

Mimicing 3D Transformations of Emotional Stylised Animation with Minimal 2D Input

Fabian Di Fiore* Frank Van Reeth†

Expertise Centre for Digital Media - Limburg University Centre
Wetenschapspark 2, B-3590 Diepenbeek
Belgium

Abstract

In this paper we introduce a novel approach to assist a graphical artist throughout the creation of traditional facial animation. We focus on eliminating the time-consuming process of drawing all the emotions of a character, which has to be seen from different viewpoints. Furthermore, we aim at preserving the animator's freedom of expressing the artistic style he is bearing in mind.

To establish these goals, we introduce the concept of *facial emotion channels*, of which each represents a facial part expressing an emotion. Furthermore, we present a novel approach through which an emotionally meaningful 2D facial expression from one point of view can be created from a reference expression from another point of view. The provided solution is easy to use and empowers a much quicker cartoon production, without hampering the animation artists' creativity.

CR Categories: I.3.m [Computer Graphics]: Miscellaneous—Computer-Assisted Traditional Animation

Keywords: facial animation, facial expressions, cartoon animation, computer-assisted animation, example-based animation, computer animation

1 Introduction

Traditional animators speak from experience when they say that animating the face is one of the most challenging and rewarding tasks. Under normal circumstances, people immediately can tell by the first look on someone's face under which emotional state the person finds himself.

Although the facial expressions of humans are limited by anatomical constraints, some still manage to pull dozens of faces of which each conveys an emotion. Cartoon characters, on the other hand, lack these constraints and hence are capable of making countless faces of which a substantial part is beyond realism. Drawing all emotions for all characters is without doubt a labour-intensive process. Certainly when the characters have to be seen from multiple viewpoints, most time is spent on drawing facial expressions.

*e-mail: fabian.difiore@luc.ac.be
†e-mail: frank.vanreeth@luc.ac.be

Within the boundaries of our study, it is our goal to assist the animator throughout the time-consuming process of traditional facial animation, especially the individual drawing of all the emotions of a character which has to be seen from different viewpoints. Furthermore, we aim at preserving the animator's freedom of expressing the artistic style he is bearing in mind.

Figure 1 shows an image of the type and look of facial expressions we would like to create and animate. As can be seen, the animator does not mimic reality exactly: emphasis is put on specific expressive details that cannot exist in the real 3D world. In this example the animator wishes to express the astonishment of the wolf by focusing on the orientation of the ears, the tiny pupils and the small mouth.

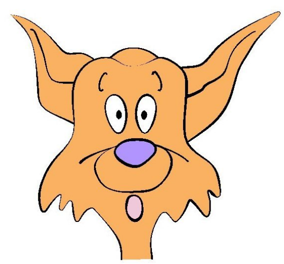

Figure 1: A wolf staring at the camera. The orientation of the ears, the tiny pupils and the small mouth indicate that he is surprised.

To establish these goals, we first introduce the concept of *facial emotion channels* (FECs), of which each can be seen as the representation of a particular facial part expressing a specific emotion. Furthermore, we present a novel approach through which an emotionally meaningful 2D facial expression from one point of view can be created from a reference expression from another point of view. The provided solution is easy to use and empowers a much quicker cartoon production while we do not limit the animation artists in their creativity.

This paper is organised as follows. Section 2 describes previous work in the field and indicates the differences with our philosophy. In Section 3 we first elucidate modelling and animation in 2.5D. After introducing the term *facial emotion channels* (FECs), the automatic generation of these FECs is elaborated on. Section 3.3.2 explains into detail the automatic generation process and Section 4 provides clarifying examples. Finally, we end with our conclusions and ongoing future research (Section 5).

2 Previous Work

In this section we look at and evaluate existing techniques present in realistic facial animation systems, 2D animation systems and systems that exploit 3D models.

2.1 Towards Realism

Starting with [Parke 1972], a lot of research has been carried out into the field of realistic facial modelling and animation.

For the modelling part this has led to the development of interesting techniques including (i) *physics based muscle modelling* [Platt and Badler 1981] which mathematically describes the properties and behaviour of human skin, bone and muscle systems, (ii) the use of *free-form deformations* [Kalra et al. 1992] where the muscles are embedded in an imaginary flexible control volume that can be manipulated by displacing control points and whose deformations are reflected onto the muscle, and (iii) the use of *spline muscle models* such as subdivision surfaces [Catmull and Clark 1978] to support smooth and flexible deformations and to model sharp creases on a surface or discontinuities between surfaces.

For the animation part, the difficulties in creating life-like character animations led to performance driven approaches. These approaches include (i) animation using *motion tracking* [Li et al. 2001] where markers on one's face are continuously tracked and deliver (precise) motion data which can be used for driving specific animation systems, and (ii) animation employing motion *retargetting* [Gleicher 1998] where the animated motion from one character is adapted to another.

We refer the interested reader to [Noh and Neumann 1998] for an extended survey and classification (taxonomy) of facial modelling and animation methods.

As can be seen, there's a wide range of tools and techniques available which aim at creating realistic (facial) animations. Since we are interested in creating lively cartoon animations starting from 2D, we limit this discussion to research that employs some of the discussed techniques (whether adapted or not) to produce 2D animations.

