Multi-level Performance-driven Stylised Facial Animation

Fabian Di Fiore Frank Van Reeth
Hasselt University
Expertise Center for Digital Media
Wetenschapspark, 2
B-3590 Diepenbeek (Belgium)
e-mail: {fabian.difiore, frank.vanreeth}@uhasselt.be

Abstract

In this paper, our objective is to assist a graphical artist throughout the creation of stylised facial animations that exhibit characters with a hand-drawn look and resemble real life without mimicking it. To this end, we present a hybrid approach combining benefits from performance-driven facial animation and user-controlled 2D modelling and animation techniques. Once we capture the movement and timing of facial components, 'multi-level facial motion descriptors' are extracted according to MPEG-4 characterisation. Within the animation system these motion descriptors are mapped onto a set of artist-drawn facial channels where the parameters for each channel control the degree of caricature applied to the movement allowing flexible caricaturing from a single set of movement data.

1 Introduction

It takes a talented artist to bring a traditional drawn animation character's face to life. Existing animation systems are still too limited to properly assist an artist in creating traditional facial animations. They either aim at creating very precise life-like character animations, or they are very dedicated facial animation systems.

Our objective is to create stylised facial animations that exhibit characters with a hand-drawn look and resemble real life.

To establish these goals we developed a hybrid approach combining benefits of 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 facial components of an actor performing the animation. On the other hand, we opt for a structured 2D methodology as the face to which the captured facial movements are applied will be drawn by an artist.

2 Related Work

In this section we look at existing techniques starting from realism, over 2D animation systems to exploiting 3D geometries.

2.1 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 discussed techniques targeted at modelling and/or animating 2D animations. 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, akin looking new animations can be generated.

Although the described techniques are promising and deliver very appealing results, major issues can be identified. In the animation stage, they don't offer much freedom of exaggeration whereas the modelling stage implicates a lot of tedious and cumbersome work for the animator.

2.2 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]. The author succeeded in attaining her goal: to take a simpler, more artistic approach. However, one almost always ends up with similar looking animations. 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.

2.3 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].

Despite the automatic generation when turning to 3D, heavy modelling and animation of 3D objects are involved and, in addition, the results suffer from being too '3D-ish' since the underlying geometry is rendered too accurate missing the typical liveliness of 2D animation.

3 Our Approach

The novelty of our approach lies in how we combine benefits from performance-driven facial animation with user-controlled 2D modelling and animation techniques. On the one hand, performance-driven facial animation is employed to extract the movement and timing of facial components in order to drive hand-drawn faces. On the other hand, as the face to which the captured facial movements are applied will be derived from an artist-drawn animation, we employ a structured 2D methodology which clearly distinguishes between a modelling and a separate animation phase. The latter is similar to the 3D animation process and has been proven to be very useful for the purpose of creating convincing 3D-like animations starting from pure 2D drawings [9] while preserving the artist's creativeness.

Figure 1 gives an overview of the different components of our approach. These are explained into detail in the following subsections.

3.1 Facial Motion Data Capture

Facial motion data is captured from an acted performance employing a 3D Dynamic Capture System (3D-DCS) which facilitates the capture of 3D models of human faces [10]. The data from the 3D-DCS is in the form of a sequence of VRML frames, each of which contains an entire 3D model of the imaged head. The markup sequence step determines the underlying facial movements of the imaged head using a set of facial features which are located in each of the frames.

