An Algebra of Facial Expressions

Aldo Paradiso

Fraunhofer IPSI

D-64293 Darmstadt, Germany

paradiso@ipsi.fraunhofer.de

ABSTRACT

In this paper we introduce an algebraic structure consisting of a set of operators and related properties defined over a set that is a generalization of the MPEG-4's Facial Animation Parameters. This structure, which we call Algebra of Expressions, is then discussed and interpreted. A set of examples of algebraic expressions applied onto a facial model is then presented, showing that the algebra of expressions may be used: a) to describe, manipulate, and generate in a compact way facial expressions; b) as a tool to further study and better understand the role of emotions conveyed by facial expressions and their relationships; c) as a basis to define an animation algorithm for MPEG-4 compliant facial models.

1 Introduction

Humans need to communicate each other, and they perform such a task in a unique way, that employs the contemporary use of human senses and skills. In human-computer interaction (HCI) has emerged the fact that software systems have become new active actors in the communication scenario. Such actors, referred as agents, chatterbots, or characters, have some communication capabilities, some know-how and some reasoning skills. In such a picture embodied conversational agents represent an evolution, towards the personifications of the agents, in terms of appearance, communication skills and simulation of senses.

In figure 1 some different paradigms employed in HCI have been outlined. The sight is the sense most commonly used to communicate, and facial expressions are among the communication channels visually interpreted as well. The face conveys most of the signals sent among humans. Some of them may occur contemporary, such as visemes, emotions, comments, and biological needs. We concentrate our work on such channels (outlined in red or in bold in picture 1).

In our representation, the visual patterns are described and coded with MPEG-4 Facial Animation Parameters that in turn are applied onto a facial model via facial displays. MPEG-4 is an object-based multimedia compression standard, which allows for encoding of different audio-visual objects of a scene independently. In particular, MPEG-4
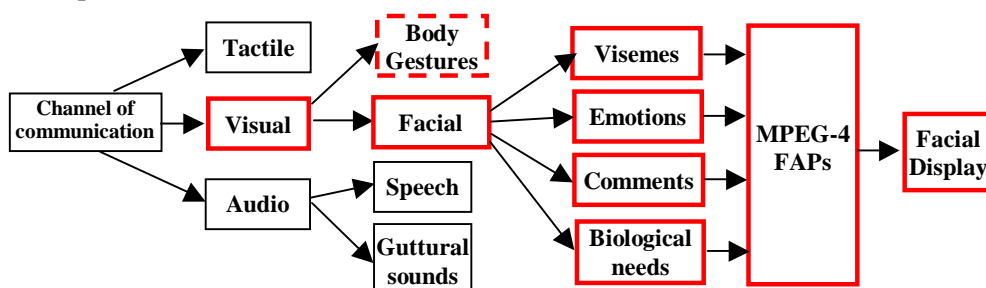

Figure 1 - Communication channels employed in HCI

enables the integration of facial animation with multimedia communications and presentations and allows facial animation over low-bandwidth communication channels. In this approach, each facial expression is coded as an ordered sequence of 68 integer numbers, called Facial Animation Parameters (FAPs). Each of the parameters 3-68 acts on points defined on a synthetic face (feature points), and each point governs the deformation of its surrounding area, resembling muscle movements [8]. This approach is an evolution of the FACS coding system developed by Ekman and colleagues in 1972 [4].

We do not address directly body gestures, but the Algebra of Expressions that we introduce later may be employed to describe and generate such movements as well. In addition, we do not address speech. However, the representation model we introduce below allows speech synchronization, and we implemented a facial animation prototype where a TTS system has been integrated [6].

Several works have been carried out where synthetic characters have been designed to simulate human senses. Among them it is worth noting the complete work of Cassell et al. Cassell's team developed REA [2], a system designed to sense users actions and behaviors during a face-to-face conversation, carrying on reactions and answers based not only on dialogical but even on non-verbal cues. Results achieved in the analysis developed by Cassell's team as well as their terminology has been adopted in this work [1].

Classifications of facial animation techniques and approaches can be found in [3], [13], and [7]. Other works can be found in [19]. Most of them address the possibility of generating facial expressions, and relate them to the ongoing dialogue. However, the adopted solutions lack of generality. Most systems work optimally in

narrow environments, where special facial models are needed, or a limited set of facial expressions has been defined, and tailored *ad-hoc* to the application. In other words, there is the need of an ideal facial animation system that gives the possibility to fully control a set of parameters for both the definition of the facial model and the definition of facial expressions and animations. Also, an ideal parameterization and interface should allow the animator to easily specify any individual face with any speech and expression sequence [14]. The FACS system has been the best current basis for low-level expression parameterization, but it was too abstract from the animator's viewpoint. The development of MPEG-4 facial animation model, with the introduction of the FAPs, has been a great development of the FACS. Still, MPEG-4 is too low-level, and it misses an upper layer that allows defining, mastering, and manipulating facial expressions as a whole. As a matter of facts, the human face can generate around 50.000 distinct facial expressions, which correspond to about 30 semantic distinctions [18]. Clearly the human face is extremely expressive. Is it possible to cover this amount with a minimal set of expressions and a set of transformations on such a set, without manipulating directly the set of facial animation parameters?

