Generation of Virtual Characters from Personality Traits

Fabrizio Nunnari[1,3]✉ and Alexis Heloir[2,3,4]

[1] fabrizio.nunnari@dfki.de
[2] alexis.heloir@dfki.de
[3] DFKI-MMCI, SLSI group, Saarbrcken, Germany
[4] LAMIH UMR CNRS/UVHC 8201, Valenciennes, France

Abstract. We present a method to generate a virtual character whose physical attributes reflect public opinion of a given personality profile. An initial reverse correlation experiment trains a model which explains the perception of personality traits from physical attributes. The reverse model, solved using linear programming, allows for the real-time generation of virtual characters from an input personality. The method has been applied on three personality traits (dominance, trustworthiness, and agreeableness) and 14 physical attributes and verified through both an analytic test and a subjective study.

1 Introduction

In narrative contexts, such as motion pictures, as well as in interactive contexts, such as video games or conversational agents, a virtual character feels authentic when it fulfills the audience's expectations: the character's appearance should match her/his personality as well as her/his behavior [7]. It has been shown that the better a character looks the part, the more believable and effective she/he will be in the narrative [9,3].

This paper presents a *traits-to-attributes* mapping methodology to generate a virtual character whose physical attributes comply with most people's expectations regarding its assumed personality. The method relies on the use of attributes-based character generation software such as MakeHuman[5] (Figure 1, right), Adobe/Mixamo Fuse[6], Daz3D[7], and Poser[8]. These editors account for the customization of a default character through a set of sliders. Each slider controls the deformation of a physical attribute, such as gender, age, height, torso width, finger length, distance between eyes, and the like.

A personality model is a list of traits (dimensions), and the personality profile of an individual is expressed as the quantification, in a closed range, of each trait. The generation method that we propose has been applied to three personality traits: *dominance*, *trustworthiness*, and *agreeableness*. The *dominance*

[5] `http://www.makehuman.org/` – 10 July 2017
[6] `http://www.adobe.com/products/fuse.html` – 10 July 2017
[7] `http://www.daz3d.com/` – 10 July 2017
[8] `http://my.smithmicro.com/poser-3d-animation-software.html` – 10 July 2017

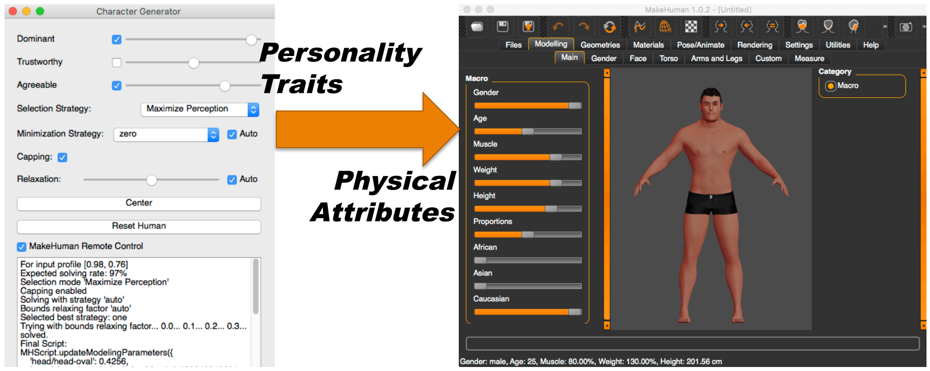

Fig. 1. Our character generation tool (left) allows a designer to provide a personality profile as input. It automatically modulates (some of the) physical attributes of a character editor (MakeHuman, right) so that the resulting character's appearance matches the personality profile given as input.

and *trustworthiness* traits were selected because of their wide acceptance as orthogonal components in the judgment space for first encounters with zero acquaintance [12,4]. Additionally, since there is evidence that people formulate a judgment in less than a second [16], these two traits are suitable candidates for experiments which need to crowd-source many votes in a limited amount of time. We extended the original setup of Oosterhof and Todorov [12] by exposing users to full-body pictures and by adding an additional trait taken from the well established OCEAN model [11] (namely: openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism). We selected agreeableness because there is evidence that people often rely on this trait to guess at the supposed personality of newcomers they meet [1]. This last choice gave us the possibility to investigate how the system behaves with non-orthogonal trait combinations.

Since both the description of a character and a personality profile can be mathematically expressed as points in a closed multi-dimensional space, the *traits-to-attributes* generation consists of mapping a point into a target space with different dimensions. The *traits-to-attributes* generation method accounts for two phases: off-line training and real-time generation. The training phase (Section 3) aims at extracting people's convictions about the association of personality to physical appearance. This knowledge gathering is performed via *reverse correlation* experiments. The generation phase (Section 4) uses linear programming to generate a virtual character from a personality profile. In the remainder of the paper, Section 2 provides an overview of the related work using reverse correlation to support character generation from personality descriptors. Section 5 presents a selection of examples demonstrating the versatility of our approach and the results of a subjective evaluation. Finally, Section 6 concludes the paper.

