

Summary

We want to present grating stimuli and natural images in the same recording session, in a interleaved manner by having a single folder with both images.

Current problem:

When we construct gratings and images as RGBA png files and and present it in Mworks the contrast is not smooth (slide 2).

Things we've tried:

- The images look fine if we use a different app to visualize (slide 3)
- The built-in gratings with the same mean/std look fine (slide 4)
- If we construct RGB images add a gaussian mask in mwel (using layers) the contrast is still not smooth (slide 5).

bunnies6-RGBA.xml gives weird grating since 2022-10 (at latest)

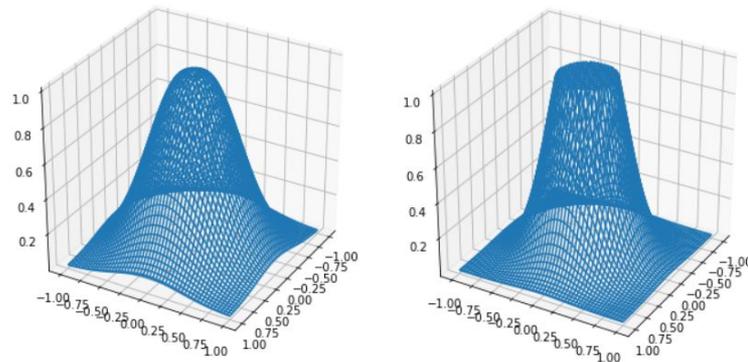
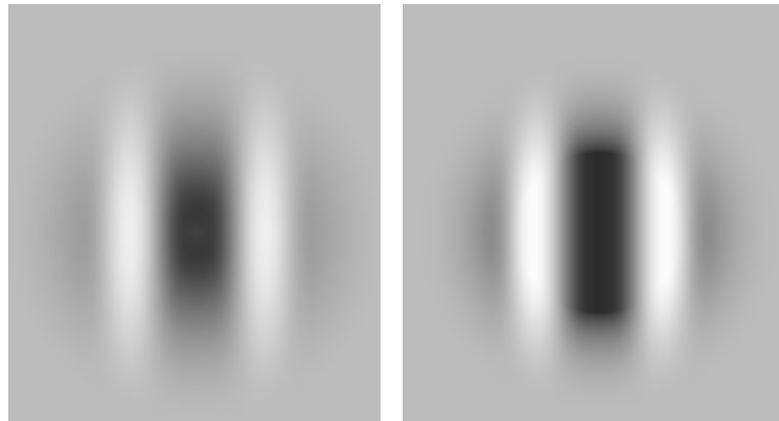
This is a screenshot of how the constructed grating looks like in mworks

Using different gaussian mask (alpha channel of RGBA png) mean and std (see mask_mean and mask_std in grating construction code) for the constructed grating on left vs right side:

Left: mask_mean = 0.1, mask_std = 0.3985

Right: mask_mean = 0.1, mask_std = 0.3

Gaussian mask visualized in 3D: bottom row



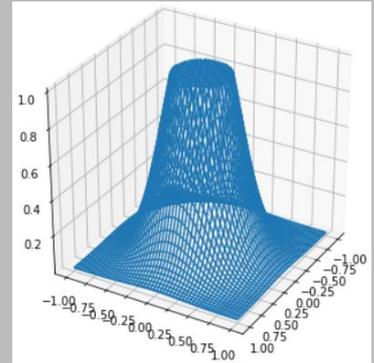
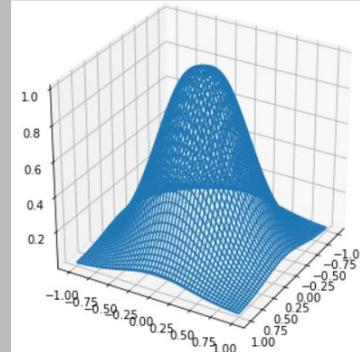
Find xml (mwel converted from xml gives the same stimulus presentation), constructed grating, and grating construction code here: [link](#)

Constructed grating looks fine in other apps, but not mworks

This is how the constructed grating (left and right side corresponding to the previous slide) looks when copied as a png and pasted to google slides

The background color is set to be the same as the screenshot background color in the previous slide: RGB [188,188,188] or hex #BCBCBC