In research on facial animation other authors have explored the possibility of combining facial expressions. Ken Perlin has developed a system – *Improv* – where he showed how to make an embodied agent react with responsive facial expression, without using repetitive pre-built animations, and how to mix those facial expressions to simulate shifting moods and attitudes [16]. A convincing demo has been given in [15] where Perlin's Noise function has been employed to provide the facial model with pseudo-natural movements. In MPEG-4 an algorithm to blend together two expressions is suggested; in this approach the concept of excitation (or intensity) of an expression is mixed with the one of combination of two expressions. On our side we extend and distinguish these two concepts, by defining two different operators, introduced in the following section. Tao et al. [17] propose a method to encode human facial movement patterns. They compress FAP streams in a more compact format, and approximate the function that represents the FAP values over time (called by them projection function) with a series function being the sum of simpler functions (called humps). The composition of hump functions may therefore approximate with arbitrary precision the projection function by adding more terms.

Other authors describe what they call "intensity of expressions" even if in most cases they do not clarify what "intensity" means: whether it refers to muscle tension or to human-perceived actions, or other. Won-Sook Lee and al. [20] describe dynamics of facial expressions in terms of three different types of envelopes. They also define an action blending technique in order to avoid discontinuities between facial movements when different expressions and visemes are dynamically concatenated in a unique animation flow. Neumann [9] proposes a process called *expression cloning* that provides a new alternative for creating facial animations for character models. Their method transfers each vertex motion vector from a source face model to a target model having possibly different geometric proportions and mesh structure.

As an alternative approach we introduce an algebraic structure consisting of a set of operators and related properties defined over a set that is a generalization of the MPEG-4's FAPs. In other words we pursue the possibility of creating new facial expressions by acting on instances of previous expressions, and we do that in a formal way, by first of all developing an algebra where a set of operators is defined on a domain modeled according to the definition of MPEG-4 FAP. Secondly, we discuss how to relate the formal construct to a world populated by "real" facial expressions.

In the following sections we introduce an algebraic structure where an initial set (of facial displays) is defined, together with a set of operators. We will then demonstrate a set of properties hold by the operators and derive considerations about possible interpretations of such a structure. Finally we will show some example of operations implemented in the Tinky system [10] and performed on a facial model.

2   Definition of the Algebra of Expressions

In [12] the authors have introduced the concept of *motivators*, which represent sets of parameters usually identifying emotions, attitudes, or any mental or physical state that contributes to generate facial displays. In the same work a preliminary development of the Algebra of Expressions (AoE) has been shown, which is now extended in this section.

2.1   Operators

As stated before, we introduce a set of operators that allow us to manipulate FAP streams in order to generate new facial expressions, based on a set of initial expressions.

2.1.1 Facial states

A Facial state S is defined as a set of 66 integer numbers $(s_3, ..., s_{68})$ where, to each element s_i $(i=3, ..., 68)$ apply respectively the constraints defined in the MPEG-4 Facial Animation Parameter definitions table.[1] These constraints impose to some of the elements to be ≥ 0.

Domain: We denote the domain, i.e. the space of all the facial states, with the symbol $\mathbb{S}$. Such a domain is a subset of the Cartesian product of the integer set, $\mathbb{Z}^{66}$. That is, there exists a family of sets $S_3, S_4, ..., S_{68}$ such that

$$\mathbb{S} = S_3 \times S_4 \times ... \times S_{68}$$

All the sets S_i $(i=3, ..., 68)$ are subject to the constraints defined in the MPEG-4 FAP definitions table (see table of FAPs in [MPEG-4]). For example, the element $s_3 \in S_i$, corresponding to FAP 3 (*open_jaw*) may only assume positive values.

2.1.2 Continuity

Each S_i $(i=3, ..., 68)$ is a sub range of the set $\mathbb{Z}$. This means that, for any $m, n \in S_i$, with $m<n$, if exists an integer p such that $m<p<n$, then $p \in S_i$. In other words, all the elements of S_i are contiguous integers. We denote this property as **continuity** (not to be confused with the continuity property of the real numbers). Since we have defined every S_i $(i=3, ..., 68)$ holding the continuity property, we indicate it simply by stating that $\mathbb{S}$ is **continuous**.

2.1.3 Sum

If we have two facial states $S=(s_3, ..., s_{68})$ and $T=(t_3, ..., t_{68})$, with $s_i, t_i \in \mathbb{Z}$ (natural numbers), we define the sum operator $+_v$ as the following facial state:

$$+_v(S, T) = |s_i \cdot v| + |t_i \cdot (1-v)| \qquad\qquad i=3, ..., 68 \text{ and } v \in [0,1] \qquad\qquad (1)$$

If $v=0.5$ this operator will produce a facial state being the mean of the first two ones. Since v is a real number, the products $s_i \cdot v$ and $t_i \cdot (1-v)$ are real number as well. Therefore we used the symbols $|\ |$ to denote the approximation to the closest integer number.

For simplicity we denote $+_v(S, T)$ with $S +_v T$.

Indeed, the parameter $v \in [0,1]$ introduces an infinite family of operators. However, for simplicity, we continue to denote $+_v$ as a single operator. The following properties will be hold by the whole family of operators, because we will prove them for every $v \in [0,1]$.