[Rose III et al. 2001] presented an inverse-kinematics methodology which exploits the interpolation of *example*-based motions and positions. The key issue of their system is to allow an artist's influence to play a major role in ensuring that the system always generates plausible results. Starting from a small number of example motions and positions, an infinite number of interpolated motions between and around these examples are generated at high frame rates. This methodology is highly focused on positioning articulated figures and therefore does not lend itself to the generation of facial animation.

[Bregler et al. 2002] use capturing and retargetting techniques to track the motion from traditionally animated cartoons and retarget it onto new 2D drawings. That way, by using animation as the source, similar new animations can be generated. This approach leads to very impressive results, but unfortunately some severe drawbacks prevent it to be used extensively. The retargetting process is very dependent on a good choice of the source and target key-shapes which one has to select and draw manually. The animator has to watch carefully that the chosen key-shapes cover the entire cartoon space (the entire range of possible poses). Furthermore, the creation of the target-key shapes — these shapes replace the source key-shapes — is a very tedious task since each source key-shape requires a target key-shape to be drawn manually.

[Fidaleo and Neumann 2002] presented a facial animation framework based on a set of Co-articulation Regions (CR) for the control of 2D animated characters. CR's are parameterised by muscle actuations and are abstracted to high-level descriptions of facial expression. In practice, video footage (of an actor) is analysed and used to control expressive gestures. The major advantage of this system consists of the independence between the actor and the controlled object. This is maintained by mapping through a single set of controlled parameters. But this implies that each new character requires a neutral face frame *and* an explicit reconstruction sample for each CR state to be defined, Furthermore, the current system is not capable of handling animated characters (for example, a person who's looking around).

The described techniques are promising and deliver very appealing results. However, major issues are present in the systems which are heavily based on realistic input. In the animation stage, they don't offer much freedom of exaggeration and other artistic modifications. Furthermore, the modelling stage involves a lot of tedious and cumbersome work for the animator.

The first issue arises out of working with real motion data; real motion data generates a very realistic look, but in fact that's what we want to escape from when creating cartoon animations. What we need is some kind of *unrealistic* (exaggerated, caricatural, ...) motion data. But that is impossible when working with real actors unless the data is tweaked, which unfortunately is a hard task.

Secondly, our philosophy behind creating animations is not of the kind that an animator has to engage himself in placing markers on someone's face, specifying feature points on images or objects, drawing all extreme poses to cover cartoon space or as an alternative doing extensive 3D modelling, ... and in the end still ending up with an animation that does not resemble what he was bearing in mind.

To summarise, techniques and methods to create realistic facial animation are very advanced and promising but are neither in the animation nor in the modelling stage suitable for the fascinating world of computer assisted traditional facial animation.

2.2 Sticking to 2D

In this section we elaborate on research conducted in the field of pure 2D animation. To be more precise, both the modelling and animation stage integrally take place in 2D.

In 1996, [Thórisson 1996] described a dedicated facial animation system, *'ToonFace'*, that uses a simple scheme for generating facial animation. In this system, a face gets divided into seven main features, each with a specific fixed number of control points of which the position can be fixed, move in one dimension or move in two dimensions. Drawing is done by filled two-dimensional polygons which can have an arbitrary number of vertices. These polygons are differentiated between *free polygons* which cannot be animated, *feature-attached polygons* which are associated with a whole feature and inherit its movements, and *point-attached polygons* which only change when a specific control point is being moved. The author succeeded in attaining their goal, that is to take a simpler, more artistic approach. However, one almost always ends up with similar animations. The use of fixed regions consisting of predefined control points which at their turn have definite degrees of freedom involves a lot of limitations which cannot be overcome by the animator. Other major drawbacks include that the characters always have to look at the camera, and that creating new extreme poses involves a lot of work since all control points have to be manipulated by hand.

Recently, various commercial products have been developed for animating virtual actors, talking heads, virtual announcers, etc. [Ruttkay and Noot 2000] discuss *'CharToon'* which is an interactive system to design and animate 2D cartoon faces. The architecture consists of following three components: (i) the *face editor* (modeler) with which a face can be build up from pre-cooked components, (ii) the *animation editor* to define time curves in order to animate the components of the face, and (iii) the player which generates the frames of the animation. More recently [Ruttkay and Noot 2001], the system is extended to let the animator define and impose constraints on control points in order to ensure a desired animation. The use of *'CharToon'* is similar to *'ToonFace'* but it solves also some of its problems by means of the variable amount of control points and the introduction of skeleton-animated basic components. Despite its wide range of potential applications (faces on the web, games for kids, ...) the system is too specific for creating professional cartoon animations. Two major drawbacks compared to our approach are that transformations outside the drawing plane are not supported, and that all key frames still have to be created manually.

We conclude that 2D facial animation systems are much easier to use than realistic approaches but unfortunately do suffer from too many constraints by which the animator's artistic expression is constrained.

2.3 Towards 3D

This section discusses systems which *turn* to 3D (information) in order to create appealing 2D animations.

Recently popular, non-photorealistic rendering (NPR) [Reynolds n. d.] techniques (in particular, 'Toon Rendering') are used to automatically generate stylised cartoon renderings. Starting from 3D geometrical models, NPR techniques can generate possibly stylised cartoon renderings depicting outlines with the correct distortions and occlusions. Despite the automatic generation, it requires heavy modelling and animation of 3D objects and in any case the results suffer from being too '3D-ish' since the underlying 3D geometry is rendered too accurate.