2 Related work

Reverse Correlation (RC) is an experimental method aiming at highlighting which features of a large set of stimuli better predict judgments. In the field of social perception, it was introduced by Mangini et al. [8] to find out which elements of the human face influence the perception of gender, and discriminate among expressions (happy, sad). An RC perception experiment usually consists of presenting a set of pictures, each one showing a person or an avatar, to a number of subjects and asking them to rank the pictures using Likert scales, along one or more traits. When the synthetic images are parameterized using a set of morphological descriptors, it is possible to highlight the relationship between the perception of an abstract trait (e.g., dominance) and the underlying morphological parameters describing the test images.

The work presented above delivers models able to predict the judgment of an image from the descriptors of the image. Conversely, for authorial purposes, prediction models should be reverted and allow for the creation of a reliable stimulus for an expected judgment. This approach has recently been a subject of few, but promising, investigations. For example, Durupinar et al. [5] developed a system to alter the style of an animation of a virtual character based on the input of an OCEAN personality profile.

As for the creation of the characters' mesh, Vernon et al. [15] developed a system capable of generating new face illustrations from three personality traits: dominance, approachability, and youthful-attractiveness. Similarly to our approach, they gather votes to bootstrap a set of linear models predicting the traits which maximize the perception of the three traits. They later reverse the model using a multi-layered perceptron so they can generate plausible faces from a set of desired traits. However, since the training is based on the location of facial landmarks, rather than high-level morphological parameters, the resulting faces (cartoon-style representation of the key areas that influence the perception of the traits) are usable for illustrative purpose but not for practical authoring.

Recently, Streuber et al. [13] proposed a system for the generation of virtual characters from a set of more than 30 words describing the body shape. Their system is again based on an initial training phase and generation via reverse modeling. It generates bodies belonging to a vector space defined by height principal components obtained from a large collection of scanned human meshes. In contrast, our method focuses on supporting a high degree of uncorrelated physical attributes, leading to a significant difference in the ratio between the cardinalities of the input and the output (e.g., in our work, 3 traits to determine 14 body/face descriptors).

All the works which we surveyed so far conduct RC experiments using Likert ratings. However, Likert scales present several limitations, such as the reluctance of subjects to vote at the extremes of a scale, the subjective variation of votes across sessions, and the need to re-scale previous votes when new absolute references are met [10]. For these reasons, our RC experiments are based on the paired comparison (PC) voting system [2].

Table 1. The attributes modulating the shape of the virtual characters.

MakeHuman ID	Short Name	Description	min	max
chin/chin-bones-in—out	Chin bones	Chin lateral bones extension	0.5	1
chin/chin-height-min—max	Chin height	Distance between chin and lips	0.2	0.8
eyebrows/eyebrows-angle-up—down	Eyebrow angle	Eyebrow inclination	0.2	0.8
eyes/r-eye-size-small—big	Eye size	Size of both eyes	0.1	0.9
head/head-oval	Head ovality	Hard/soft forehead corners	0	0.8
macrodetails-height/Height	Height	From ca. 149cm to 201cm	0.25	0.75
macrodetails-universal/Muscle	Muscularity	Muscular tone of the body	0.2	0.8
macrodetails-universal/Weight	Weight	Overall mass of the body	0.2	0.8
mouth/mouth-scale-horiz-incr—decr	Mouth hscale	Mouth and lip width	0.1	0.9
mouth/mouth-scale-vert-incr—decr	Mouth vscale	Mouth and lip height	0.1	0.9
neck/neck-scale-horiz-less—more	Neck hscale	Neck width	0	1
nose/nose-scale-horiz-incr—decr	Nose hscale	Nose width	0.1	0.9
stomach/stomach-tone-decr—incr	Stomach tone	Belly in/out	0.2	1
torso/torso-vshape-less—more	Torso V-shape	Affects shoulder width	0	0.8

3 Training the model

The training phase aims at building a linear model which, given the physical aspect of a virtual character, predicts how observers would grade its personality traits. The training consists of:

– *Data collection.* Given a set of randomly generated virtual characters, and a number of traits to judge, a pool of human subjects is called to judge on the perception of each trait on each character. Each virtual character is then associated to a numerical quantification of the perception of each trait;
– *Building a prediction model.* A linear regression is run to build a model for the prediction of a trait value from the physical appearance of the character;
– *Simplifying the model.* The full linear models are simplified in order to discard the physical attributes which do not significantly contribute in the perception of a trait. The simplification is performed using two strategies: the first (p-minimization) minimizes the number of attributes, while the second (R-maximization) maximizes the prediction power for a trait.

Data collection. We collected information about the perception of personality traits in relation to physical appearance with a reverse correlation user study based on a paired comparison voting mode. The paired comparison is a preference learning technique which aims at ranking a set of items by asking a preference between two items at a time. Given N items, the number of possible pairs is $P = N * (N - 1)/2$. As output, the paired comparison associates an *estimate* value to each of the items, allowing for their relative ranking.