Closure: The set $\mathbb{S}$ is closed with respect to the sum. This means that we always get facial states that obey to the constraints imposed in the definition of $\mathbb{S}$. In fact, given two facial states $S, T \in \mathbb{S}$ we have $r_i = R = S +_v T = |s_i \cdot v| + |t_i \cdot (1-v)|$ where $i=s_3, ..., s_{68}$. For every index i and every $v \in [0,1]$ we have $s_i \leq r_i \leq t_i$. Since both S and T are continuous, then $R \in \mathbb{S}$.

Commutative property: The sum is not commutative. If we have two facial states S and T, it is easy to prove that $S+_v T \neq T+_v S$. This depends on the value of v, which is a weight defining a degree of influence of one state instead of the other. An exception arises when $v=0.5$, where the sum becomes commutative.

Associative property

The sum holds the associative property. Given three facial states S, T, and U, we have: $(S+_v T)+_\mu U = S+_v (T+_\mu U)$ for any given $v, \mu \in [0,1]$.

The associative property guarantees the fact that we may apply the operator to several states, in different order. However, the order of the operands is relevant because of the lack of commutative property

Because of the associative property, the algebraic structure $(\mathbb{S}, +_v)$ is a semigroup.

Identity: The set $\mathbb{S}$ does not have an identity with respect to the sum. In fact, for each $S \in \mathbb{S}$ it does not exist an element N such that $S+_v N = S$. Indeed, as a trivial case, such an element is S itself. That is to say, for each $S \in \mathbb{S}$, $S +_v S = S$.

2.1.4 Amplifier

An amplifier $w \in \Re$ is a scalar operator so that, if we have a facial state $S=(s_3, ..., s_{68})$, then:

$$S \cdot w = w \cdot S = (w \cdot s_3, w \cdot s_4, ..., w \cdot s_{68}) \qquad\qquad (2)$$

---

[1] We do not consider the indexes 1 and 2, corresponding to FAP 1 and 2, because these FAPs do not represent facial movements, but expressions (and visemes) as a whole, and do not apply to this analysis.

Indeed, the amplifier is a particular case of sum, where the second element is the zero vector $N = (0_3, ..., 0_{68})$. In fact, we have $S +_w N = S \cdot w$. However, for simplicity, we will consider the amplifier as a different operator, because it will be largely employed.

Closure: The set $\mathbb{S}$ is closed with respect to the amplifier operator. This means that we always get facial states that obey to the constraints imposed in the definition of $\mathbb{S}$.

Identity: The identity value for the amplifier is 1. In fact, for each $S \in \mathbb{S}$ we have $S \cdot 1 = 1 \cdot S = S$.

2.1.5 Overlapper

Let us introduce mask vectors $M = (m_3, ..., m_{68})$ where $m_i = 0$ or 1. Mask vectors will serve us to identify partial regions of the complete FAP vector without loosing generality, i.e. we deal with vectors having the same size in all our computations. If we have a facial state $S = (s_3, ..., s_{68})$ the partial region identified by M is $S_M = M^T \cdot S$ where M^T is the transpose of M. Then, if we consider two facial states S, T with masks respectively M_1 and M_2 and priorities p_1 and p_2, $p_1 \neq p_2$, we introduce the overlapping operator Ω so defined:

$$\Omega (S_{M1,p1}, T_{M2,p2}) = (g_i) = \begin{cases} g_i = s_i + t_i & \text{if } s_i \text{ or } t_i = 0 \\ g_i = s_i & \text{if } p_1 > p_2; \ g_i = t_i \text{ otherwise} \end{cases} \tag{3}$$

where $i = 3, ..., 68$.

For simplicity we denote $\Omega (S_{M1,p1}, T_{M2,p2})$ with $S_{M1,p1} \ \Omega \ T_{M2,p2}$ or simply with $S \ \Omega \ T$ if this does not bring any ambiguity. If the priorities are the same, the operation remains undefined. As for the sum, we actually introduced a family of operators, defined both by the variability of mask vectors and the one of priorities. For simplicity we will consider a single operator, meaning the whole family. The properties investigated below will hold for the whole family, if not differently stated.

Closure: The set $\mathbb{S}$ is closed with respect to the overlapper. In fact, if $R = S \ \Omega \ T$, for any $s_i \in R$, we have that $s_i \in S$ or $s_i \in T$, both elements of $\mathbb{S}$; thus R must be a vector of $\mathbb{S}$ as well.

Identity: The set $\mathbb{S}$ does have an identity with respect to the overlapper. The identity is the vector $N = (0_3, ..., 0_{68})$. In fact, for each $S \in \mathbb{S}$ we have:
$$S \ \Omega \ N = N \ \Omega \ S = S$$

Inverse: The set $\mathbb{S}$ does not have inverse elements with respect the overlapper. In other words, for each $S \in \mathbb{S}$, does not exist an element S^{-1} so that $S \ \Omega \ S^{-1} = S^{-1} \ \Omega \ S = N$. The only exception is the identity N.

Commutative property: The overlapper holds the commutative property. Given two facial states S and T, with masks M_1 and M_2, and priorities p_1 and p_2, $p_1 \neq p_2$, we have:
$$S \ \Omega \ T = T \ \Omega \ S$$

Proof: Let $R = S \ \Omega \ T$. If $r_i \in R$, then we distinguish two cases:
Case a) $r_i = s_i + t_i = t_i + s_i$ (i.e. commutative)
Case b) $r_i = s_i$ or $r_i = t_i$ depending on their priority. This result is independent of the position of the operands.
In both cases the commutative property holds.

