Conveying Emotions through Facially Animated Avatars in Networked Virtual Environments

Fabian Di Fiore, Peter Quax, Cedric Vanaken,
Wim Lamotte, and Frank Van Reeth

Hasselt University - tUL - IBBT
Expertise Centre for Digital Media
Wetenschapspark 2
BE-3590 Diepenbeek
Belgium
{fabian.difiore, peter.quax, cedric.vanaken,
wim.lamotte, frank.vanreeth}@uhasselt.be
http://www.edm.uhasselt.be

Abstract. In this paper, our objective is to facilitate the way in which emotion is conveyed through avatars in virtual environments. The established way of achieving this includes the end-user having to manually select his/her emotional state through a text base interface (using emoticons and/or keywords) and applying these pre-defined emotional states on avatars. In contrast to this rather trivial solution, we envisage a system that enables automatic extraction of emotion-related metadata from a video stream, most often originating from a webcam. Contrary to the seemingly trivial solution of sending entire video streams — which is an optimal solution but often prohibitive in terms of bandwidth usage — this metadata extraction process enables the system to be deployed in large-scale environments, as the bandwidth required for the communication channel is severely limited.

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

1 Introduction

Motivation. Conveying an 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 emotional data is clearly required.

Contribution. The main contribution of this work is to extract the emotion-related metadata from real-time video streams — most often captured through a webcam pointed towards the user's face — and applying these onto a stylised representation of an avatar. To realise the goals stated in the motivation, we developed 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. Another contribution to the research already carried out is the integration of these types of information in (large-scale) 3D environments.

Approach. Technically, the challenge is to extract precisely the right part of information from a sequence of input frames required for conveying the emotional information. Whilst just transmitting the entire sequence of video frames would be an intuitive way of solving the problem, this is often prohibitive in terms of bandwidth consumption, especially when dealing with 3D environments containing large numbers of users. Therefore, the emotional data is extracted using face and feature extraction algorithms, and only the required parameters (in the form of a set of MPEG-4 feature coordinates) are transmitted over the network. At the receiving side, the parameters are used to animate the avatar representing the originating side.

Paper Organisation. This paper is organised as follows. We start with an overview of related work in the field including realistic approaches, techniques adhering to 2D, approaches which explicitly exploit 3D geometries and some pointers to related work in the field of networked virtual environments (Section 2). Subsequently, the different components making up our system are described in detail in Section 3. We conclude this paper with some clarifying results and our conclusions.

2 Related Work

In this section we look at some existing work that is related to the topic in question, both from the viewpoints of stylised animation and immersive communication in networked virtual environments.

2.1 Facial Animation

Towards Realism. Starting with Parke [1], many researchers have explored the field of realistic facial modelling and animation. For the modelling part this

has led to the development of diverse techniques including physics based muscle modelling, the use of free-form deformations, and the use of spline muscle models. For the animation part, the complexity of creating life-like character animations led to performance-driven approaches such as motion capturing and motion retargeting.

We limit this discussion to published work employing some of the mentioned techniques targeted at modelling and/or animating 2D faces. Fidaleo et al. presented a facial animation framework based on a set of Co-articulation Regions (CR) for the control of 2D animated characters [2]. CRs are parameterised by muscle actuations and are abstracted to high-level descriptions of facial expression. Bregler et al. use capturing and retargeting techniques to track motion from traditionally animated cartoons and retarget it onto new 2D drawings [3]. That way, by using animation as the source, similar-looking new animations can be generated.

Although the described techniques are promising and deliver very appealing results, major issues can be identified when targeted to avatars. In the animation stage, they don't offer much freedom of exaggeration (e.g., extensive 3D modelling) whereas the modelling stage implicates a lot of tedious and cumbersome work for the animator (e.g., placing physical markers).

Sticking to 2D. In 1996, Kristinn Thórisson described a dedicated facial animation system, *'ToonFace'*, that uses a simple scheme (a face gets divided into seven main features) for generating facial animation [4]. Ruttkay and Noot discuss *'CharToon'* which is an interactive system to design and animate 2D cartoon faces [5]. Despite its wide range of potential applications (faces on the web, games for kids, ...) a major drawback compared to our approach is that transformations outside the drawing plane are not supported.