[Rademacher 1999] presented a view-dependent model wherein a 3D model changes shape based on the direction it is viewed from. The model consists of a base model and a description of the model's exact shape (key deformations) as seen from specific key viewpoints. Given an arbitrary viewpoint the deformations are blended to generate a new, view-specific 3D model. This way, the artistic contributions (key viewpoints) of the animator are always reflected in the generated view-specific model. However, the animator still has to construct the base model and its deformations for each key viewpoint which is undoubtedly time consuming. Moreover, this approach is directed to a type of high-end animation where a 3D look is of prime importance, whereas we want to create animations with 2D details that are too elaborate (or sometimes even impossible) to model in 3D.

To sum up, turning to 3D has the advantage of the possibility to automatically create key frames but at the cost of obtaining animations which are either too 'cold' or too '3D-ish'.

3 Our Approach

As will be clear from the subsequent subsections we opt for a 2.5D methodology which clearly distinguishes a modelling phase and an animation phase. This methodology has been proven to be very useful [Di Fiore et al. 2001] for the purpose of creating convincing 3D-like animations starting from pure 2D information. This approach can be situated in the taxonomy of [Noh and Neumann 1998] under geometry manipulations which exploit interpolation and parameterisation.

When creating a traditional hand-drawn animation [Blair 1994; Williams 2001], a specific procedure is followed. First, rough 'idea' sketches are drawn; secondly, the basic shape of the character is developed; thirdly, features and other details are added, and only afterwards the expressions of the head are drawn.

Consequently we identify two parts in the modelling phase. For the first part, we consider the first three steps which narrow down to the pure creation of animated characters (i.e. creating the body and face outlines).

For the second part, creating facial expressions, we introduce the concept of *facial emotion channels* (FECs) which can be seen as multiple 2.5D animation systems. We'll also discuss how these channels can be generated in a semi-automatic way.

The animation phase in general and the first part of the modelling phase are already handled by our previous work [Di Fiore et al. 2001]. This is briefly discussed in Section 3.1. The second part of the modelling phase is elaborated in Sections 3.2 and 3.3.

3.1 Modelling and Animation in 2.5D

The modelling and animation context of this paper is situated in our prior work [Di Fiore et al. 2001; Di Fiore and Van Reeth 2002a] in which we defined a 2.5D method for automatic in-betweening, which clearly distinguishes a modelling phase and an animation phase.

This is implemented as a multi-layered system starting with basic 2D drawing primitives (in our case sets of attributed 2D curves) at level 0. Level 1 manages and processes explicit 2.5D modelling information and is fundamental in the realisation of transformations outside the drawing plane. Characters and objects are modelled as sets of depth ordered primitives with respect to the x-axis (horizontal) and y-axis (upstanding) rotations. For each set of 'important' x-y-rotations of the object/character relative to the virtual camera, the animator draws a set of ordered curve primitives, functionally comparable to the extreme frames in traditional animation [Blair 1994; Patterson and Willis 1994]. Level 2 incorporates 3D information by means of 3D skeletons or approximate 3D objects and level 3 offers the opportunity to include high-level tools (for example a deformation tool or a sketching tool [Di Fiore and Van Reeth 2002b]).

Multi-level 2D strokes, interpolation techniques and on-the-fly resorting are used to create convincing 3D-like animations starting from pure 2D information. Unlike purely 3D based approaches, our animation still has many lively aspects akin to 2D animation. A rigid 3D look is avoided through varying line thickness and the ability to have subtle outline changes that are either impossible or hard to achieve utilising 3D models.

Our current paper builds further on this 2.5D approach and consequently we will make use of the functionality provided in that system, such as explicit 2.5D modelling and powerful automatic in-betweening.

3.2 Facial Emotion Channels

In this section we introduce the concept of *facial emotion channels* (FEC) which can be seen as the building blocks of any facial expression. Suppose the animator wants to create an animation of a character whose emotions vary from being happy to feeling sad. Clearly, these emotions will be reflected in the character's face. Figures 2(a–c) show some extreme emotions that can occur in this character's cartoon space. Note that for example when the wolf is happy, this is expressed by almost all facial parts, such as the engaging smile, staring eyes and a raise of the eyebrows.

So, instead of modelling a 'complete' happy face at once, we can model every individual part (face outlines, mouth, nose, eyes, eyebrows, ...) of the face independent of the others. That is, for each individual part, we model one neutral version (which depicts no emotion at all) and a set of emotional versions (one version for each emotion that has to be supported). Looking again at figures 2(a–c), for the mouth this can be seen as modelling a neutral, a smiling and a troubled mouth. The other facial parts are modelled in a similar way.

Concerning the animation phase, the animator only has to specify key frames in time by entering parameters using the same methods as described in [Di Fiore et al. 2001]. Afterwards, the automatic in-betweening method comes into play and generates the desired animation.

