Generating Virtual Characters from Personality Traits via Reverse Correlation and Linear Programming

(Extended Abstract)

Fabrizio Nunnari
DFKI / MMCI, Saarbrücken, Germany
fabrizio.nunnari@dfki.de

Alexis Heloir
DFKI / MMCI, Saarbrücken, Germany
LAMIH UMR CNRS/UVHC 8201, Valenciennes, France
alexis.heloir@dfki.de

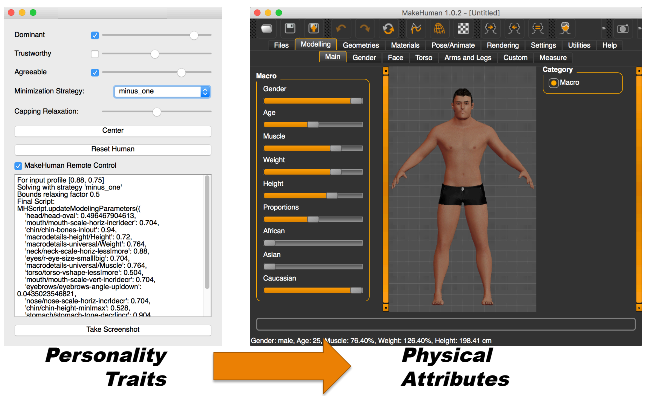

Figure 1: The character generation GUI (left) takes a 3-dimension personality profile as input. It computes a combination of physical attributes compatible with off-the-shelf avatar authoring tools (right).

Keywords

Virtual character generation, personality traits, physical attributes, reverse correlation, paired comparison.

1. INTRODUCTION

The usefulness of a virtual character depends on its ability to fulfill the user's expectations: among other aspects, the character's appearance should match her/his personality as well as her/his behavior [11]. It has indeed been shown that the better a character looks the part, the more believable and effective she/he will be in the narrative [13, 8].

This paper presents a system which generates a virtual character defined along three personality traits: Dominance, Trustworthiness, and Agreeableness. From these three traits, 14 surface physical attributes of the target character are automatically inferred: chin bones, chin height, eyebrow angle, eye size, head ovality, height, muscularity, weight, mouth and lip width, mouth and lip height, neck width, nose width, stomach tone, and torso V-shape. The system is based on a character model where each physical attribute is modulated by a morph target deforming a base character. Several authoring solutions are available off-the-shelf, either as commercial products (Adobe Fuse [1], Daz3D [2], and Poser [4]) or open-source software (MakeHuman [3], Fig. 1, right). In these solutions, each morph target is controlled by a slider in the GUI, allowing non-technically experienced users to generate plausible characters in a fraction of the time needed with professional authoring tools.

The configuration of our *traits-to-attributes* system accounts for an initial **training** phase, based on a *reverse correlation* experiment (e.g., [12, 9]), from which we infer a multivariate linear model explaining the relationship between the perception of the three personality traits and the fourteen physical attributes. The inverse model – solved using linear programming – allows for the real-time **generation** of virtual characters from an input personality profile.

Similar work allows for the generation of faces from personality traits [16] or full bodies from shape descriptors [15]. Our system handles both bodies and faces. Furthermore, the generated characters are described with a set of high-level, character-designer-friendly physical descriptors, allowing for a further manual refinement of the characters. Finally, the method we present specifically addresses the challenge of manipulating a high number of descriptors on a limited number of traits (14 attributes from 3 traits, in this study).

2. TRAINING THE MODEL

In an experiment, we collected information about the perception of three personality traits in relation to fourteen physical attributes with a *reverse correlation* study based on *paired comparison* voting mode [7]. The paired comparison method is a preference learning technique which aims at ranking a set of N items by asking for a preference between two items at time. As output, the paired comparison associates an *estimate* value to each of the items, allowing for a relative ranking.

Fifty participants voted on 50 pairs randomly selected from a set of 50 randomly generated virtual characters; subjects had to declare which of the two characters looked more dominant, more trustworthy, and more agreeable (Fig. 2). The computation of the PC estimates results in associating each virtual character to a triplet of values, which indicate to what degree an observer perceives the character as dominant, trustworthy, and agreeable. The estimates were computed using the *prefmod* R module [5, 10].