A set of 50 randomly generated virtual characters was judged by a panel of 50 volunteers on three personality traits: dominance, trustworthiness, and agreeableness. The pictures of the virtual characters were generated using the open source software MakeHuman. MakeHuman allows for the customization of a default character using more than 200 sliders. Each slider modulates the influence of a morph target. For the generation of the characters we selected 14 attributes (listed in Table 1) which visibly alter the appearance of the character

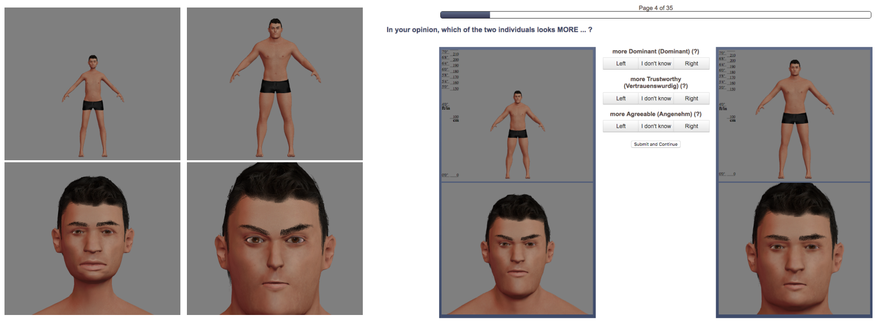

Fig. 2. Left: The characters generated by simultaneously fully minimizing (left) and maximizing (right) all the 14 physical attributes. Right: An example of the voting page shown during the data collection.

in frontal view. In addition, in order to limit biases related to gender or ethnicity discrimination, we locked the *gender* slider to fully masculine and the *ethnicity* to fully Caucasian. Figure 2, left, shows the extreme virtual characters that can be generated by minimizing or maximizing all the 14 physical attributes at once.

Each of the 50 participants voted on 50 pairs of virtual characters through an interface shown in Figure 2, right. Participants were university students from various faculties belonging to different nationalities (30 DE, 5 IN, 3 RU, 2 MEX, 2 IT, 8 not listed); mean age 23.44 years (sd=3.86); level of education: 26 high-school diploma, 11 bachelor's degree, and 13 master's degree. The experiment was conducted in a German university. The experimenters provided instructions in either German or English and supervised the voting session. Each voting session lasted up to 15 minutes. Each voter was rewarded with a meal coupon with a value of 2.85 Euros. On average, the time needed to vote on a pair was 11.38 secs (sd=6.49), and each of the 1225 possible pairs was voted 2.04 times.

The computation of the PC estimates associates each virtual character to a triplet of values, which indicate how much an observer perceives the character as *dominant*, *trustworthy*, and *agreeable*. For each trait t, the minimum and maximum estimates (E_t^{min}, E_t^{max}) are used to normalize the input traits in a range $[0, 1]$. The estimates were computed using the *prefmod* [6] R module. All the experiments were conducted on the online DeEvA platform[9].

Building the prediction model. We derive three separate linear models, one for each of the three personality traits $t \in T$, by performing a **linear regression** between the character's attributes (predictors) and the trait estimates (measured variable). The output of each regression is: i) an intercept value i_t, and ii) a row vector of coefficients C_t. Given a set of attribute values as column vector $X = \{x_a, a \in A\}$, where A is the set of physical attributes, the formula to estimate the value p_t of the personality trait t is:

$$p_t = i_t + C_t \times X \tag{1}$$

[9] `https://deeva.mmci.uni-saarland.de/` – 10 July 2017

Table 2. The prediction models selected via p-minimization and R-maximization. The C_x and the p columns report respectively the slope and the p-value of the linear regression for each physical attribute (*=< 0.05, **=< 0.01, ***=< 0.001).

Selection	**p-minimization**						**R-maximization**					
Trait (t)	**Dominant**		**Trustworthy**		**Agreeable**		**Dominant**		**Trustworthy**		**Agreeable**	
#selected attr.	6		4		4		12		7		9	
adjusted R^2	0.903		0.726		0.747		0.917		0.749		0.769	
intercept (i_t)	-2.156		0.129		0.316		-2.174		-0.241		-0.012	
Attribute Name	$\mathbf{C_D}$	**p**	$\mathbf{C_T}$	**p**	$\mathbf{C_A}$	**p**	$\mathbf{C_D}$	**p**	$\mathbf{C_T}$	**p**	$\mathbf{C_A}$	**p**
Chin bones	-	-	-	-	-	-	0.223	0.196	-	-	-	-
Chin height	-	-	-	-	-	-	0.344	*	0.601	*	-	-
Eyebrow inclination	2.100	***	-2.080	***	-2.234	***	2.092	***	-2.258	***	-2.313	***
Eye size	-	-	0.451	*	0.432	*	-	-	0.455	*	0.328	0.094
Head ovality	-	-	-0.661	***	-0.782	***	-0.178	0.077	-0.740	***	-0.675	***
Height	1.688	***	0.829	**	0.897	**	1.846	***	1.131	***	0.970	**
Muscularity	0.335	*	-	-	-	-	0.326	*	0.314	0.165	0.324	0.166
Weight	0.605	***	-	-	-	-	0.538	***	-	-	-	-
Mouth/lip width	-	-	-	-	-	-	-0.124	0.239	-	-	0.284	0.130
Mouth/lip height	-	-	-	-	-	-	-0.110	0.273	-	-	0.221	0.197
Neck width	0.521	***	-	-	-	-	0.483	***	-	-	-	-
Nose width	-	-	-	-	-	-	-0.212	0.065	-	-	0.210	0.298
Stomach tone	-	-	-	-	-	-	-	-	-	-	-0.287	0.089
Torso V-shape	0.524	***	-	-	-	-	0.409	***	-0.251	0.180	-	-

Simplifying the prediction model. Each of the three linear models is simplified into two simpler models using a **backward elimination**. The backward elimination approach is an iterative model selection technique which reduces the number of predictors (here, the physical attributes) explaining a variable. We apply the backward elimination with two selection strategies: *p-value minimization* and *R-correlation maximization*.

