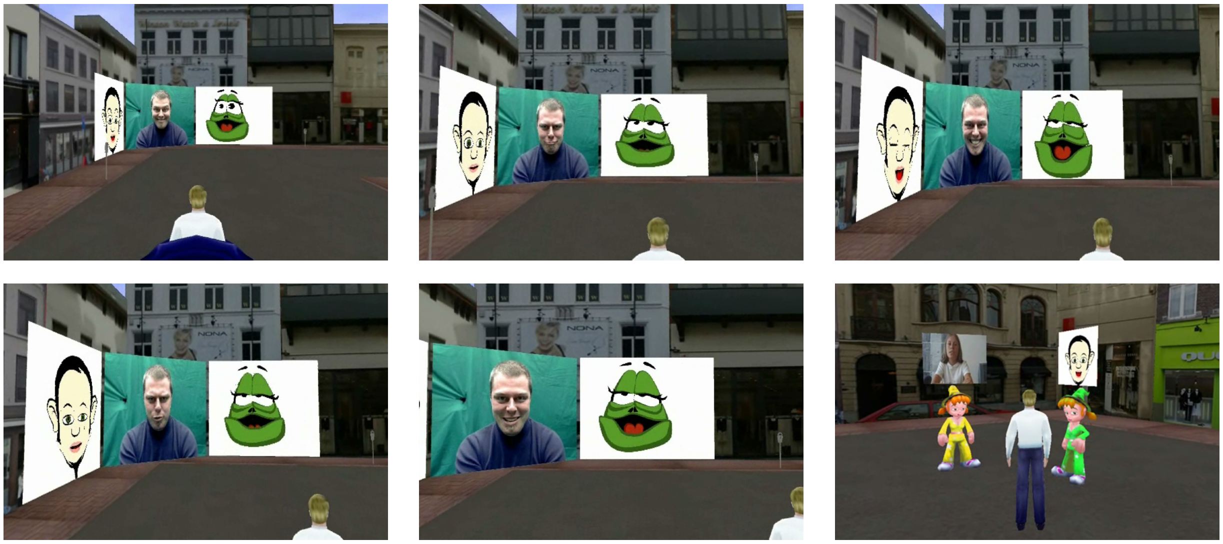

Figure 8: Snapshots of a generic scene consisting of 3 billboards. The first billboard visualises the input data whereas the second and third depict the emotion conveyed by the first billboard but retargeted to a drawn man's face and a drawn frog's face. The last image shows two users communicating through their avatars.