Then, we derive three separate linear models, one for each of the three personality traits, by performing a **lin-**

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

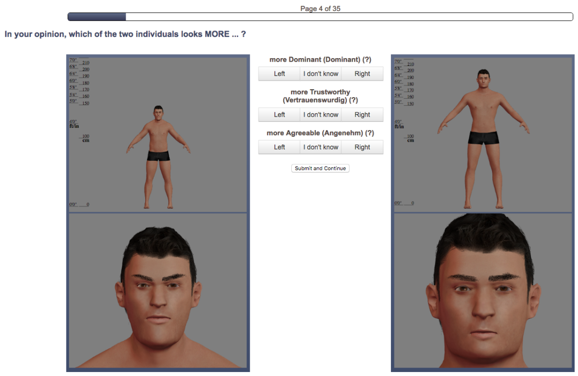

Figure 2: An example of the voting page where subjects had to choose which character was more dominant, trustworthy and agreeable.

ear regression between the character's attributes (predictors) and the trait estimates (measured variable). Finally, each of the three linear models is simplified into two simpler models using a **backward elimination**: an iterative model selection technique which reduces the number of predictors (here, the physical attributes) explaining a model. Two selection modes lead to models able to trigger the perception of a trait either by using a minimal number of attributes (p-minimization) or by maximizing the prediction power (R-maximization).

3. CHARACTER GENERATION MODEL

The linear models derived during the training are combined into a single **linear system** which is reverted to calculate the expected physical attributes from a set of personality trait values. In this work, we solve linear problems using the simplex method as implemented in the `linprog` function exposed by the `scipy.optimize` python module [6]. The input to the solver is a personality profile P, expressing a degree between 0 and 1 for each of the three traits. The output is the list of values for each of the fourteen physical attributes. The constraints for the solver are the min/max range for each attribute as used for the generation of the random characters, the coefficients derived from the linear regressions of the training phase, and an objective function for the minimization. In order to improve the solutions offered by the solver in its basic configuration, we added further constraints, as described below.

Filter by solvability rate We implemented a simulation procedure which solves the problem for all combinations of traits and selection modes on thousands of random inputs. The resulting percentage of coverage of the input space can be used to warn the user if it is impossible to generate characters for some combinations of traits.

Choosing the objective function We conceived and tested six different objective functions. To assess the efficacy of each minimization strategy, we solved the *traits-to-attributes* problem using the same data computed during the training, and we measured the overall mean squared error (MSE) for each attribute. According to our assessment, there is no "best" objective function: each function can minimize the error for some of the combinations of traits and selection modes (p-min or R-max). Hence, given a user input, the generation procedure accounts for the selection of the objective function which minimizes the error.

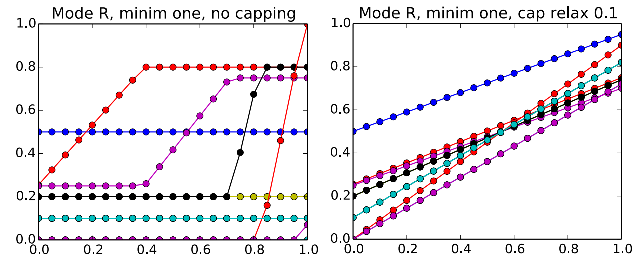

Figure 3: Attribute values (y) as a function of dominance (x); unbound (left) or coerced (right).

Coerce attributes progression As defined so far, the *traits-to-attributes* module provides solutions which suffer from unpredictable uneven increments (see Fig. 3, left). For the author's 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 uses the values of the coefficients of the objective function: if a coefficient is positive/negative (i.e., the solver tends to minimize/maximize the variable), we impose a lower/upper bound proportional to the input trait. In the case of multiple input traits, the bounds are set to the minimum/maximum value among all traits. With this first solution, the attributes are better distributed, but the system is more likely to be unsolvable. Hence, we introduce a relaxation factor $R \in [0, 1]$ which softens the capping constraints. Fig. 3, right, shows the behavior with $R = 0.1$. Further solvability tests show that with this strategy both the MSEs and the average error decrease.