The model selection based on **p-minimization** considers the p-value associated to each variable as a result of the regression and discards the variable with the higher p-value above the threshold α. The algorithm iterates using the reduced variable set until there are no variables with p-value $\geq \alpha$. In this work, we use $\alpha = 0.05$. In contrast, the model selection based on **R-maximization** takes into account the correlation factor R of the initial regression. Then it computes how the correlation varies by removing each variable, one at a time. The algorithm removes the variable which causes the lowest increment of R and iterates until no removal increases the correlation factor. The selection by p-minimization leads to models with a minimal set of variables, while the selection with R-maximization leads to models with the highest prediction power. The former method allows us to trigger the perception of a trait using the smallest possible number of attributes, leaving more freedom to the author for further customization, while the latter method is more suitable to maximize the perception of the trait. The linear regressions and the model selection were computed in the R programming environment using the default `lm` function. The maximization was based on the adjusted-R-squared correlation factor.

Table 2 shows the results for all of the six models selected using both p-minimization and R-maximization strategies for each trait. As expected, in the p-minimization mode there are fewer attributes compared to the R-maximization. Since the correlation of the estimates between trustworthiness and agreeableness is very high (r=0.965), they share the same attributes. On the other hand, the correlation is lower for dominance vs. trustworthiness (r=-0.402) (confirming the findings of Oosterhof et al. [12]) and for dominance vs. agreeableness (r=-0.400). These finding match with previous research. Concerning the perception of dominance, Toscano et al. [14] already reported on the importance of the inclination of the eyebrows, and Windhager et al. [17] reported the relevance of the rectangularity of the face and of the chin bones. Furthermore, both of the aforementioned works recognized the influence of lip thickness, mouth width, and the eyes' aperture. This experiment, which includes full-body pictures, adds the relevance of the height, weight, muscularity, neck and shoulder width, and stomach tone to the perception of these three traits.

4 Character generation model

The generation of the character takes as input a set of traits and a quantification of their desired level of perception for an observer. The output consists of the values of the physical attributes needed to build an avatar whose appearance triggers the perception of the input personality. In the rest of this section:

- *Problem statement* describes how the generation problem is posed as a linear programming problem;
- *Filter by solvability rate* addresses the issue of unsolvability of a linear problem for some trait combination. It shows how to estimate the solvability chances through simulation;
- *The objective function* illustrates why we need to define several objective functions and how to find, for each trait combination, the function which minimizes prediction errors;
- *Coerce attribute progression* illustrates an additional pair of constraints which improve the smoothness of the solution space; and
- *Evaluating the coercion* presents a quantitative measurement of the improvements introduced by the coercion mechanism.

Problem statement. The linear models derived in the previous section can be combined into a single linear system which can be reverted to calculate the expected physical attributes from a set of personality trait values. Given a personality profile $P = \left\{p_t, t \in T, E_t^{min} \leq p_t \leq E_t^{max}\right\}$, the values of X (physical attributes) that lead to the perception of P can be calculated by solving the linear problem:

$$\arg\min_X \{G * X\} = \arg\min_X \{g_a x_a, a \in A\} \tag{2}$$

subject to:

$$P - I = \mathbf{C} * X \tag{3}$$

Table 3. For each trait combination, the table shows a *solvability rate*, i.e., the chance of being able to generate a character from a given personality profile.

Selection Traits	p **#Attrs**	**#Coeffs**	**Solve Rate**	R **#Attrs**	**#Coeffs**	**Solve Rate**
D	6	6	100.0%	12	12	100.0%
T	4	4	100.0%	7	7	100.0%
A	4	4	100.0%	9	9	100.0%
D,T	8	10	91.1%	13	19	99.7%
D,A	8	10	90.0%	14	21	97.2%
T,A	4	8	3.3%	11	16	62.7%
D,T,A	8	14	4.1%	14	28	55.8%

$$a^{min} \leq x_a \leq a^{max}, a \in A \tag{4}$$

where $G = \{g_t, t \in T\}$ is the row vector of coefficients of the *objective function*, and $I = \{i_t, t \in T\}$ is the vector of intercepts. The matrix $C = (c_{t,a})$ contains on each line the coefficients C_t of the linear model of a trait (see Table 2): each column is associated to an attribute, and $c_{t,a} = 0$ if the attribute has been eliminated from the trait during the model selection. Finally, each $x_a \in X$ is bound to its min/max values, as documented in Table 1. As for the dimensions: $|P| = |I| = |T|$, the number of traits; $|X| = |A|$, the number of attributes; and the matrix C has $|T|$ rows and $|A|$ columns.