3.2 Facial Motion Extraction

In this part, data describing the movement of facial components is extracted [11] from the captured facial motion data. The extracted motion is made available on a multi-level basis according to the MPEG-4 characterisation [12]: (i, low

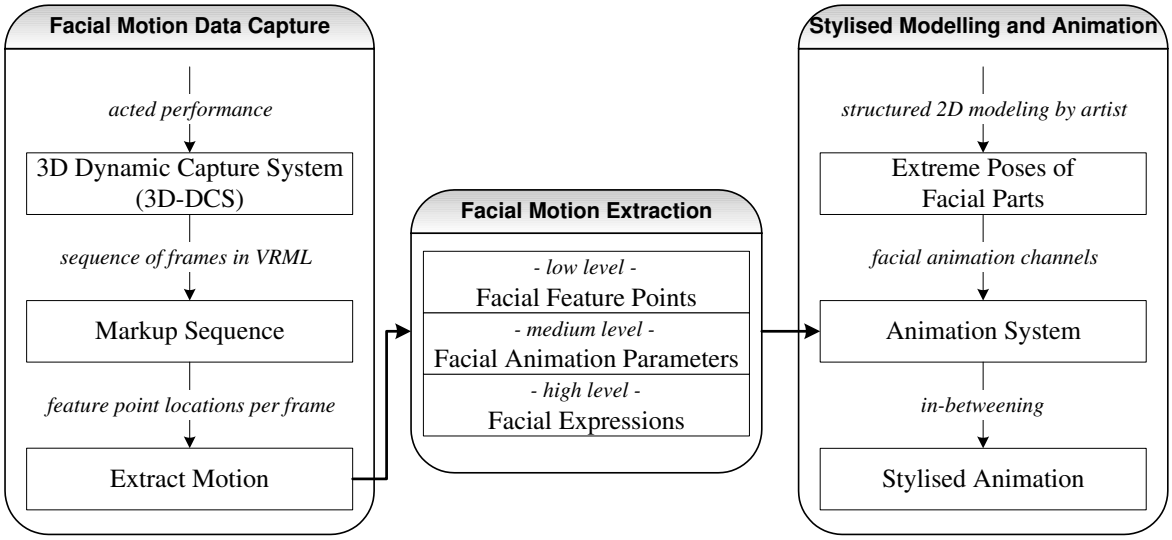


Figure 1: Overview of the different components of our approach.

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). FAPs describe the movements of features on the face in terms of units defined as a set of specific facial measurements, the Facial Animation Parameter Units (FAPUs); and (iii, high level) Motion in terms of MPEG-4 Facial Expressions.

3.3 Stylised Modelling and Animation

The focus of this paper lies on the third component of our approach which deals with obtaining artistically drawn faces and animating them. To this end we employ user-controlled structured 2D modelling and animation techniques [9] in combination with automatic in-betweening.

Considering 2D animation from a technical standpoint, two different categories can be distinguished: (i) transformations in a plane parallel to the drawing canvas (the XY plane), and (ii) transformations outside the drawing plane. The former category of transformations is relatively easy to deal with, whereas the latter is the main cause of all the trouble in automating the in-betweening process (i.e. the underlying subproblems of silhouette changes as well as self-occlusion). It is in the latter type of animation where the 3D structure comes into play that is underlying the objects and characters in traditional animation but which is not present in 2D drawings.

To tackle this without introducing too much 3D information, we developed a solution based on structured 2D modelling and animation techniques [9]. This is implemented as a multi-layered system. At level 0, objects are modelled as sets of depth-ordered 2D drawing primitives. Level 1 manages and processes explicit 2D modelling information and is fundamental in realising transformations outside the drawing plane: for each set of 'important' XY-rotations of the object relative to the virtual camera, the animator draws a set of ordered 2D primitives. This is functionally comparable to the extreme frames in traditional animation. Level 2 incorporates 3D information by means of 3D skeletons or approximate 3D objects, while level 3 offers the opportunity to include high-level tools. Multi-level 2D strokes, interpolation techniques and on-the-fly resorting are used to create convincing 3D-like animations starting from pure 2D information.

The following subsections elaborate on the modelling and animation phase.

3.3.1 Modelling Extreme Poses of 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 matter according to a classical Hierarchical Display Model (HDM).

In Section 3.3 we explained that 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 modelled by the animator in order to cover out-of-the-plane animation. These versions can easily be made by altering a duplicate of the original HDM. On top of this, several 'expressive' versions of each facial channel can be modelled covering the range borne in the animator'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).