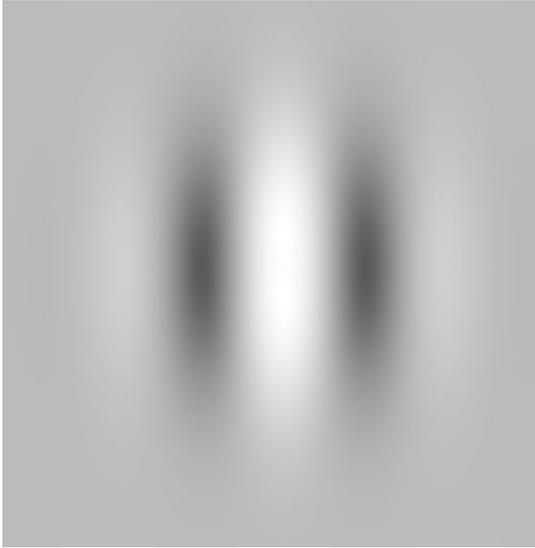
Maybe mworks is rendering the alpha channel of png in a way we didn't expect



built-in grating of mworks looks fine with .mwel

2023-01

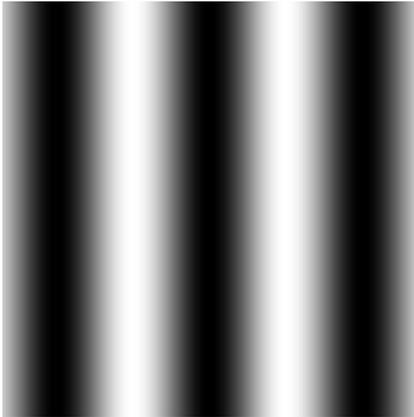
twoStim_2P_Frames.mwel



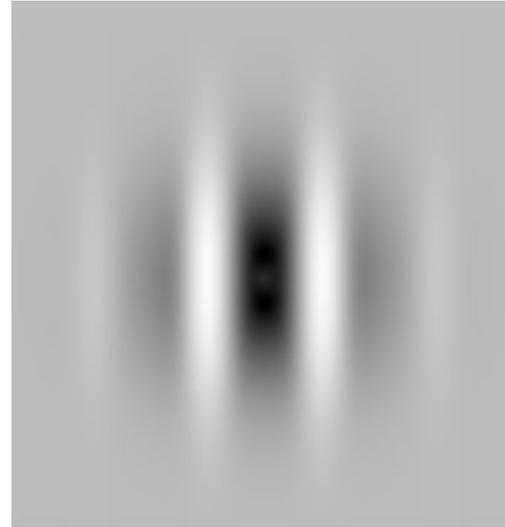
Find xml and mwel here: [link](#)

Using layer-mask function in mworks still results in weird grating

Png placed in `Image` folder: this is the constructed grating with only RGB channels, but without the alpha channel / gaussian mask



Screenshot of what is shown in mworks



Find mwel, constructed grating, and grating construction code here: [link](#)

Using layer-mask function in mworks; see parameters in code

```
79 var maskType = 'gaussian' (  
80   persistent = 1  
81   groups = 'Grating Parameters'  
82 )  
83 var gratingMean = 0.1 (  
84   persistent = 1  
85   groups = 'Grating Parameters'  
86 )  
87 var gratingStd = 0.3 (  
88   persistent = 1  
89   groups = 'Grating Parameters'  
90 )  
91 var gratingEdge = 0.125 (  
92   persistent = 1  
93   groups = 'Grating Parameters'  
94 )
```

```
185 stimulus_group images {  
186   // load image stimuli  
187   range_replicator (  
188     from = 1 // always from 1, when meaningful img start from 1.png  
189     to = 6 // update when using new paradigm. value should be n_img. TODO: update when change stim  
190     step = 1  
191     variable = index  
192   ) {  
193     image_file image_${index} (  
194       path = 'Image/${index}.png'  
195       x_size = xSize  
196       y_size = ySize  
197       x_position = xPosition  
198       y_position = yPosition  
199       deferred = explicit  
200     )  
201   }  
202 }  
203  
204 // make a rectangle with a hole in the center  
205 layer gray_surround { // live_queue_stimulus (gray_surround) after drawing image stim1  
206   rectangle (  
207     color = 0.5, 0.5, 0.5 // gray like background  
208     x_size = xSize  
209     y_size = ySize  
210     x_position = xPosition  
211     y_position = yPosition  
212   )  
213   mask transparency_alpha_mask (  
214     mask = maskType  
215     std_dev = gratingStd  
216     mean = gratingMean  
217     edge_width = gratingEdge  
218     x_size = xSize  
219     y_size = ySize  
220     x_position = xPosition  
221     y_position = yPosition  
222     inverted = true // inverted so that the hole is transparent, showing the stimulus image beneath  
223   )  
224 }
```

Find mwel, constructed grating, and grating construction code here: [link](#)