For convenience, the input is provided as normalized personality profile vector $\hat{P} = \{\hat{p_t} \in [0, 1], t \in T\}$. The non-normalized vector P of estimates is calculated as: $P = \{E_t^{min} + (E_t^{max} - E_t^{min}) * \hat{p_t}, t \in T\}$, where E^{min} and E^{max} are the vectors of minimum and maximum estimates for each trait. In this work, we solve linear problems using the simplex method as implemented in the `linprog` function exposed by the `scipy.optimize`[10] python module.

Filter by solvability rate. A linear problem might be impossible to solve. In order to provide feedback to the user about the feasibility of a request, we precompute a *solvability rate* for each trait combination for both p and R selection. As can be seen in Table 3, the percentage of solvability drops when the model combination contains highly correlated traits (trustworthiness and agreeableness). The DTA combination in R-maximization rises to 55.8% thanks to the fact that the (higher number of) attributes distribute differently between the T and A traits. Further analysis will consider only trait combinations where the success rate is above 90%, plus the combination of all the three traits (DTA) in R-maximization selection, because it provides the most interesting study case.

The objective function. In linear programming, each coefficient $g_a \in G$ of the objective function (Equation 2) is ideally associated to a *cost*. Among an infinite number of possible solutions, the solver will choose a vector X which minimizes the overall cost.

However, in our case there is not an obvious cost associated to a physical attribute. Hence we conceived and tested six different strategies to assign, for

[10] `https://www.scipy.org/` – 10 July 2017

Table 4. Top: The average MSE for each objective function in different selection/trait combinations. The bold text highlights the minimum value(s) for each condition.

		Minimization Strategy						
Selection	**Traits**	**zero**	**one**	**minus_one**	**sign_count**	**sum_coeff**	**sum_coeff_over_p**	**Solve Rate**
p	D	0.289	0.290	0.301	0.301	0.289	**0.286**	100%
	T	0.244	0.244	0.261	**0.233**	0.244	0.251	100%
	A	0.239	0.239	0.259	**0.236**	0.239	0.245	100%
	D,T	0.252	0.252	0.263	0.262	0.252	**0.248**	100%
	D,A	0.251	0.251	0.255	0.256	0.251	**0.245**	98%
R	D	0.311	**0.310**	0.332	0.319	0.311	0.318	100%
	T	**0.287**	**0.287**	0.311	0.300	**0.287**	0.290	100%
	A	**0.291**	**0.291**	0.324	0.310	**0.291**	0.300	100%
	D,T	**0.287**	0.291	0.307	0.289	**0.287**	0.299	100%
	D,A	0.285	**0.284**	0.306	**0.284**	0.285	0.294	100%
	D,T,A	0.266	**0.264**	0.289	0.275	0.266	0.278	100%

each attribute $a \in A$, the corresponding coefficient $g_a \in G$:
zero: $g_a = 0$. The solver is subject only to the equality (Equation 3) and to the variable boundaries (Equation 4).
one: $g_a = 1$. The solver will push all variables to their minimum value.
minus_one: $g_a = -1$. The solver will push all variables to their maximum value.
pos_neg_usage_count: $g_a = \sum_{t \in T} -sgn(c_{t,a})$. The solver will maximize variables with a high number of positive coefficients (vice versa for negative ones).
sum_coeff: $g_a = \sum_{t \in T} -c_{t,a}$. The higher the overall coefficients sum, the more the variable will be favored in the maximization. The aim is to give a higher priority to maximization of variables with stronger positive correlations (vice versa for negative correlations).
sum_coeff_over_p: $g_a = \sum_{t \in T}(-c_{t,a}/p_{t,a})$. As the sum_coef strategy, plus each coefficient will be divided by its p-value resulting from the linear regression. The smaller the p-value, the higher the absolute value of the cost. This strategy increases the influence for variables with higher significance.

In order to assess the efficacy of each minimization strategy, we solved the *traits-to-attributes* problem using the same data used for training. In practice, we took the personality triplet associated to each of the 50 virtual characters and back-calculated their physical attributes. Then, we measured the mean squared error (MSE) for each attribute and averaged all of them together. The MSEs are normalized on each attribute min/max range. Table 4 shows the MSE of each minimization strategy for each condition. More elaborate strategies, taking into account the coefficient and the significance of an attribute, helped in reducing the error only for the p-minimization mode (fewer attributes). Given a trait combination, the *traits-to-attributes* model will use this table during the generation phase to select the minimization strategy which leads to the smallest error.

The average error among all conditions, in percentage ($\sqrt{MSE}*100$), is 51.9% (min 48.3%, max 55.6%, sd 2.5%). The next paragraph explains the reason for such high error and presents a strategy to reduce it.