3.3.2 Animation System

After the extreme poses of the facial parts are modelled, the extracted facial motion data can be applied to animate the drawn animation face. 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 the animator can enforce any arbitrarily control point or user-selected part of the channel 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). This happens in an easy and interactive way 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. 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. 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.

Once the desired mappings have been made, the extracted facial motion data of previous step (Section 3.2) is loaded into the animation system and all keyframes are automatically set, hence, driving the animation. At this point our automatic in-betweening method comes into play generating the desired animation.

4 Results

All examples are driven by external gathered facial motion data (produced by the University of Glasgow [11]) and have been compared visually (using a pre-visualisation tool) with their data input. Our results show that there is a clear resemblance between the motion captured data and the final animated output.

Figure 3 depicts some stills of generated animations. The first row visualises the input data (MPEG-4 FAPs). The second row depicts the same animation but retargetted to a drawn man. For this animation 18 extreme frames were used consisting of 9 versions which are used to cover different views multiplied by 2 emotional versions 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 — there are 66 FAPs defined by MPEG-4. The third row shows some snapshots of the animation sequence retargeted to a 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.

Besides drawing facial channels, our animation system supports also the possibility to create facial channels by incorporating scanned drawings or real images depicting extreme poses. Our approach, however, stays the same

except for the modelling part (see Section 3.3.1) for which we now provided a tool which allows the animator to define a layered mesh structure over certain image parts that contain interesting information. The animator first creates an initial mesh (using subdivision surfaces) for only one version of the facial part. This is shown in Figure 4(a). For each other version (view-dependent or emotional) the user only has to modify a copied instance of the initial mesh (see Figures 4(b–c)). Concerning the animation phase, the animator, for example, still can define FAP regions as described in Section 3.3.2. Then, during the animation, in-between images of these 'real' channels are constructed by warping the meshes imposed on the extreme frames to each other in the same order as defined by the layered structure. Figures 4(d–f) show some generated results. For this example 27 images were used — 9 view-dependent multiplied by 3 emotional versions.

5 Conclusions

In this text we explained the data path from the acting out of a sequence of facial movements in front of a dynamic capture system to their use in a stylised animated sequence.

To attain our goals we developed a hybrid approach combining benefits of 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 facial components. As the face to which the captured facial movements are applied will be derived from an artist-drawn animation, we opted for a structured 2D methodology. Within the animation system MPEG-4 facial motion descriptors are mapped onto a set of drawn facial channels where the parameters for each channel control the degree of caricature applied to the movement allowing flexible caricaturing from a single set of movement data.

The provided solution demonstrates how an animator can remain in full control of employing performance-driven facial animation data to generate stylised animations.

Acknowledgements

We gratefully express our gratitude to the European Fund for Regional Development (ERDF), the Flemish Government and the Flemish Interdisciplinary institute for Broadband Technology (IBBT), which are kindly funding part of the research reported in this paper. Part of the work is also funded by the European research project IST-2001-37116 'CUSTODIEV'.

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.

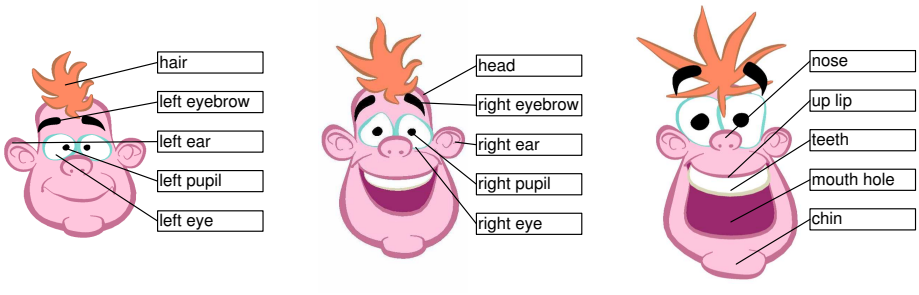
[8] Paul Rademacher. View-dependent geometry. In Alyn Rockwood, editor, *Proceedings of SIGGRAPH*, pages 439–446, Los Angeles, 1999. ACM, Addison Wesley Longman.