This gives the animator the opportunity to create countless different facial expressions without having to model each expression manually, contrary to earlier systems [Ruttkay and Noot 2000;

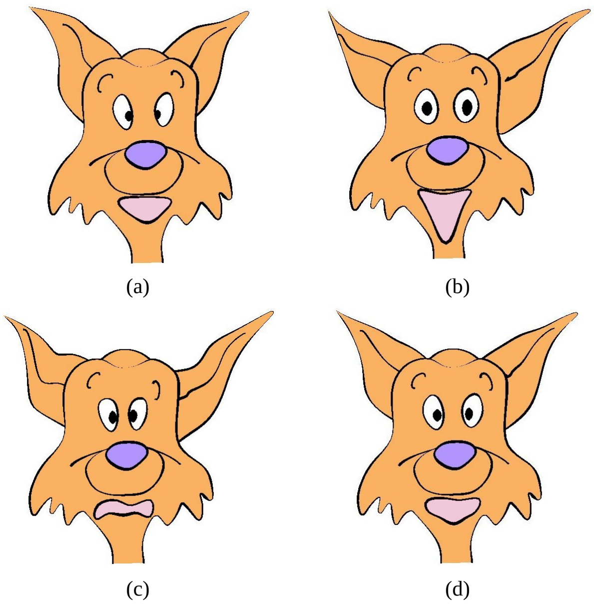

Figure 2: Facial expressions of a wolf depicting different emotions: a) no emotion (neutral), b) laughing, c) troubled and d) mixed emotions (semi-laughing and semi-troubled).

Thórisson 1996] where facial expressions are modelled as one entity and hence are fixed. This is illustrated in figure 2(d) which shows the same wolf but this time with 'mixed emotions': the facial expression is build up from an user-defined interpolation of a happy and a troubled face (semi-laughing and semi-troubled). Note that no modelling took place to create this expression, only some easy understandable parameters had to be chosen by the animator.

One can argue that the modelling of all emotions for every facial part involves a lot of work. It is obvious that the different wanted emotions have to be modelled at least once because that's the only way an animator can convey the artistic images he is bearing in mind. Section 3.3 presents a solution for the case when facial expressions have to be seen from multiple viewpoints.

For a particular viewpoint, each facial part expressing an emotion is called *facial emotion channel* (FEC)[1]. In practice, the animator decides himself how many FECs to create and what each FEC should depict. That way, it is perfectly possible that the animator creates only one FEC to represent both the eyes and the eyebrows instead of creating two FECs where the first one corresponds to the eyes and the second one to the eyebrows.

Also, a FEC does not necessarily consist of only one emotional version. In fact, a FEC can be any N-tuple of versions depicting the same emotion. Each emotional version is identified with a percentage (0% – 100%) indicating the degree of emotion. A zero percentage represents no emotion, 100% maps to the extreme degree of emotion whereas the in-between percentages correspond to emotional versions with an intermediate degree. This is particularly very useful when the animator wants another behaviour than obtained with our automatically in-betweening approach. Suppose we have a FEC depicting a smiling mouth. If the channel consists of only two emotional versions of the mouth (e.g. 0% and 100%), then a 50% smiling mouth would be the average of the neutral (0%) and extreme (100%) version. However, if the animator for some artistic reasons is not satisfied with this result, he can create the intermediate version (in our case 50%) himself, which then has priority over the generated version.

Finally, the user also has the option to combine subsets of FECs into abstract groups, called *emotion groups*. That way, often used facial expressions can be stored and used in a much more intuitive way by only selecting the appropriate *emotion group*. In fact, this functionality is also present in more limited systems, like [Ruttkay and Noot 2000] and [Thórisson 1996].

To summarise, the concept of *facial emotion channels* enables the animator to easily and intuitively create countless different facial expressions without having to model each one by hand.

3.3 Automatic Generation of Facial Extreme Frames

In this section we show how an emotionally meaningful 2D facial expression from one point of view can be created from a reference expression from another point of view.

3.3.1 Overview of Our Approach

In the previous section, we introduced the concept of *facial emotion channels*. Essentially, FECs can be seen as the representation of a particular facial part depicting a specific emotion.

Now, recall that in order to achieve convincing 3D-like animations, our system [Di Fiore et al. 2001] requires the character to be modelled as seen from different viewpoints — about eight viewpoints suffice to fully cover all rotational angles around the upstanding axis. Consequently, each FEC needs to be remodelled for every viewpoint. As a result, the number of FECs to be modelled increases proportional to the number of viewpoints, and therefore also the time spent doing modelling.

We introduce a novel approach which aims at minimising this labour-intensive process. The ideal case would be to develop a tool that meets following requirements; (i) to automatically generate *all* FECs for all viewpoints, and (ii) to take into account the animator's artistic input.

Obviously, in order to properly convey the artistic feelings the animator is bearing in mind (what does a happy face look like?), at least for one viewpoint all FECs should be defined.

Figure 3 shows some facial expressions of a young boy. Besides the neutral expression (a), we recognise a happy, a troubled and a surprised face (b). (In fact, we have a neutral, a happy, a troubled and a surprised FEC for each facial part). Now, when multiple viewpoints need to be supported, the only manual intervention of the animator is *showing* the system how the neutral FECs look like for the other viewpoints. This can easily be done by altering a duplicate of the original neutral FEC or by using techniques described in [Fleerackers 2002] and [Kort 2002]. As a result, the system comes into play and automatically generates all other FECs for each viewpoint. Figure 3(c) shows the *hint* of the animator, whereas the images of figure 3(d) are the automatically generated expressions. We refer to Section 3.3.2 for an in-depth description of our approach.

Note that already for a trivial scene (e.g. five emotions: laughing, troubled, surprised, crying, sad; and 4 viewpoints) twenty modelling interventions can be eliminated, and this for each facial part!

Moreover, as can be deduced from the pictures of figure 3(d) the original artistic input of the animator is preserved throughout the automatic generation of the FECs. Nevertheless, the animator has the possibility to alter the generated results at any time.