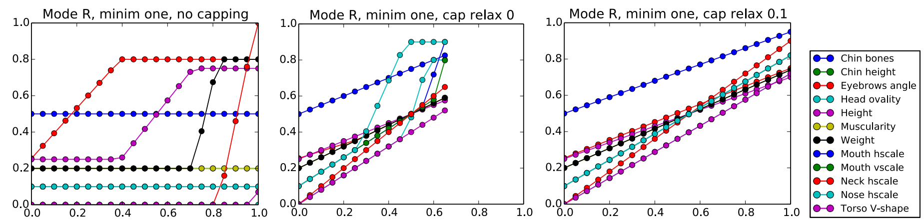

Fig. 3. Variation of the attribute values (y) as a function of the level of *dominance* (x). Left: unbounded; Center: capped with no relaxation; Right: capped with relax, at 0.1.

Coerce attribute progression. As defined so far, the *traits-to-attributes* model provides solutions with an uneven increment of the attribute values over the input range. For example, as can be seen in Figure 3 (left), as the level of dominance increases, the solver maximizes the attribute values one by one, and some of the attributes are locked at their maximum/minimum value. Overall, this increments the error rate of the solver. Also, for authorial purposes a smoother and more evenly distributed increment of all attributes over the trait range would be preferred.

Hence, we introduced a *capping* mechanism to drive the attributes towards a smoother increment. The capping mechanism is based on the sign of the coefficients of the objective function: if a coefficient g_a is negative (i.e., the solver tends to maximize the variable) we impose as upper bound to the variable x_a the same value of the input trait. Vice versa, the bound is on the lower value when g_a is positive. In the case of multiple input traits, the upper and minimum bounds are set to the maximum and minimum value among all traits. Figure 3 (center) shows the result of this strategy: the behavior of the attributes has improved, but the bounding restriction leads more easily to unsolvable problems. Hence, we introduce a relaxation factor $R \in [0, 1]$ which softens the capping constraints. The resulting capping strategy is formally expressed as:

$$\begin{cases} x_a \leq a^{max} - (a^{max} - a^{min}) * \max\{p \in \hat{P}\} * (1 - R) & g_a < 0 \\ x_a \geq a^{min} + (a^{max} - a^{min}) * \min\{p \in \hat{P}\} * (1 - R) & g_a > 0 \end{cases}$$

If the coefficient g_a is 0, the variable bounds are in any case constrained to $a^{min} \leq x_a \leq a^{max}$, as defined in the basic model. With $R = 0$ the bounds are strict and the linear problem is harder to solve, while with $R = 1$ there is no more effect of the capping and the "one-by-one increment" behavior arises. Figure 3 (right) shows the behavior with $R = 0.1$. To verify that the capping mechanism improves the precision, we recomputed the MSEs table with the *capping* enabled and a relaxation set at 0.25. Although the solve rate decreased from 99.8% to 84.3%, the average error decreased from 51.9% to 31.7% (sd 5.1%).

Evaluating the coercion. We evaluated the behavior of the coercion mechanism for all of the three traits in both p and R selection modes. For each trait (D, T, or A), we solved the traits-to-attributes problem with and without the capping

Table 5. Correlation gain between noCap and Cap conditions.

	p						R					
	noCap		Cap				noCap		Cap			
trait	mean	(sd)	mean	(sd)	gain	Fisher's p	mean	(sd)	mean	(sd)	gain	Fisher's p
D	0.619	(0.345)	0.992	(0.005)	**60.29%**	<0.001	0.276	(0.386)	0.964	(0.067)	**249.23%**	<0.001
T	0.776	(0.179)	0.820	(0.339)	5.68%	0.395	0.497	(0.385)	0.940	(0.085)	**88.95%**	<0.001
A	0.781	(0.176)	0.895	(0.186)	**14.60%**	<0.01	0.536	(0.356)	0.907	(0.181)	**69.29%**	<0.001

mechanism (Cap, noCap) by providing an input from 0.0 to 1.0 in 101 equidistant steps. In the Cap condition, the relaxation factor is automatically determined by trying to progressively solve the problem with relaxation starting from 0.0 and 0.1 step increments. When 1.0 is reached, the problem is by definition solved as in the noCap condition. In our tests, about 50% of the problems were solved with relaxation at 0.0. The remaining problems were solved with a relaxation value uniformly distributed between 0.1 and 1.0. We computed the correlations between the input trait value and every output attribute. Table 5 reports the average of the correlations among all attributes. The correlations systematically incremented in the Cap condition. The last column reports the significance of the difference between the two correlations using a Fisher r-to-z transformation. With the exception of trustworthiness in p mode, all the correlations increased significantly.

5 Examples and validation study

A prototype GUI (see Figure 1, left) allows artists an interactive exploration of the personality space through a set of sliders. The user can enable or disable each trait independently and can decide to minimize the number of attributes or maximize the perception of the traits. The system automatically selects the objective function which minimizes the error. Also, it tries to solve the problem with an initial relaxation factor of 0.0 and increments of 0.1. A text area previews the script that will be executed by MakeHuman. For the execution, we implemented a MakeHuman plugin which allows for the remote execution of scripts via TCP connections. Figure 4 shows several examples of generation using one to three attributes at the same time.

