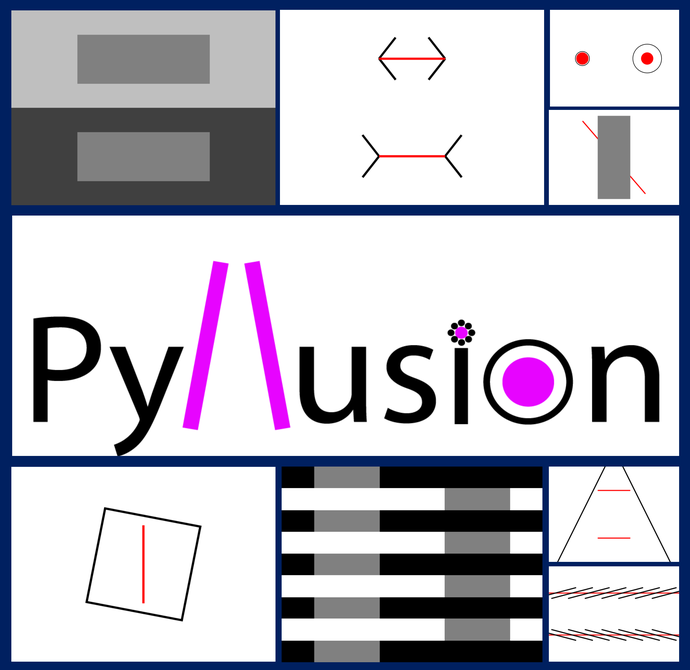
A Parametric Framework to Generate Visual Illusions using Python

Zen Juen Lau[1], Tam Pham[1], W. Paul Boyce[2], S.H. Annabel Chen[1] , & Dominique Makowski[1]
[1] Nanyang Technological University (Singapore), [2]University of New South Wales (Australia)

*An **open-source package** to programmatically generate illusions in Python*



*Many classic illusions, like the, **Ebbinghaus** or the **Ponzo** illusions, are already available!*

https://github.com/RealityBending/Pyllusion
See our documentation here

Background

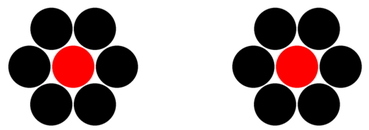
- Visual illusions are powerful tools for investigating human perception and neurocognitive disturbances[1, 2, 4]
- Illusory paradigms manipulate specific stimulus features of an illusion to infer certain perceptual mechanisms (e.g., size of outer circles in the Delboeuf illusion[3])
- Difficulty in replicating experimental designs (on top of other issues such as small number of stimuli in each study) lead to inconsistent findings in illusion research
- Lack of a dedicated software for generating and reporting illusion stimuli for easy reproduction and re-usage by other researchers

Aims

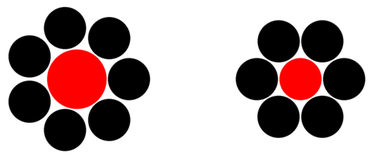
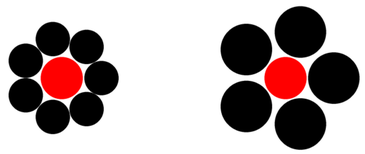
- Facilitate ease of parameter manipulation in illusory stimuli so that results can be interpreted with respect to the central parametric properties
- Encourage efforts to replicate common-use and investigate rarely-tested illusions
- Encourage testing of a battery of different illusions
- Develop new illusions and functionalities based on needs of the community

Design Philosophy

Pyllusion utilizes two main parameters, **illusion strength** and **difference**. For instance, in the **Ebbinghaus illusion:**



illusion_strength=0, difference=0

Parameters	Description	What modulating it looks like
Difference	objective size difference of inner red circles i.e., target features in which participants are supposed to judge	 *illusion_strength=0, difference=1*
Illusion strength	size of outer circle relative to inner circle i.e., strength of surrounding context in biasing perception of unequally sized red circles	 *illusion_strength=1, difference=0*

These terms manipulate different stimulus features depending on the illusion (e.g., lengths in Ponzo illusion, angle displacement in Zollner illusion), but are kept constant across the different illusions for **consistent interpretation of results based on parametric properties.**

Code Demonstration

Step 1: Specify Parameters

Parameters for each illusion can be generated using *`IllusionName()`* function such as *`Delboeuf()`, `Ebbinghaus()`, `MullerLyer()`* etc.

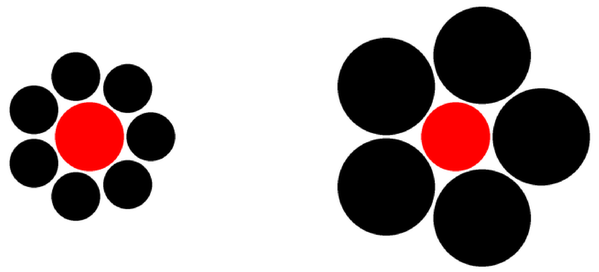
```
import pyllusion # Load package
# Create parameters
ebbinghaus=pyllusion.Ebbinghaus(illusion_strength=2,
                                difference=0)
# Visualize parameters to prepare for
later rendering
ebbinghaus.get_parameters()
```

```
{'Difference': 0,
 'Size_Inner_Left': 0.25,
 'Size_Inner_Right': 0.25,
 'Size_Inner_Difference': 0.00,
 'Illusion': 'Ebbinghaus',
 'Illusion_Strength': 2,
 'Illusion_Type': 'Incongruent',
...}
```

Step 2: Render Illusion

Each class object (i.e., the illusion name) which contains the dictionary of parameters, is rendered as 1) a PIL image (using Pillow library) with *`to_image()`* or 2) a PsychoPy stimuli with *`to_psychopy()`*

```
ebbinghaus.to_image(height=600, width=800)
```



***Note**. The PsychoPy stimuli rendering is shown via psychopy_example_script.mp4*

References

[1] Chen, C., Chen, X., Gao, M., Yang, Q., & Yan, H. (2015). Contextual influence on the tilt after-effect in foveal and para-foveal vision. Neuroscience Bulletin, 31(3), 307–316.
[2] Corbett, J. E., & Enns, J. T. (2006). Observer pitch and roll influence: The rod and frame illusion. Psychonomic Bulletin & Review, 13(1), 160–165.
[3] Delboeuf, J. (1893). Sur une nouvelle illusion d'optique.
[4] Roberts, B., Harris, M. G., & Yates, T. A. (2005). The roles of inducer size and distance in the ebbinghaus illusion (titchener circles). Perception, 34(7), 847–856.