In this section we introduced a novel approach that proves to be very helpful in assisting the animator in creating FECs. As a result, the time-consuming aspects of modelling FECs have been minimised, by the automatic generation of most FECs, while respecting the artistic input provided by the animator.

[1] Throughout this text, we use the terms *facial emotion channel*, FEC and channel to denote the same concept.

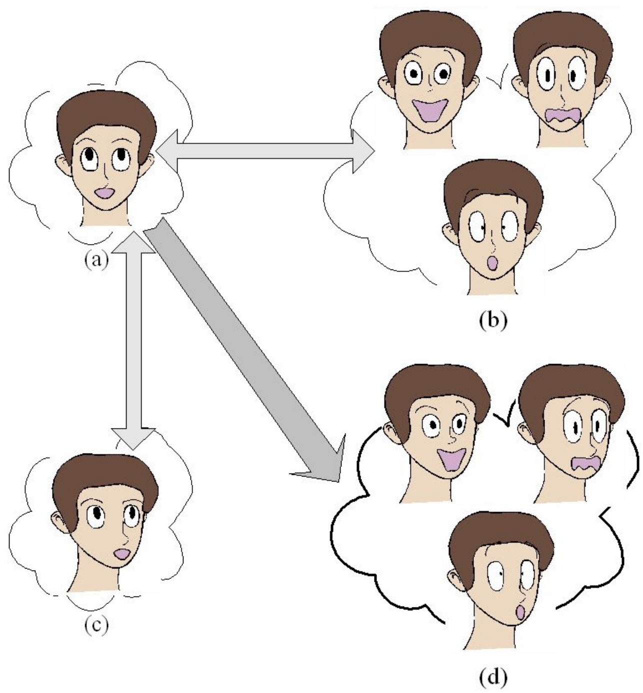

Figure 3: Overview chart of the automatic generation of facial extreme frames. a–b) Neutral and emotional versions modelled by the animator. c) Hint of the animator for a new viewpoint. d) Automatically generated facial expressions.

3.3.2 Technical Approach

Consider again the example in Section 3.3.1 for creating an animation of a young boy. We explain our algorithm on the basis of a single emotion e_j, $j : 1..J$, for one particular facial part F_p, $p : 1..P$, since each pair of an emotion and a facial part gets individually streamed to the algorithm, independently of the others.

The user always starts to model the FECs as seen from a particular viewpoint $vp_k, k : 1..K$, in our case we start with a front facing viewpoint vp_0. Suppose we have two FECs for facial part F_p; one to represent neutral emotion e_n, $F_p^{e_n,vp_0}$, and another, $F_p^{e_j,vp_0}$, to depict the emotional version of F_p. This is illustrated in figures 3(a–b).

Since each FEC is a collection of ordered primitives (in our case curves) [Di Fiore et al. 2001], we write FEC $F_p^{e_n,vp_0}$ as the set of curves $c_l^{e_n,vp_0}$, $l : 1..L$. In a similar way, the emotional FEC $F_p^{e_j,vp_0}$ is built up from the curves $c_l^{e_j,vp_0}$.

At its turn, each curve $c_l^{e_j,vp_0}$ consists of I control points $p_{l,i}^{e_j,vp_0}$, $i : 1..I$. Regarding the fact that each emotional version can be represented as a modified version of the neutral version, each emotional curve also can be seen as a modified version of its neutral equal[2].

In our case these modifications actually are transformations and hence for each control point $p_{l,i}^{e_j,vp_0}$ of the emotional curve $c_l^{e_j,vp_0}$ we store a weighted matrix $M_{l,i}^{e_j}$ which describes the transformation of the i^{th} control point, $p_{l,i}^{e_n,vp_0}$, of the neutral curve, $c_l^{e_n,vp_0}$, to itself.

[2] [Kort 2002] describes a cost function based approach to determine the correct matching of curves.

That way, we express $p_{l,i}^{e_j,vp_0}$ by following equation:

$$p_{l,i}^{e_j,vp_0} = M_{l,i}^{e_j} \cdot p_{l,i}^{e_n,vp_0} \tag{1}$$

Now, in order to automatically create FECs for the other viewpoints vp_k, $k : 1..K$, the user is asked to help the system by doing a demonstration. The user starts with the earlier modelled neutral FEC, $F_p^{e_n,vp_0}$, and transforms it (by using a set of predefined tools or by using techniques described in [Flerackers 2002] and [Kort 2002]) into a new version which represents the same neutral version but corresponding to viewpoint vp_k. We refer to the newly created version as $F_p^{e_n,vp_k}$. This is illustrated in figure 3(c).

For each control point $p_{l,i}^{e_n,vp_k}$ of the curve $c_l^{e_n,vp_k}$ we store a weighted matrix $M_{l,i}^{vp_k}$ which describes the 'viewpoint' transformation of the i^{th} control point, $p_{l,i}^{e_n,vp_0}$, of the curve, $c_l^{e_n,vp_0}$, to this point.

Consequently, we can express $p_{l,i}^{e_n,vp_k}$ by following equation:

$$p_{l,i}^{e_n,vp_k} = M_{l,i}^{vp_k} \cdot p_{l,i}^{e_n,vp_0} \tag{2}$$

At this point, the automatic generation comes into play. In order to find the emotional versions of $F_p^{e_n,vp_k}$ we need to find all curves $c_l^{e_j,vp_k}$. In practice, it narrows down to calculating the control points $p_{l,i}^{e_j,vp_k}$, which are in fact the emotional counterparts of $p_{l,i}^{e_n,vp_k}$.