We ran three experiments to assess the quality of the generation model. In *Experiment 1*, following the same procedure described in Section 3, 36 participants (27 male, 9 female) of different nationalities (16 DE, 4 IN, 3 CH, 13 not listed) voted on 25 pairs randomly composed from 11 virtual characters. The characters were created by modulating *input* dominance from 0.0 to 1.0 with 0.1 increment steps. Similarly to the training experiment (Section 3), subjects had to answer "Which of the two characters looks more dominant?". An analysis of the paired comparison data determined a level of *perceived* dominance for each character. A linear regression between the *input* and the *perceived* dominance led to a correlation factor of 0.984.

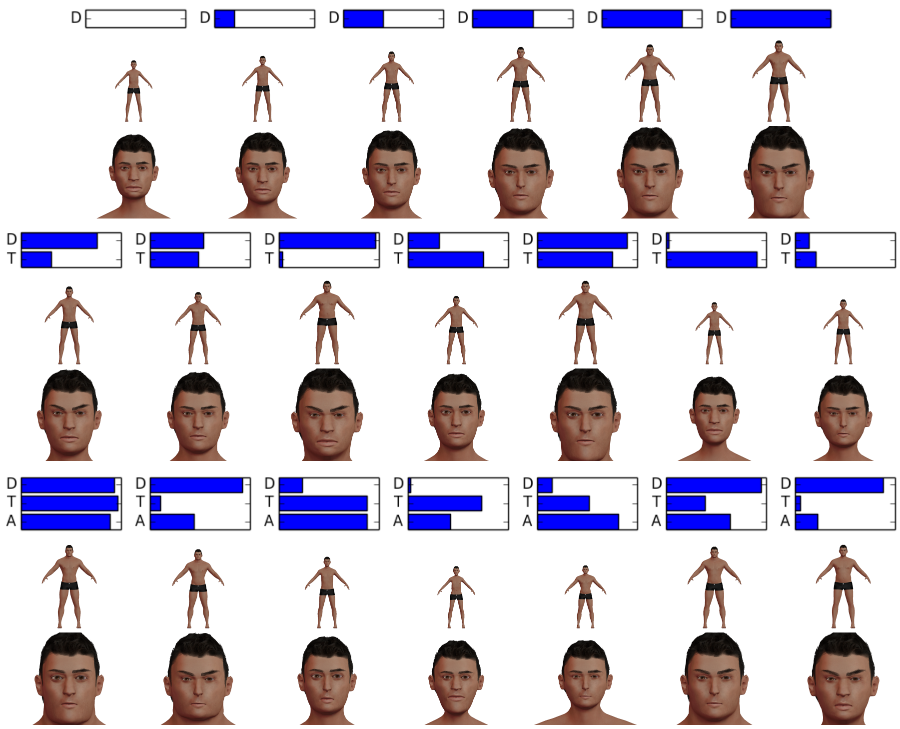

Fig. 4. Examples of generation in Maximize Perception mode (12 physical attributes). Top: progression of *dominance* from 0% to 100%. middle: generation with *dominance* and *trustworthiness*. bottom: including *agreeableness*.

In *Experiment 2*, 37 participants (28 male, 9 female) of different nationalities (16 DE, 4 IN, 3 CH, 14 not listed) voted on 25 pairs from 50 randomly generated virtual characters. The characters were created by randomly modulating both dominance and trustworthiness in the range 0.0 to 1.0. Subjects had to answer which of the two characters looked more dominant and which one more trustworthy. An analysis of the paired comparison data determined a level of *perceived* dominance and *perceived* trustworthiness for each character. A linear regression between the *input* and *perceived* dominance led to a correlation factor of 0.933, while the correlation between *input* and *perceived* trustworthiness is 0.716.

In *Experiment 3*, 40 participants (28 male, 12 female) of different nationalities (21 DE, 4 IN, 4 GR, 11 not listed) voted on 50 random pairs from 125 randomly generated virtual characters. The generation of the characters, the voting method, and the data analysis are the same as for Experiment 2, with the addition of agreeableness. The correlation between *input* and *perceived* trait is 0.903 for dominance, 0.733 for trustworthiness, and 0.717 for agreeableness.

We assessed the capability of the model to scale on multiple traits using a Fisher r-to-z transformation. The test measures the significance of the difference

Table 6. Results of the validation study. Left: The correlations between input and perceived trait values. Right: the significance of the variation of the correlations.

Exp	Traits	#participants	#characters	correlation D	correlation T	correlation A
1	D	36	11	0.984	-	-
2	D, T	37	50	0.933	0.716	-
3	D, T, A	40	125	0.903	0.735	0.717

Trait	Experiments	Fisher's p
D	1 vs. 2	**0.067**
D	2 vs. 3	**0.259**
D	1 vs. 3	0.016
T	2 vs. 3	**0.810**

between two correlation factors, and we applied it on four pairs of correlations. In this case, the absence of significant difference is desirable, because it means that the correlation between the input and the perceived values of a trait is not affected by the introduction of more traits to the model. For *dominance*, there is no significant difference in the correlations between Experiments 1 and 2 (p=0.067), meaning that the insertion of trustworthiness did not degrade the perception of dominance. As well, there is not significant difference in the correlations between Experiments 2 and 3 (p=0.259), meaning that introducing agreeableness did not degrade the perception of dominance when dominance and trustworthiness are generated together. However, there is a significant difference between Experiments 1 and 3 (p=0.016), meaning that the capability to generate a dominant character significantly decrease when trustworthiness and agreeableness are added to the model. Yet, the significance is modest and the correlation for dominance is still above 0.9. For *trustworthiness*, there is no significant difference between experiments 2 and 3 (p=0.810), meaning that the capability of generating a trustworthy character in the dominance/trustworthiness model doesn't degrade when including agreeableness into the model.