Towards 3D. Recently popular, non-photorealistic rendering (NPR) [6,7] techniques (in particular, 'Toon Rendering') are used to automatically generate stylised cartoon renderings. Starting from 3D geometrical models, NPR techniques can generate stylised cartoon renderings depicting outlines with the correct distortions and occlusions. In order to introduce more concepts of 2D animation, Paul Rademacher presented a view-dependent model wherein a 3D model changes shape based on the direction it is viewed from [8].

Starting from 3D has the advantage of the possibility to automatically create extreme frames but at the cost of heavy modelling and, in addition, the results suffer from being too '3D-ish' when one pursues the typical liveliness of 2D animation.

2.2 Immersive Communication

Communication between users in a virtual environment was traditionally done through the use of text-based chat, or, in most recent examples, through real-time VOIP sessions. Obviously, the application of video communication in this

context would be an ideal solution, as the immersion in the 3D world remains at an optimal level — not requiring the user to be distracted by the use of cumbersome input devices such as a keyboard.

Various ways to integrate video into an NVE-like application can be envisioned. For example, in [9], a video recording is made of a person turning 360 degrees in front of a camera setup. Each time the avatar needs to be displayed in virtual reality, a set of two images is selected from the obtained sequence. The use of two distinct images is needed for stereo visualisation in a CAVE environment. Of course, the images being displayed are static, and only their position in the scene is subject to change.

The authors of [10] also use an immersive display technology, and use the positional data gathered from an electromagnetic tracking device to place the avatar in virtual space. Stereo-image video cameras are placed in the corners of the immersive display setup to capture moving images of the user together with depth information, in sync with the captured motion information. All data is subsequently transmitted to the other clients, which choose those frames from the sequence that were captured with the stereo camera that matches best with the position of the avatar from their viewpoint. While the system yields good results, it is only applicable in small-scale deployment due to high deployment costs and bandwidth issues, as data from every camera has to be transmitted to each client.

In [11], (full body) video captured from a camera is applied to a three-dimensional mesh of a human figure. Some enhancements are made to provide additional features beside pure video (comments and gestures). The system described is used mainly for guiding users through vast virtual spaces. We should point out here that this system can only be used off-line (i.e. with pre-recorded sequences).

Finally, [12] uses an off-line reconstructed head model to project real-time video onto. Again, the setup used is an immersive display setup and obtains positional data from tracking devices. Segmentation of the video images is facilitated as only information on the user's head is relevant. As the system was designed for use in a controlled local area network environment, no encoding and/or decoding of video sequences was needed, optimising the quality obtained. Other examples of related work are presented in [13,14,15].

Simplified implementations of similar technology often come packaged with webcams, such as those made by Logitech. It is important to note that these types of software only work with specific hardware and often use simplified methods to extract facial information (e.g., histogram- or colour-based methods). Also, there is no provisioning for transmission of the relevant feature data in any appropriate standard.

3 Our Approach

The established ways of conveying emotions between users in virtual environments vary from just transmitting the entire sequence of video frames, over