[9] 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.

[10] W. P. Cockshott, S. Hoff, and J.-C. Nebel. An experimental 3D digital TV studio. In *IEE Proceedings - Vision, Image & Signal Processing*, 2003.

[11] 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.

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



(a) (b) (c)

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) $e_{exaggerated}$.

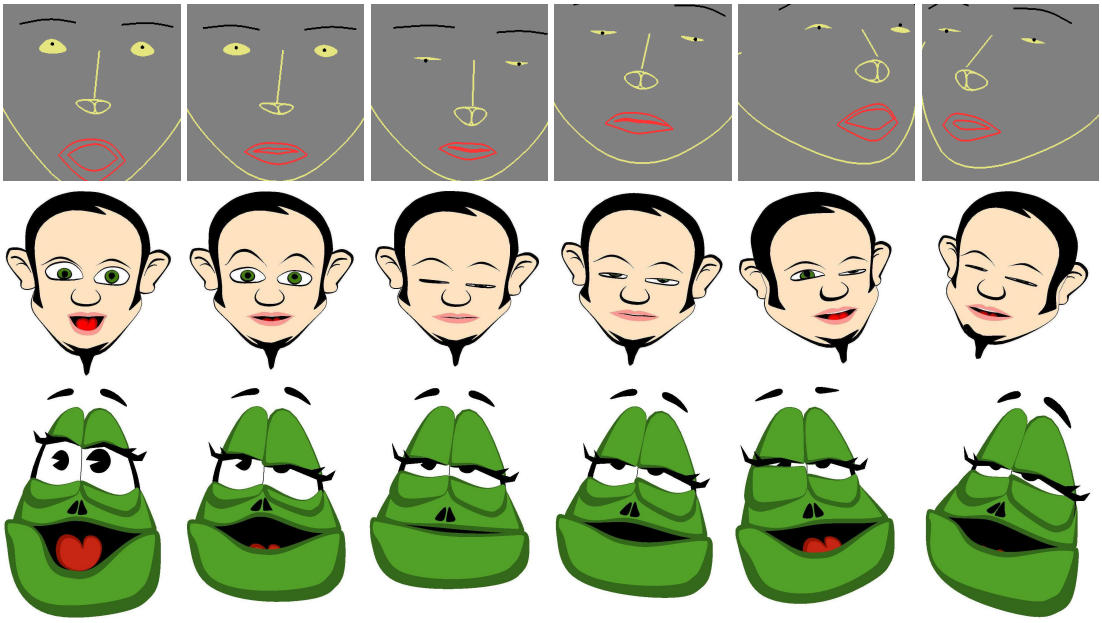
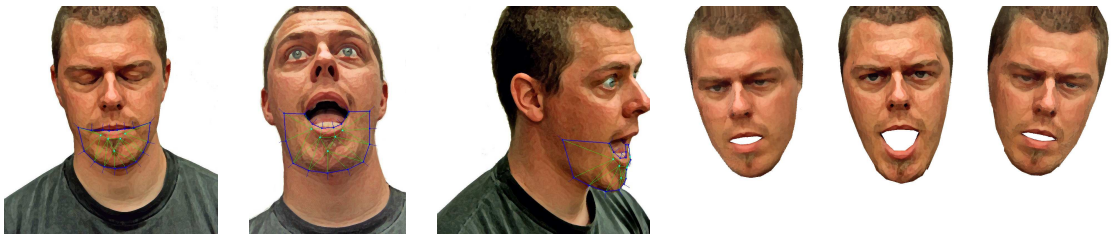


Figure 3: Some snapshots of generated animation sequences. The first row visualises the input data (MPEG-4 FAPs). The second and third row depict the same animation sequence but retargetted to a drawn man's face and a drawn frog's face.



(a) (b) (c) (d) (e) (f)

Figure 4: a–c) Example of 'real' facial channel defined by a layered mesh structure. d–f) Generated results.