Analogous to equation 1 we express $p_{l,i}^{e_j,vp_k}$ as follows:

$$p_{l,i}^{e_j,vp_k} = M_{l,i}^{e_j} \cdot p_{l,i}^{e_n,vp_k} \tag{3}$$

Using equation 2 we get:

$$\begin{aligned} p_{l,i}^{e_j,vp_k} &= M_{l,i}^{e_j} \cdot (M_{l,i}^{vp_k} \cdot p_{l,i}^{e_n,vp_0}) \\ &= (M_{l,i}^{e_j} \cdot M_{l,i}^{vp_k}) \cdot p_{l,i}^{e_n,vp_0} \end{aligned}$$

The weighted matrices $M_{l,i}^{e_j}$ and $M_{l,i}^{vp_k}$ are diagonal matrices[3] and so commutative under multiplication:

$$\begin{aligned} p_{l,i}^{e_j,vp_k} &= (M_{l,i}^{vp_k} \cdot M_{l,i}^{e_j}) \cdot p_{l,i}^{e_n,vp_0} && (4) \\ &= M_{l,i}^{vp_k} \cdot (M_{l,i}^{e_j} \cdot p_{l,i}^{e_n,vp_0}) \\ &= M_{l,i}^{vp_k} \cdot p_{l,i}^{e_j,vp_0} && (5) \end{aligned}$$

To summarise, the emotional control points $p_{l,i}^{e_j,vp_k}$ can be seen as (i) an emotional version of the neutral control points of viewpoint vp_k (equation 3), or (ii) a viewpoint vp_k specific version of the emotional control points of viewpoint vp_0 (equation 5), or (iii) the result of a combined operation on the neutral control points of viewpoint vp_0 (equation 4).

Although each solution delivers exactly the same results, we use the third solution since it expresses the new control point as the result of an operation on its neutral counterpart of the starting viewpoint.

So, from now on we calculate each unknown $p_{l,i}^{e_j,vp_k}$ by following equation:

$$p_{l,i}^{e_j,vp_k} = M_{l,i}^{e_j,vp_k} \cdot p_{l,i}^{e_n,vp_0} \tag{6}$$

The automatically generated expressions are pictured in figure 3(d).

[3] Each matrix describes the transformation from one control point to another.

4 Examples

We have used the approach of our paper on some examples. For demonstration reasons, we choose to animate only certain elements of the face (mouth, nose, eyes, eyebrows). Also, the following examples only address motion from head-on to profile. However, our approach is as much suitable when the camera is tilted or when the character turns away from the camera. For example, when the character also has to look up or down, it suffices to create only the new neutral FECs which represent rotations around the horizontal axis. All other emotional FECs will be generated automatically in a similar way as described in Section 3.3.

Figure 4 shows some snapshots of facial expressions of a cute girl seen from different viewpoints. The different facial parts that we modelled are the eyes, the pupils and the mouth. Figures 4(a–d) show the animator's artistic input expressions. We recognise the neutral (a), laughing (b), troubled (c) and surprised (d) expression.

In order to generate emotional versions for the other viewpoints, two new neutral versions (e and i) are provided (one for each viewpoint). This results in the generation of respectively expressions (f–h) and expressions (j–l).

Notice that the outlines of the mouth in figure 4(g) cross the outlines of the face. This is due to the fact that the proportions of the facial parts are intentionally not geometrically correct with respect to 3D.

Figure 5 consists of images depicting some facial expressions of a young boy. We separately modelled the eyes, the pupils, the eyebrows, the nose and the mouth. Figures (a–d) are the source expressions as modelled by the animator whereas figures 5(f–h) show the generated expressions.

As can be derived from our examples, the drawing style from the input images is successively preserved throughout the automatic generation process. This clearly demonstrates the benefits of our concept.

5 Discussion and Future Work

In this paper, we presented a novel approach to assist the animator throughout the time-consuming process of traditional facial animation, especially the individual drawing of all the emotions of a character which has to be seen from different viewpoints. At the same time we need to preserve the freedom of an animator to express the artistic style he is bearing in mind.

We accomplished our goals by first introducing the concept of *facial emotion channels* (FECs), of which each can be seen as the representation of a particular facial part expressing a specific emotion. Instead of modelling a complete face at once, each individual facial part is modelled separately. Hence, creating countless facial expressions is feasible without having to model all of them. Furthermore, we provided a novel approach (and according examples) through which most FECs can be generated automatically. Starting with a minimal input of the animator (all FECs for *one* viewpoint and a neutral FEC for each other viewpoint) *all* emotional FECs for all other viewpoints are automatically generated. As a result, the time-consuming process of creating all FECs is reduced to a minimum while we still preserve the artistic style provided by the animator.

We believe this work is significant for its novel contribution to computer-assisted expressive animation since it attempts to bridge the gap between techniques that either employ purely 2D approaches which constrain the animator's freedom, or employ 3D and realistic approaches which aim at creating high-end animations where a 3D or realistic look is significant.

Currently, we are investigating the incorporation of geometrical constraints. For instance, the lack of these constraints can lead to situations like figure 4(g) where in this case the mouth appears a little outside the girl's face. While at this moment the animator can adjust the results at any time, we are doing research on how to impose geometric constraints which for example avoid the mouth from appearing outside the face. Depending on the defined pay-off functions, this could then be corrected by for instance reorienting the mouth, scaling it up or down, etc.