Associative property: The overlapper holds the associative property. Given three facial states S, T, and U, we have: $(S_{M1,p1} \ \Omega \ T_{M2,p2}) \ \Omega \ U_{M3,p3} = S_{M1,p1} \ \Omega \ (T_{M2,p2} \ \Omega \ U_{M3,p3})$ for any given M_1, M_2, and M_3 and any priority p_1, p_2, and p_3, with $p_1 \neq p_2 \neq p_3$.

Proof: Let us have $s_i \in S$, $t_i \in T$, and $u_i \in U$. For each set of three values s_i, t_i, and u_i their value may be zero or not. If we indicate with v a value $\neq 0$ we distinguish 8 major cases, corresponding to the different combinations of zeroes and values v's:

1) 000	2) 00v	3) 0v0	4) 0vv
5) v00	6) v0v	7) vv0	8) vvv

Then, for each of these combinations, different priority values must be considered. Moreover, for each combination, 6 different cases of priority arise, and the total number of cases is 6x8=48. However, priorities apply only if at least two values are $\neq 0$. Thus cases 1, 2, 3, and 5 may be proved without involving priorities, as follow:
Case 1) $(0 \ \Omega \ 0) \Omega \ 0 = 0 \ \Omega \ 0 = 0 \ \Omega (0 \ \Omega \ 0)$
Case 2) $(0 \ \Omega \ 0) \ \Omega \ v = 0 \ \Omega \ v = v = 0 \ \Omega (0 \ \Omega \ v)$
Case 3) $(0 \ \Omega \ v) \ \Omega \ 0 = v \ \Omega \ 0 = v = 0 \ \Omega (v \ \Omega \ 0)$
Case 5) $(v \ \Omega \ 0) \ \Omega \ 0 = v \ \Omega \ 0 = v = v \ \Omega (0 \ \Omega \ 0)$
The rest of the cases must be considered with priorities. Let be p_1, p_2, and p_3 respectively priorities of S, T, and U. We call v_1, v_2, and v_3 values with priorities p_1, p_2, and p_3.
Case 4a) $p_2 > p_3$ then $(0 \ \Omega \ v_2) \Omega \ v_3 = v_2 \ \Omega \ v_3 = v_2 = 0 \ \Omega (v_2 \ \Omega \ v_3)$

Case 4b) $p_2 \leq p_3$ then $(0\ \Omega\ v_2)\Omega\ v_3 = v_2\ \Omega\ v_3 = v_3 = 0\ \Omega(v_2\ \Omega\ v_3)$
Case 6a) $p_1 > p_3$ then $(v_1\ \Omega\ 0)\Omega\ v_3 = v_1\ \Omega\ v_3 = v_1 = v_1\ \Omega(0\ \Omega\ v_3)$
Case 6b) $p_1 \leq p_3$ then $(v_1\ \Omega\ 0)\Omega\ v_3 = v_1\ \Omega\ v_3 = v_3 = v_1\ \Omega(0\ \Omega\ v_3)$
Case 7a) $p_1 > p_2$ then $(v_1\ \Omega\ v_2)\Omega\ 0 = v_1\ \Omega\ v_2 = v_1 = v_1\ \Omega(v_2\ \Omega\ 0)$
Case 7b) $p_1 \leq p_2$ then $(v_1\ \Omega\ v_2)\Omega\ 0 = v_1\ \Omega\ v_2 = v_2 = v_1\ \Omega(v_2\ \Omega\ 0)$
Case 8) In such a case the result is always determined by the element with higher priority, no matter the way in which the overlapper applies. As example, consider the case $p_1 > p_2$ and $p_1 > p_3$. We have:
$(v_1\ \Omega\ v_2)\Omega\ v_3 = v_1\ \Omega\ v_3 = v_1$. But also $v_1\ \Omega\ (v_2\Omega\ v_3) = v_1\Omega\ v_3$ or $v_1\Omega\ v_2$ that both give v_1.
The associative property guarantees the fact that we may apply the operator to several states, in different order. Both the order of the operations and the one of the elements may be arbitrarily changed, because of the associative and commutative property, respectively.

Because of the associative property, the algebraic structure $(\mathbb{S},\ \Omega)$ is a semigroup. In addition, it is commutative, and has an identity. Therefore it is a commutative monoid.

Distributivity: The operators defined above hold the distributive property as well. In particular the amplifier distributes over the sum, as follows:

Given two facial states $S,\ T \in \mathbb{S}$, we have: $w\cdot(S +_v T) = w\cdot S +_v w\cdot T$ $\qquad v\in[0,1]$

Proof: Given two facial states $S = (s_3,\ ...,\ s_{68})$ and $T = (t_3,\ ...,\ t_{68})$, we have, for each $i=3,\ ...,68$

$$w\cdot(s_i +_v t_i) = w\cdot(s_i\cdot v + t_i\cdot(1-v)) = w\cdot s_i\cdot v + w\cdot t_i\cdot(1-v) = w\cdot S +_v w\cdot T$$

The distributivity of the amplifier over the overlapper is stated as follows:

Given two facial states $S,\ T \in \mathbb{S}$, we have: $w\cdot(S\ \Omega\ T) = w\cdot S\ \Omega\ w\cdot T$