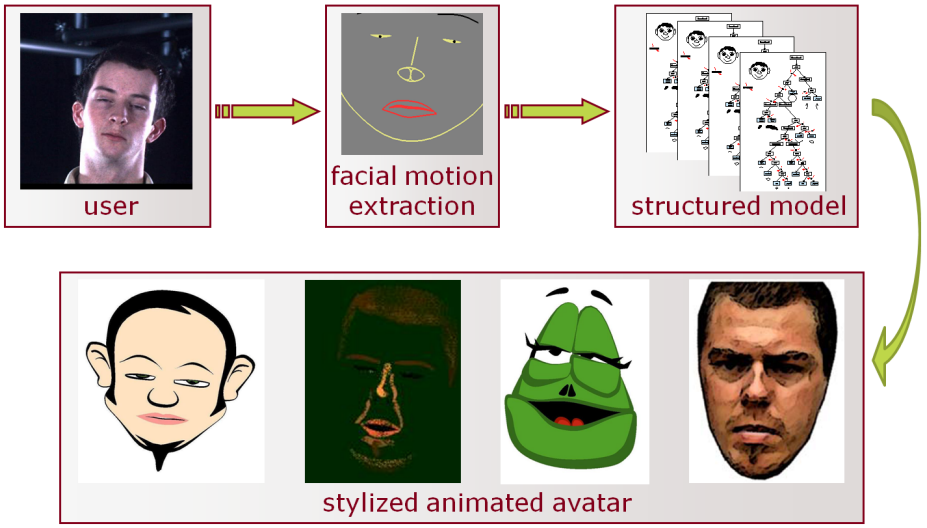


Fig. 1. Overview of the different components of creating and animating an avatar.

performance-driven animation, to manually select one's emotional state. This is often prohibitive either in terms of bandwidth consumption or in terms of manual input.

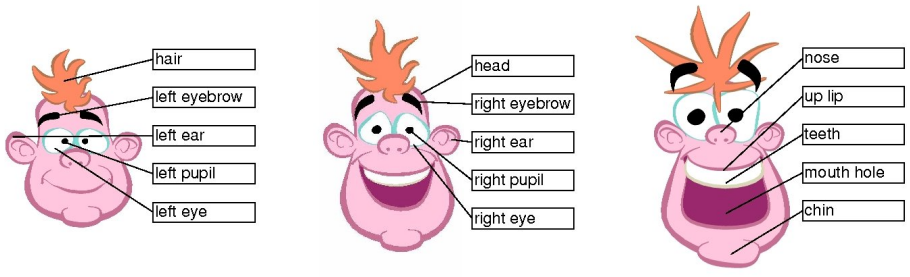
In our approach (Figure 1) the emotional data is extracted automatically using face and feature extraction algorithms, and only the required parameters (in the form of a set of MPEG-4 feature coordinates) are transmitted over the network. At the receiving side, the parameters are used to animate the avatar representing the originating side. 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. Another contribution to the research already carried out is the integration of these types of information in (large-scale) 3D environments.

3.1 Avatar Creation

Modelling Extreme Poses of Drawn Facial Parts. 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 [16]. 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 [17,18,19]. 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).



(a) (b) (c)

Fig. 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 Extreme Poses of Real Footage. 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 mesh structures over certain image parts that contain interesting information. In fact these meshes can be grouped/layered together in the usual way such as the Hierarchical Display Model (HDM). First, one or more initial meshes are created (using subdivision surfaces) for only one image, corresponding to one extreme frame. Then, other

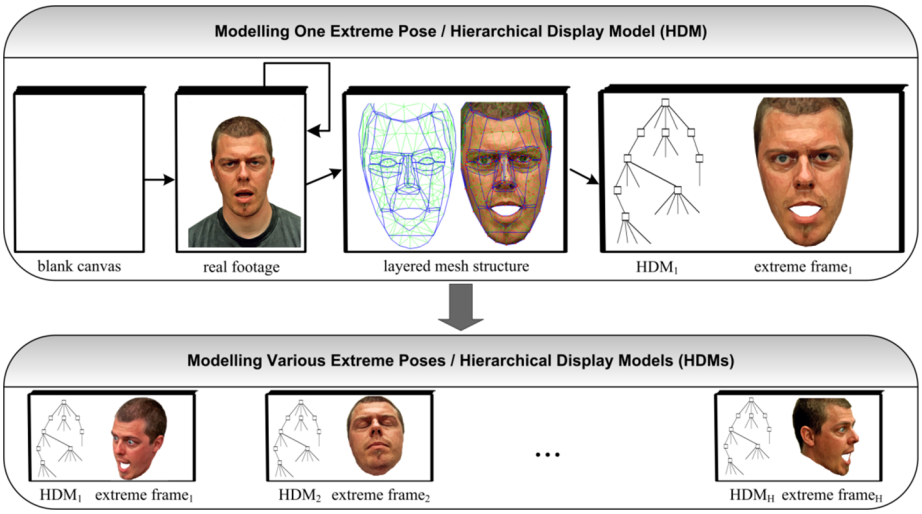


Fig. 3. Overview of the different components of creating an avatar by incorporating scanned drawings or real images depicting extreme poses.