Acknowledgements

We gratefully express our gratitude to the European Fund for Regional Development and the Flemish Government, which are kindly funding part of the research reported in this paper.

Part of the work is funded by the European research project IST-1999-56412 '3DINCTRAP'. In this context, we would like to express our thanks to the various people in the project providing ideas and for helping with the implementation.

Many thanks to Johan Claes for the valuable paper reviews and to 'Xemi' Morales for his artistic contribution.

Furthermore, we would like to acknowledge ANDROME NV for freely putting available to us their CreaToon plugin SDK.

References

BLAIR, P. 1994. *Cartoon Animation*. Walter Foster Publishing Inc., ISBN: 1–56010–084–2.

BREGLER, C., LOEB, L., CHUANG, E., AND DESHPANDE, H. 2002. Turning to the masters: Motion capturing cartoons. In *Proceedings of SIGGRAPH 2002*, vol. 21(3), ACM, 399–407.

CATMULL, E., AND CLARK, J. 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer–Aided Design 10* (September), 350–355.

DI FIORE, F., AND VAN REETH, F. 2002. Employing approximate 3D models to enrich traditional computer assisted animation. In *Proceedings of Computer Animation 2002*, 183–190.

DI FIORE, F., AND VAN REETH, F. 2002. A multi-level sketching tool for pencil-and-paper animation. In *Sketch Understanding: Papers from the 2002 American Association for artificial Intelligence (AAAI 2002) Spring Symposium. Technical Report SS–02–08*, 32–36.

DI FIORE, F., SCHAEKEN, P., ELENS, K., AND VAN REETH, F. 2001. Automatic in–betweening in computer assisted animation by exploiting 2.5D modelling techniques. In *Proceedings of Computer Animation 2001*, 192–200.

FIDALEO, D., AND NEUMANN, U. 2002. CoArt: Co-articulation Region Analysis for Control of 2D Characters. In *Proceedings of Computer Animation 2002*, 17–22.

FLERACKERS, C. 2002. *New Algorithms and Techniques in 2.5D Animation*. Master's thesis, Transnationale Universiteit Limburg.

GLEICHER, M. 1998. Retargetting motion to new characters. In *Proceedings of SIGGRAPH 1998*, ACM, 33–42.

KALRA, P., MANGILI, A., MAGNENAT-THALMANN, N., AND THALMANN, D. 1992. Simulation of facial muscle actions bases on rational free form deformations. In *Proceedings of Eurographics 1992*, vol. 11(3), 59–69.

KORT, A. 2002. Computer aided inbetweening. *Symposium on Non–Photorealistic Animation and Rendering (NPAR2002)* (June), 125–132.

LI, I.-C., YEH, J.-S., AND OUHYOUNG, M. 2001. Realistic 3D facial animation parameters fom mirror–reflected multi–view video. In *Proceedings of Computer Animation 2001*, 2–11.

NOH, J., AND NEUMANN, U. 1998. A survey of facial modeling and animation techniques. Tech. rep., USC 99-705.

PARKE, F. I. 1972. Computer generated animation of faces. In *Proceedings of ACM National Conference*, 451–457.

PATTERSON, J. W., AND WILLIS, P. J. 1994. Computer assisted animation: 2D or not 2D? *The Computer Journal 37*, 10, 829–839.

PLATT, S., AND BADLER, N. 1981. Animating facial expressions. In *Proceedings of SIGGRAPH 1981*, vol. 15(3), ACM, 245–252.

RADEMACHER, P. 1999. View–dependent geometry. In *Proceedings of SIGGRAPH 1999*, Addison Wesley Longman, Los Angeles, A. Rockwood, Ed., ACM, 439–446.

REYNOLDS, C. Stylized depiction in computer graphics. World Wide Web, `http://www.red3d.com/cwr/npr/`.

ROSE III, C. F., SLOAN, P.-P. J., AND COHEN, M. F. 2001. Artist-directed inverse-kinematics using radial basis function interpolation. In *Proceedings of EUROGRAPHICS 2001*, vol. 20(3), 239–250.

RUTTKAY, Z., AND NOOT, H. 2000. Animated chartoon faces. *NPAR 2000: Symposium on Non–Photorealistic Animation and Rendering* (June), 91–100.

RUTTKAY, Z., AND NOOT, H. 2001. FESINC: Facial Expression Sculpturing with INterval Constraints. In *Proceedings of Autonomous Agents 2001 (AA01). Workshop on Representing, Annotating and Evaluating Non–Verbal and Verbal Communicative Acts to Achieve Contextual Embodied Agents*.

THÓRISSON, K. R. 1996. Toonface: A system for creating and animating interactive cartoon faces. Tech. rep., MIT Media Laboratory, Learning and Common Sense 96–01, April.

WILLIAMS, R. 2001. *The Animator's Survival Kit*. Faber and Faber Limited, ISBN: 0–571–20228–4, 3 Queen Square London WC1N 3AU.