Proof: Let us have $s_i \in S$ and $t_i \in T$, with masks respectively M_1 and M_2 and priorities p_1 and p_2. For each $i=1,\ ...,68$ we have the following cases:

a) s_i or $t_i = 0$ then $w\cdot(S\ \Omega\ T) = w\cdot(s_i\ \Omega\ t_i) = w\cdot s_i$ or $w\cdot t_i = w\cdot S\ \Omega\ w\cdot T$

b) s_i and $t_i \neq 0$, then $w\cdot(S\ \Omega\ T) = w\cdot(s_i\ \Omega\ t_i)$ for each $I=1,\ ...,68$. Suppose $p_1 > p_2$, we have
$w\cdot(s_i\ \Omega\ t_i) = w\cdot s_i = w\cdot S = w\cdot S\ \Omega\ w\cdot T$. With $p_1 \leq p_2$ we have: $w\cdot(s_i\ \Omega\ t_i) = w\cdot t_i = w\cdot T = w\cdot S\ \Omega\ w\cdot T$

However, the overlapper does not distribute over the sum, nor the sum distributes over the overlapper. This means that these two operators may not be applied with an arbitrary order to the facial states; otherwise the result may be, in general, different. In other words, for some A, B, and C $\in \mathbb{S}$ we have:
$A\ \Omega\ (B +_v C) \neq (A\ \Omega\ B) +_v (A\ \Omega\ C)$, and $A +_v (B\ \Omega\ C) \neq (A +_v B)\ \Omega\ (A +_v C)$.

3 Interpretation of the Algebra of Expressions

The operators introduced above define an algebraic structure that is not complex nor special, because it does not introduce a new mathematical concept. What makes sense is its *interpretation* in a different domain. Interpreting the algebra of expressions means establishing a relationship between two domains belonging to different universes. In particular, the relationship we are interested in is an isomorphism. Recognizing, or defining an isomorphism between two domains means that determinate actions produced in one domain reflect precise actions performed into the other domain. In other words, establishing an isomorphism between two domains means adding semantics to both of them. Our aim is to build interpretations that match with the common sense, that are reasonable, and that are supported by scientific argumentations.

3.1.1 Facial Displays

We want to interpret $\mathbb{S}$, the domain of all the facial states defined according to the definition in 2.1.1. An interpreted domain usually resides in some real world. In our case we distinguish at least two different interpretations, according to our common sense and experience. The domains corresponding to the interpretations are the following:

o The **human-like facial expressions**. Elements of $\mathbb{S}$ are facial expressions, which are usually recognizable onto human faces. A facial expression in such a sense is a static picture of the face. Facial movements are not considered here.

o The **cartoon-like facial expressions**. The range of movements allowable to cartoons is larger than for humans. Expressions may be exaggerated, and other movements are possible and acceptable, which are impossible for humans. Each cartoon animator may impose his constraints, depending on his experience, on the acceptance of the spectators, and on the cartoon identity.

In both interpretations the range of allowable values, i.e. instances of $S\in\mathbb{S}$, is limited by the physical (for humans) or pseudo-physical (for cartoons) constraints imposed by muscle limited stretching properties. Therefore we introduce a more limited domain, imposed by these constraints, called set of *facial displays*.

We denote such a set with $\mathbb{E}$. We say that instances of vectors belonging to $\mathbb{E}$, if applied onto an MPEG-4 compliant facial model, produce a reasonable effect. By *reasonable effect* we intend the following:

a) Each vector $E \in \mathbb{E}$ has integer elements e_i ($i=3, \ldots, 68$) assuming a value that describes, or simulates, the stretch value of a facial action, i.e. one or more muscles that produce a visible effect onto the face, according to the notation defined in the MPEG-4 standard.

b) Each vector E represents a FAP stream that, if applied onto a facial model, let humans recognize some type of facial display.

The set $\mathbb{E}$ is a strict subset of $\mathbb{S}$. This assertion is valid because:

a) Every element belonging to $\mathbb{E}$ belongs to $\mathbb{S}$ as well, i.e. facial displays are particular cases of facial states.

b) There is some element belonging to $\mathbb{S}$ that does not belong to $\mathbb{E}$. In fact, there are some facial states that do not correspond to any human-like facial expression. For example, the facial state $S=(s_3, \ldots, s_{68})$, where every $s_i = 0$ except $s_3 = 5000$ corresponds to a FAP stream where the FAP *open_jaw* = 5000. Such a value is too large for any feasible opening of a human lower jaw. For cartoons-like facial displays the range is greater than for humans, however there exist an upper limit for such displays as well.

If we denote with $\mathbb{H}$ the set of human-like facial expressions and with $\mathbb{C}$ the cartoon-like facial expressions, we observe that:

$$\mathbb{H} \subset \mathbb{C} \subset \mathbb{S}$$

For simplicity, we define $\mathbb{E} = \mathbb{H} \cup \mathbb{C}$, the union of human-like and cartoon-like facial displays. If not stated otherwise we will talk about the set of facial displays $\mathbb{E}$.

3.1.2 Continuity in the interpreted domains

We need to verify if the continuity property, defined in paragraph 2.1.2 holds in the interpreted domain. This verification is necessary because the interpreted domains are smaller than the original one.

We begin by observing that facial displays have an anatomical interpretation. In fact, by identifying facial displays with FAP streams defined by the MPEG-4 standard, we derive that they represent some muscle displacement over the face.