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).

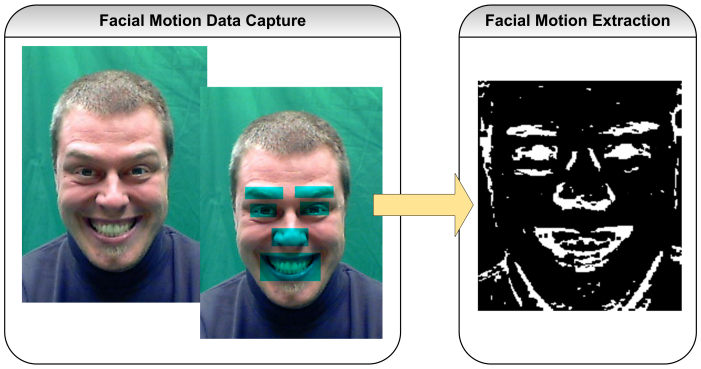


Fig. 4. Overview of the process of facial motion data capture and extraction.

More in detail, data describing the movement of facial components is extracted [20] and made available on a multi-level basis according to the MPEG-4 characterisation [21]: (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). We define these groups as Facial Expression Channels (FECs).

These are analogous to the captured MPEG-4 Facial Expressions and can be considered as groupings of FAPS expressing a specific emotion. 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. Note that each desired FAP region only has to be defined once for one of the extreme frames. The selection automatically gets propagated to the other extreme frames.

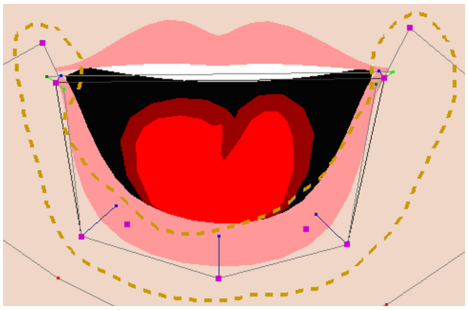


Fig. 5. Example of a user defined FAP region.

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 [22] for an in-depth explanation of the animation system.

3.4 Visualisation of Results in 3D Environments

While we won't go into great detail on the network architecture behind the networked virtual environment that is used to test the results, we will mention some of the features that show that the integration of this type of information can be done with relative ease. The networked virtual environment being used is based on a client/server architecture, with an intermediate layer of 'proxy' servers that are employed to channel the data flows and which mitigate the need

for multiple (persistent) connections between a client and the servers responsible for state management (here referred to as the 'logic' servers). As the architecture is already designed with the ability to efficiently channel all positional data related to avatars (as is true for all NVE architectures), it becomes a trivial extension to include the emotional data that is generated by our system. In effect, the main difference between the two types of information is in their update rate (which is much higher for positional updates) and their relative importance. While the data required for conveying emotions can be considered non-essential, the same is not true (to an extent) for positional data.

4 Results

Figure 6 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.

Discussion. All results are driven by real-time gathered facial motion data and have been evaluated visually by the authors. Certainly, the degree of resemblance depends on the quality (i.e. lifelike modelling versus exaggerated modelling) and the size (i.e. number of view-dependent or emotional extreme frames) of the created emotion space. However, in general we can conclude that there is a clear resemblance between the features of the user's emotional expressions and the final animated output.