4. CONCLUSION

A prototype GUI (Fig. 1, left) allows artists an interactive exploration of the personality space through a set of sliders. In this version, the user manually selects both the objective function and the value of relaxation. In future versions, the system will automatically choose them in order to maximize the solvability range and minimize the errors. Future experiments will focus on supporting a higher number of traits, such as all of the Big Five [14] simultaneously. This method has the potential to reshape the traditional production pipeline used to produce virtual characters. In the near future, character designers might rely on this method to draft a first version of a character coinciding with its audience's preconceptions relating appearance to non-surface traits like personality, moral alignment, political opinions, spiritual beliefs, etc. This solution integrates well into existing production pipelines: once generated, the designer has the freedom to further refine the character manually by adjusting sliders in the character authoring tool originally used in the production pipeline. The work presented in this paper is the first of a series aiming at shifting the input dimensions for the generation of virtual characters from geometric to non-geometric descriptors. In fact, the same method can be applied to any subjective descriptors, such as beauty, scariness, appeal, empathy, and the like, paving the way for a generation driven by textual descriptions of characters in natural language.

REFERENCES

[1] Adobe/Mixamo Fuse. `http://www.adobe.com/products/fuse.html`. Accessed 1 Feb 2017.

[2] Daz3D. `http://www.daz3d.com/`. Accessed 15 Nov 2016.

[3] MakeHuman. `http://www.makehuman.org/`. Accessed 1 Feb 2017.

[4] Poser. `http://my.smithmicro.com/poser-3d-animation-software.html`. Accessed 15 Nov 2016.

[5] prefmod. `www.jstatsoft.org/article/view/v048i10`. Accessed 15 Nov 2016.

[6] SciPy. `https://www.scipy.org/`. Accessed 15 Nov 2016.

[7] T. Bramley. Paired comparisons methods. In *Techniques for Monitoring the Comparability of Examination Standards*, pages 246–294. Qualification and authority, London, 2007.

[8] J. Cassell. Social practice: becoming enculturated in human-computer interaction. In *Universal Access in Human-Computer Interaction. Applications and Services*, volume 5616 of *Lecture Notes in Computer Science*, pages 303–313. Springer Berlin Heidelberg, 2009.

[9] R. Dotsch and A. Todorov. Reverse correlating social face perception. *Social Psychological and Personality Science*, 3(5):562–571, Sept. 2012.

[10] R. Hatzinger and R. Dittrich. prefmod: an R package for modeling preferences based on paired comparisons, rankings, or ratings. *Journal of Statistical Software*, 48(10):1–31, 2012.

[11] A. B. Loyall. *Believable agents: building interactive personalities.* PhD thesis, Mitsubishi Electric Research Laboratories, 1997.

[12] M. C. Mangini and I. Biederman. Making the ineffable explicit: estimating the information employed for face classifications. *Cognitive Science*, 28(2):209–226, 2004.

[13] S. Mansour, M. El-Said, C. Rude-Parkins, and J. Nandigam. 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 WSEAS International Conference on Communications*, '06, pages 387–395. WSEAS, 2006.

[14] R. R. McCrae and O. P. John. An introduction to the five-factor model and its applications. *Journal of Personality*, 60(2):175–215, June 1992.

[15] S. Streuber, M. A. Quiros-Ramirez, M. Q. Hill, C. A. Hahn, S. Zuffi, A. O'Toole, and M. J. Black. Body Talk: crowdshaping realistic 3d avatars with words. *ACM Trans. Graph.*, 35(4):54:1–54:14, July 2016.

[16] R. J. W. Vernon, C. A. M. Sutherland, A. W. Young, and T. Hartley. Modeling first impressions from highly variable facial images. *Proceedings of the National Academy of Sciences*, 111(32):E3353–E3361, 2014.