

CS 1112 Spring 2020 Project 2 Part B due Monday, Feb. 17, at 11:00 PM

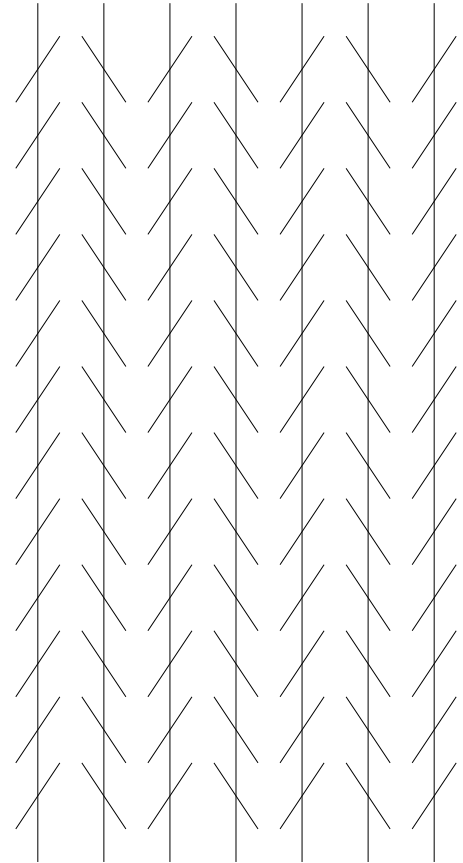
Part A, specifying problems 1 and 2 along with rules and objectives, is in a separate document.

3 Drawing the Zöllner illusion

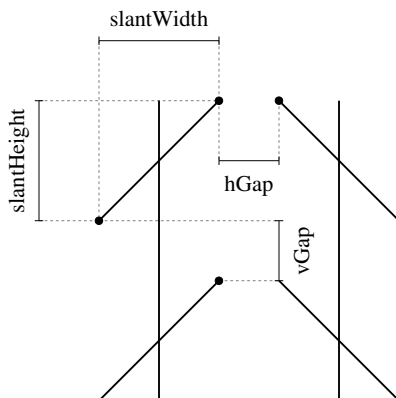
Are the vertical lines in this picture angled with respect to one another, or are they in fact perfectly parallel? For many, the lines appear to slope towards or away from each other, despite them being the same distance apart at the top as at the bottom. This illusion, published in 1860, has implications for how we perceive angles and depth. Your job is to reproduce the illusion using Matlab graphics. But rather than specializing to a single “instance” of the illusion, your code will be *parameterized*, allowing users to customize the length, spacing, and number of lines in order to study their effect on what we perceive.

We have chosen to parameterize the illusion in the following way:

- numCols: The number of vertical lines to draw
- numRows: The number of slant lines drawn on each vertical line (note that the direction of slant in one column is opposite that of neighboring columns)
- slantWidth: The width of each slant line
- slantHeight: The height of each slant line
- hGap: The horizontal gap between slant lines in adjacent columns
- vGap: The vertical gap between slant lines in the same column



The picture above contains 7 columns and 12 rows. See the figure below for a graphical reference to the length parameters. Note that vGap may be negative and the illusion will still work.



Write a script, drawZoellner, that draws the Zöllner illusion in Matlab’s figure window according to these parameters. The parameters must be declared as variables at the top of the script, and changing a parameter should only require changing the corresponding variable initialization (that is, you should not hard-code assumed parameter values in your drawing routines; only use the variable names). For testing, try the following values:

```
numCols = 5;  
numRows = 5;  
slantWidth = 2;  
slantHeight = 2;  
hGap = 1;  
vGap = 1;
```

Use the `plot()` function to draw the lines (refer to Project 1); you may choose any color for the vertical and slanted lines. Use `hold on` to allow the lines to “stack up” in the same figure window, and use `axis equal off` to keep the figure clean and preserve geometry (see Lecture 6). We’re not too picky about how far above and below the slants the vertical lines should extend, but they should not start below the top of the first slant, nor end below the bottom of the last slant, and they shouldn’t extend so far above and below the slants that there’s obvious room for more rows.

Remarks

Parameterization is an important concept in engineering design as well as in computation. A good choice of parameters allows the end result to adapt self-consistently when a single parameter is varied, rather than having to re-tune the entire design by hand. This makes it easier to experiment with different design variations, and it also makes your code more robust and maintainable. Consequently, often your first task when tackling a problem computationally should be to identify a good set of parameters.

A parameterized solution requires more testing than a one-off solution, however. After you have written some code for this assignment, be sure to **test it** with a variety of different parameter values to ensure that the result scales appropriately. As an exercise, what parameter values will reproduce the image at the start of this problem? (You may optionally write your answer in a comment at the bottom of your script, and we’ll let you know how close you were.)

Submission

Submit your file `drawZoellner.m` on CMS (after registering your group), in addition to your files for Part A.