A facial display represents, in general, a set of facial muscles during an effort state, in which they elongate (or contract) of a certain value. It is opportune to take into consideration the properties of muscles, and in particular the elongation property. Because of such a property, muscle lengths may assume all values between a range, whose extension is determined by anatomical properties.

Therefore we can assert that the continuity holds also in the interpreted domain $\mathbb{E}$.

3.1.3 Closure in the interpreted domains

Closure of the sum: The output value of the sum is always a value in between of the two given ones. Since $\mathbb{E}$ is continuous, the output values belong to $\mathbb{E}$ as well. Therefore the domain $\mathbb{E}$ is closed with respect to the sum.

Closure of the amplifier: The amplifier is a particular case of sum; therefore $\mathbb{E}$ is closed with respect to the amplifier as well.

Closure of the overlapper: If we have two facial displays S, T $\in \mathbb{E}$, The display $R = S \, \Omega \, T$ belongs to $\mathbb{E}$ as well. This is because for any $r_i \in R$, we have $r_i \in S$ or $r_i \in T$, both elements of $\mathbb{E}$. Thus, in the interpreted domain $\mathbb{E}$ the overlapper is closed.

4 Examples

The operators introduced above have been implemented as methods included in an Application Programming Interface developed in Java. The API has been incorporated as a part of the Tinky and Fanky systems, developed by the author in [10] and [11].

We are now in the position of showing some concrete examples of using the operators introduced above.

The facial model on which the operators have been applied has been employed in both Tinky and Fanky systems. It consists in a VRML-based 3-D cartoon face, with low level of details, but enough to represent facial expressions. Figure 2 shows an example of use of the sum operator over two expressions (anger and smile) applied onto the same face, with *v=0.5*. In the example of figure 3 the amplifier operator has been applied onto a smiling expression, where the initial value has been amplified by *w= 2.0*.

The amplifier may be used to inhibit an expression, of course. Figure 4 shows an angry expression mediated with a value of *w=0.5*. Negative values have been tested as well. Indeed, we found out that such values give interesting results: in general, negative values produce accordingly a *negation* of the expression. The negation of a smiling face seems to be a sad face, the negation of an angry face results in a surprised face. We are currently investigating the effects of providing negative values to the amplifier.
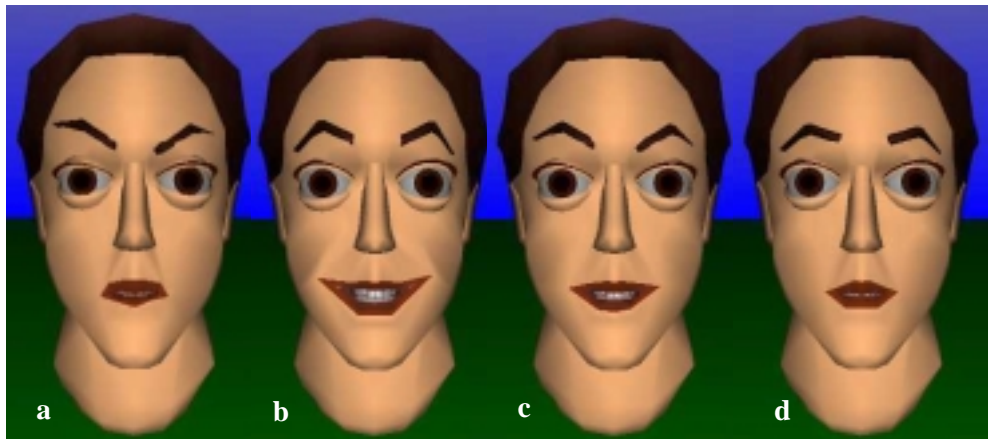
Figure 2 - Merging expressions. In order we have: anger (a), smile (b), their weighted sum (c), and the neutral expression (d).

Figure 5 shows the employment of the overlapper. In (a) a smiling face and in (b) an angry face are shown. The mask associated to (a) covers the mouth area, while the mask associated to (b) covers the eyes and eyebrows areas. Their overlap, according to these masks, is shown in (c). Note that the figure in (c) resembles a surprised expression. This shows how it is possible to derive new expressions from old ones that are semantically different. By using the same expression switching the role of the masks a different overlap is produced, as shown in (d): the mouth is smiling and the eyes are

Figure 3 - Emphasizing a smile

Figure 4- inhibiting an angry expression

angry. Again, this expression holds a semantic different from the previous ones, signaling smartness, arrogance or sense of superiority.

The operators of the algebra of expressions may be combined together, so to arrange more complex algebraic expressions. They can be used to manipulate original displays in order to derive new ones even completely different.

For example, if we overlap a neutral expression *N* with the eyebrows area of an angry expression *A* and then we amplify the result by *w= -1.3* we obtain a resulting expression: $R = -1.3*(N \, \Omega \, A)$. The result is shown in figure 6. The new expression demonstrates slight surprise, or is a conversational signal of interest. Again, note that this expression is semantically very different from the original ones.

Asymmetric expressions may be produced as well by employing the operators above defined.