5 Conclusions

In this paper we have presented a combination of existing technology allowing users to convey emotions through their avatars in a 3D virtual environment, without major impact on the bandwidth requirements. This is achieved by capturing real-time video using off-the-shelf webcam hardware, and analysing the resulting data flows to extract the facial features important for (human) emotion recognition. Instead of using an actual recognition process, we determine the important feature coordinates and send these to the receiving side, where they can

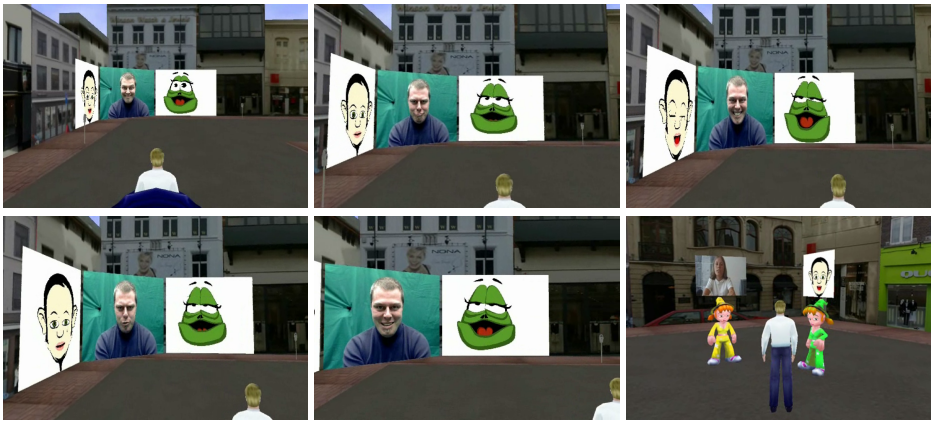


Fig. 6. Snapshots of a generic scene consisting of 3 billboards. The first billboard visualises the input data whereas the second and third depict the emotion conveyed by the first billboard but retargeted to a drawn man's face and a drawn frog's face. The last image shows two users communicating through their avatars.

be 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. The results have been shown to be applicable in a generic 3D environment.

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 and Jeroen Dierckx for their help.

References

1. Frederic I. Parke. Computer generated animation of faces. In *Proceedings of ACM National Conference*, pages 451–457, 1972.
2. 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.
3. 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.
4. 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.

5. Zsófia Ruttkay and Han Noot. Animated chartoon faces. *NPAR2000: Symposium on Non-Photorealistic Animation and Rendering*, pages 91–100, June 2000.
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. Paul Rademacher. View-dependent geometry. In Alyn Rockwood, editor, *Proceedings of SIGGRAPH*, pages 439–446, Los Angeles, 1999. ACM, Addison Wesley Longman.
9. 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.
10. 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.
11. 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.
12. Vivek Rajan, Satheesh Subramanian, Damin Keenan, Andrew Johnson, Daniel Sandin, and Thomas Defanti. A realistic video avatar system for networked virtual environments. In *Proceedings of IPT 2002*, Orlando, FL, 2002.
13. Zicheng Liu, Zhengyou Zhang, Chuck Jacobs, and Michael Cohen. Rapid modeling of animated faces from video. *The Journal of Visualization and Computer Animation*, 12(4):227–240, 2001.
14. R.S. Wang and Y. Wang. Facial feature extraction and tracking in video sequences. In *IEEE International Workshop on Multimedia Signal Processing*, pages 223–238, 1997.
15. Peter Quax, Chris Flerackers, Tom Jehaes, and Wim Lamotte. Scalable transmission of avatar video streams in virtual environments. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2004.
16. Fabian Di Fiore, Philip Schaeken, Koen Elens, and Frank Van Reeth. Automatic in-betweening in computer assisted animation by exploiting 2.5D modelling techniques. In *Proceedings of Computer Animation (CA2001)*, pages 192–200, November 2001.
17. Preston Blair. *Cartoon Animation*. Walter Foster Publishing Inc., ISBN: 1-56010-084-2, 1994.
18. Ronen Barzel. Faking dynamics of ropes and springs. *IEEE Computer Graphics and Applications*, 17:31–39, 1997.
19. Richard Williams. *The Animator's Survival Kit*. Faber and Faber Limited, ISBN: 0-571-20228-4, 3 Queen Square London WC1N 3AU, 2001.
20. 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.
21. Igor S. Pandzic and Robert Forchheimer. *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. John Wiley & Sons, ISBN: 0-470-84465-5, 2002.
22. 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.