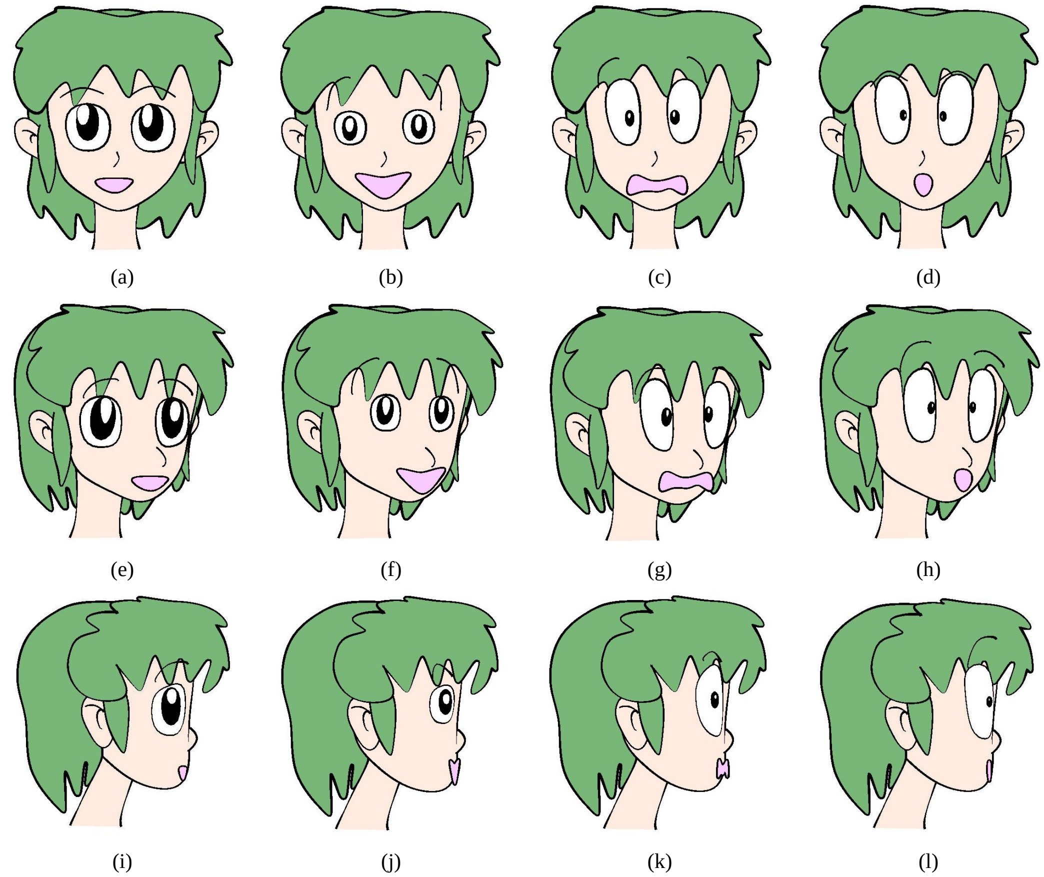

Figure 4: Facial expressions of a cute girl. a) Neutral emotion. b–d) Facial expressions depicting emotions. e, i) Viewpoint specific neutral versions which in combination with (a–d) lead to the generation of respectively (f–h) and (j–l).

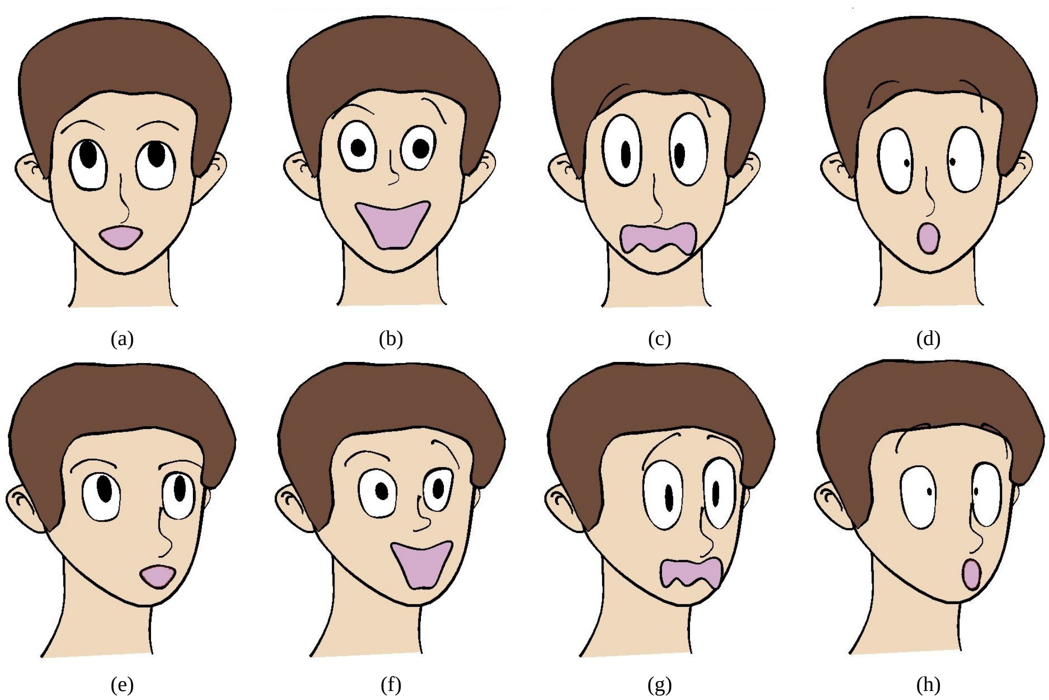

Figure 5: Facial expressions of a boy. a) Neutral version. b–d) Source expressions. e) Viewpoint specific neutral version. f–h) Generated expressions.