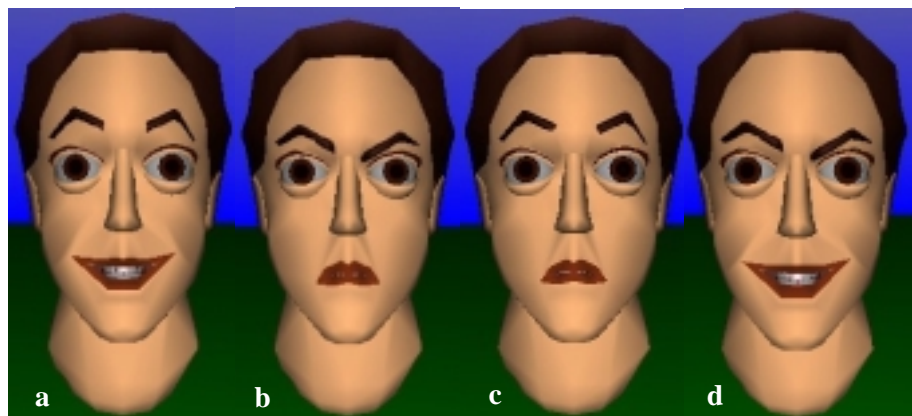A way to produce an asymmetric expression by using the algebra, for

Figure 5 - Overlapping expressions. In order we have: smile (a), angry (b), one overlap (c), and a second different overlap (d).

example, is taking an expression *E*, amplify (or inhibit) it by a value *w*, and then overlap the left (or the right) part of the result with *E*. If we define M_1 and M_2 respectively as left and right masks we have:

$$R = (E \cdot w)_{M1} \; \Omega \; E_{M2}$$

Figure 7 shows an example, where an asymmetric smile is produced. The value of the amplifier operator in this case is *w=2*.

The employment of the overlapper is somehow complicated by the necessity of defining masks. In practice, such masks allow defining patches to be glued together, like in a patchwork. However, it is not straightforward to define masks. In order to make such a task easier, a GUI containing a window with colored labels have

Figure 6 – The outcome of the expression R = -1.3*(N Ω A)

been defined in the Tinky system, that allows to easily define masks by clicking onto check buttons identifying facial areas.

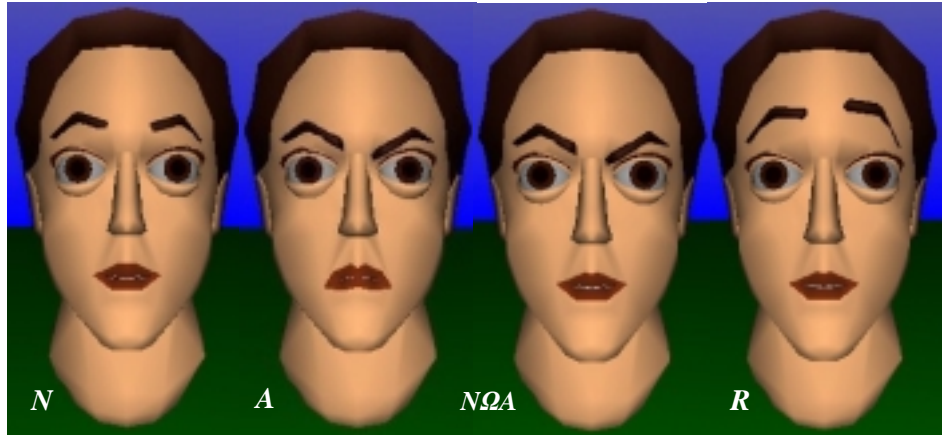By using the operators introduced above it is possible to produce a large and differentiated set of expressions starting from very few ones. A question arises here: is it possible to isolate a basic set of displays sufficient to produce all, or a great amount, of semantically different facial displays?

A contribution in such a direction has been given by the work of Ekman. His research on facial expressions brought to identify the six major expressions universally recognizable [5]. According to his research, other expressions may be considered as derivative ones. It is certainly possible to manipulate such a basic set of expressions with our algebra and derive new ones. However, identifying a basic set of recognizable expressions does not necessarily mean to identify a basic set of vectors of our algebra with which we can extensively produce new displays. In addition, the notion of facial display involves facial clues that do contribute in facial communication, but that are not necessarily facial expressions conveying emotion.
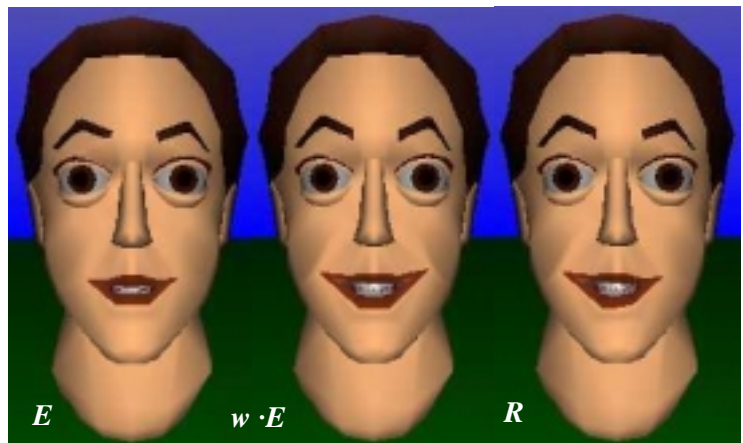

Figure 7 – The outcome of the expression R = (E*2.0) Ω E

5 Conclusions and Future work

The introduction of algebraic expressions built using the AoE operators and acting on an initial set of facial displays defines implicitly a language, where the operators and the displays are the alphabet and the algebraic expressions are the words and sentences. The syntax and the grammar rules are defined by the properties shown in section 2, that provide a criterion to distinguish well-formed words (noun phrases) from wrong ones. Examples of sentences have been given in figure 6 and 7. In figure 6 the sentence has the following meaning: "take only upper part of expression *E* (i.e. eyebrows) and reverse it". The sentence of figure 7 has the meaning: "Make expression *E* asymmetric". In both cases the expression is a variable, which will be given a value in order to finalize the sentence. As a general remark, we can state that this algebra represents a formal definition of the semantics of natural language sentences describing facial expressions.