Overall, the above described results (summarized in Table 6), suggest that the method scales relatively well when adding more traits to the generation process. Time and resource constraints prevented us from running studies with the three remaining combinations (trustworthiness and agreeableness, alone and paired), which is desired for future work.

6 Conclusions and future work

This paper introduced a method to draft virtual characters whose appearance suggests to observers a given personality. The method is composed of a training phase, based on reverse correlation experiments, and a real-time generation phase, exploiting linear programming. The method accounts for a *coercion* constraining mechanism which improves the linearity of the solutions. A subjective user study suggests that the system scales well when generating characters using both orthogonal and quasi-co-linear traits. In future experiments we will investigate how the method behaves with a higher number of traits, such as all of the Big Five [11] simultaneously. Although this work focuses on personality traits, the same method can be applied to any kind of subjective descriptor, such as beauty, scariness, appeal, empathy, and the like, paving the way for the generation of virtual characters based on textual input.

Acknowledgements The authors would like to thank Nicolas Erbach for his precious help with the code development and in conducting the user studies.

References

1. Ames, D.R., Bianchi, E.C.: The agreeableness asymmetry in first impressions: Perceivers' impulse to (mis)judge agreeableness and how it is moderated by power. Personality and Social Psychology Bulletin 34(12), 1719–1736 (2008)
2. Bradley, R.A., Terry, M.E.: Rank analysis of incomplete block designs. Biometrika 39, 324–335 (1952)
3. Cassell, J.: Social practice: becoming enculturated in human-computer interaction. In: Universal Access in Human-Computer Interaction. Applications and Services, Lecture Notes in Computer Science, vol. 5616, pp. 303–313. Springer (2009)
4. Dotsch, R., Todorov, A.: Reverse correlating social face perception. Social Psychological and Personality Science 3(5), 562–571 (2012)
5. Durupinar, F., Kapadia, M., Deutsch, S., Neff, M., Badler, N.I.: PERFORM: Perceptual approach for adding OCEAN personality to human motion using laban movement analysis. ACM Transactions on Graphics 36(1), 1–16 (2016)
6. Hatzinger, R., Dittrich, R.: prefmod: an R package for modeling preferences based on paired comparisons, rankings, or ratings. Journal of Statistical Software 48(10), 1–31 (2012)
7. Loyall, A.B.: Believable agents: building interactive personalities. PhD dissertation, Carnegie Mellon University, Pittsburgh (1997)
8. Mangini, M.C., Biederman, I.: Making the ineffable explicit: estimating the information employed for face classifications. Cognitive Science 28(2), 209–226 (2004)
9. Mansour, S., El-Said, M., Rude-Parkins, C., Nandigam, J.: The interactive effect of avatar visual fidelity and behavioral fidelity in the collaborative virtual reality environment on the perception of social interaction. In: Proceedings of the 10th International Conference on Communications. pp. 387–395. WSEAS (2006)
10. Martinez, H., Yannakakis, G., Hallam, J.: Don't classify ratings of affect; rank them! Affective Computing, IEEE Transactions on 5(3), 314–326 (2014)
11. McCrae, R.R., John, O.P.: An introduction to the five-factor model and its applications. Journal of Personality 60(2), 175–215 (1992)
12. Oosterhof, N.N., Todorov, A.: The functional basis of face evaluation. Proceedings of the National Academy of Sciences 105(32), 11087–11092 (2008)
13. Streuber, S., Quiros-Ramirez, M.A., Hill, M.Q., Hahn, C.A., Zuffi, S., O'Toole, A., Black, M.J.: Body Talk: Crowdshaping realistic 3d avatars with words. ACM Trans. Graph. 35(4), 54:1–54:14 (2016)
14. Toscano, H., Schubert, T., Sell, A.N.: Judgments of dominance from the face track physical strength. Evolutionary Psychology 12(1), 1–18 (2014)
15. Vernon, R.J.W., Sutherland, C.A.M., Young, A.W., Hartley, T.: Modeling first impressions from highly variable facial images. Proceedings of the National Academy of Sciences 111(32), E3353–E3361 (2014)
16. Willis, J., Todorov, A.: First Impressions: Making Up Your Mind After a 100-Ms Exposure to a Face. Psychological Science 17(7), 592–598 (2006)
17. Windhager, S., Schaefer, K., Fink, B.: Geometric morphometrics of male facial shape in relation to physical strength and perceived attractiveness, dominance, and masculinity. American Journal of Human Biology 23(6), 805–814 (2011)