Tracking two pleasures

Aenne A. Brielmann & Denis G. Pelli

**Detailed analysis description**

In the following, we provide a detailed description of all analyses that were carried out with MATLAB that shall enable reconstruction of our analyses with a different software.

**analyses01: Correlation between baseline ratings, average beauty, and valence**

For each participant, we calculated the Pearson correlation coefficient between her baseline beauty ratings for each image and 1) mean beauty ratings for those images obtained from a separate sample of mTurk participants by our lab (Brielmann & Pelli, under review; <https://github.com/aenneb/OASIS-beauty>), and 2) mean valence ratings for those images provided by Kurdi and colleagues (Kurdi, Lozano, & Banaji, 2017; <https://osf.io/6pnd7/>). Rating pairs with missing values were excluded from the calculation on a pairwise basis. We then calculated the mean, minimum, and maximum correlation across participants.

**analyses02: ICC**

To calculate intraclass-correlation coefficient (ICC), we first combined the data of all participants into a single matrix. Next, we used the MATLAB function ICC (<https://www.mathworks.com/matlabcentral/fileexchange/22099-intraclass-correlation-coefficient-icc>; see overview for a detailed description) to calculate the ICC for baseline ratings. Data from participants with missing values in their baseline ratings, for the main data this was participant #1, were excluded from this analysis. ICC was calculated as “'A-1': case 2: The degree of absolute agreement among measurements.”

**analyses03: Cronbach error analyses**

The analyses of Cronbach’s alpha for errors were based on Haberman, Brady, and Alvarez (2015). They measure the consistency of participants’ errors within each trial type: pre-cued one-pleasure, pre-cued combined-pleasure, post-cued one-pleasure, and post-cued combined-pleasure.

STEP 1: For each participant, we first split the data into the four trial types described above. We then calculate the mean absolute difference between these ratings and the baseline ratings given by the same participant for each image (for one-pleasure trials) or the average of the baseline ratings for both presented images (for combined-pleasure trials). These mean absolute differences are the error for each participant per trial type.

STEP 2: Next, we calculate Cronbach’s alpha for each trial type using the errors calculated in step 1. We used the MATLAB function cronbach() to do so (<https://www.mathworks.com/matlabcentral/fileexchange/7829-cronbach-s-alpha>). Again, data from participants with missing values in their baseline ratings, for the main data this was participant #1, were excluded from this analysis.

STEP 3: Based on the alpha values in step 2, we calculate the maximum correlation obtainable for the combination of each trial type as the square root of the products of the trial types to be correlated.

**Analyses 04: LOOCV for one-pleasure trials**

Leave-one out cross-validation analyses were conducted for each individual participant separately. We fit three different models based on Eq. 1:

(1)

where is reported pleasure, w is the target weight, 0.5 ≤ *w* ≤ 1, and *a* and *b* are constants. For one-pleasure trials, *P*1 represents target single-pleasure and *P*2 distractor single-pleasure. For combined-pleasure trials, *P*1 represents the left image’s single-pleasure and *P*2 the right image’s single-pleasure.

All models are special cases of Eq. 1. The models of interest for the LOOCV one-pleasure analyses are the accurate observer, compulsory averaging, and partial compulsory averaging models. These models assume no linear transformation, i.e. *a* = 0 and *b* = 1. Crucially, the compulsory averaging model has a weight *w* = 0.5, whereas an accurate observer has *w* = 1. Values for *w* fall in between for the partial compulsory averaging model.

The models were fit on all but the one left-out trial. The models were distinguished by constraining the permissible range of the model parameters *a*, *b*, and *w* as described above. To fit these models we used the built-in MATLAB function fmincon() with a maximum of 108 iterations and function evaluations and a termination tolerance for *x* of 1-20. The fmincon() function finds the minimum of a problem specified by, in our case:

Where *lb* is the specified lower bound and *ub* is the specified upper bound for the variable *x*.

The start value for *w* was set to 0.5. The cost function for the minimization problem was the root mean square error between model predictions and observed values for ratings on individual trials.

For the one-image trials, only the partial compulsory averaging was fit on the training trials. The accurate observer and compulsory averaging model were not fit since they do not contain a free parameter. For each model and each observer, we calculated the average root mean square error (RMSE) between model predictions and observed rating in the left-out test trials.

Last, we calculate the mean RMSE per model across participants as well as the standard error of this mean.

**Analyses 06: LOOCV for combined pleasure trials**

The analyses for combined-pleasure trials followed the same procedure as the analyses for one-pleasure trials with exception of the model specifications.

For combined-pleasure trials, the evaluated models were the accurate observer, compressive, and expansive models. All three models assume *w* = 0.5. The compressive model predicts a slope of 0 < *b* < 1, and an intercept *a* > 0. The expansive model predicts a slope of *b* > 1 and an intercept of *a* < 0. In contrast, the accurate observer model predicts an intercept of *a* = 0 and a slope of *b* = 1.

**References**

Haberman, J., Brady, T. F., & Alvarez, G. A. (2015). Individual differences in ensemble perception reveal multiple, independent levels of ensemble representation. Journal of Experimental Psychology: General, 144(2), 432–446. <http://doi.org/10.1037/xge0000053>

Kurdi, B., Lozano, S., & Banaji, M. R. (2017). Introducing the Open Affective Standardized Image Set (OASIS). Behavior Research Methods, 49(2), 457–470