By building a set of predefined words (i.e. algebraic expressions) of such a grammar it will be possible to implicitly define a large set of facial displays, in a compact way, which is independent of the particular source facial model. Indeed, a very limited set of facial expressions is probably already enough to produce a

virtually unlimited set of expressions that only the practice may better define. This observation may be the basis to define an animation algorithm for MPEG-4 compliant facial models, where animations are produced by a decoder able to interpret the construct of this algebra into sequences of facial expressions.

Future work includes the development of *ad-hoc* formulas (i.e. algebraic expressions) with which we may define and represent transformations of facial expressions, like e.g. asymmetric expressions, negations of expressions, derivative expressions, combinations of emotions and visemes, combinations of comments (in Cassell's meaning) and visemes, etc. We are also engaged in developing an animation algorithm able to exploit this algebra and use its constructs as keyframes of animations. In such a sense an effort of defining an algebra of animations, which has the task of formalizing the structure of such an algorithm, is currently under development as well.

6 References

[1] Cassell, J.; Sullivan, J.; Prevost, S.; Churchill E. Embodied Conversational Agents. The MIT Press, Cambridge, Massachusetts, 2000. ISBN 0-262-03278-3. Pp. 17-21.

[2] Cassell, J.; Sullivan, J.; Prevost, S.; Churchill E. Embodied Conversational Agents. The MIT Press, Cambridge, Massachusetts, 2000. ISBN 0-262-03278-3. Pp. 29-63.

[3] Dendi, Vikram R. A face for a robot: The path to creating a face for a socially interactive robot. Applications to Human computer Interaction. Available from the Internet at the URL: www.its.caltech.edu/~vikram/cs286/report

[4] Ekman, P., Friesen, W. Facial Action Coding System. Consulting psychologists Press, Inc., Palo Alto, CA, 1978.

[5] Elkman, P. The argument and evidence about universals in facial expressions of emotion. In H. Wagner and A. Monstead, editors, Handbook of Social Psychophysiology, pages 143-146. John Wiley, Chichester, 1989.

[6] 3rd INVITE Status Report released by the FhG IPSI to the German Federal Ministry of Education and Research, February 2001.

[7] Jun-Yong Noh: A Survey of Facial Modeling and Animation Techniques.

[8] ISO/IEC IS 14496-2 Visual, 1999 and ISO/IEC IS 14496-1 Systems, 1999.

[9] Neumann et al. Human Face Modeling and Animation. Integrated Media Systems Center University of Southern California 2001 NSF Report.

[10] Paradiso, A.; Nack, F.; Fries G.; Schuhmacher, K. The Design of Expressive Cartoons for the Web – Tinky. Proceedings of ICMCS Conference, June 7-11 1999, Florence (Italy).

[11] Paradiso, A., Zambetta, F., Abbattista, F. Fanky: A Tool for Animating Faces of 3D Agents in: Intelligent Virtual Agents - Third International Workshop, IVA2001, Madrid, Spain, September 2001 Proceedings. Edited by Angelica de Antonio, Ruth Aylett, and Daniel Ballin. Springer-Verlag - Lecture notes in computer sciences; vol. 2190: Lecture notes in artificial intelligence, pp. 242-243.

[12] Paradiso, A., L'Abbate, M. A Model for the Generation and Combination of Emotional Expressions Workshop on Representing, Annotating, and Evaluating Non-Verbal and Verbal Communicative Acts to Achieve Contextual Embodied Agents, in conjunction with the Fifth International Conference on Autonomous Agents, May 29, 2001, Montreal, Canada.

[13] Parke, F.I., Waters, K. Computer Facial Animation. Edited by A K Peters, 1996, ISBN 1-56881-014-8.

[14] Parke, F.I., Waters, K. Computer Facial Animation. Edited by A K Peters, 1996, ISBN 1-56881-014-8. Pp. 145-146.

[15] http://mrl.nyu.edu/~perlin/facedemo/

[16] Perlin, K, Goldberg, A. Improv: A System for Scripting Interactive Actors in Virtual Worlds. Computer Graphics; Vol. 29 No. 3., 1996.

[17] Tao, H., and S. Huang, Motion Patterns in Face Animation, IJCAI Conference, Workshop in Animated Interface Agents: making them intelligent, Nagoya, Japan, August 25, 1997.

[18] Terzopoulos D and Waters K: 'Analysis and synthesis of facial image sequences using physical and anatomical models', IEEE, PAMI, 15, No 6 (June 1993).

[19] http://mambo.ucsc.edu/psl/fan.html

[20] Won-Sook Lee, Escher, M., Sannier, G., Magnenat-Thalmann N., MPEG-4 Compatible Faces from Orthogonal Photos.