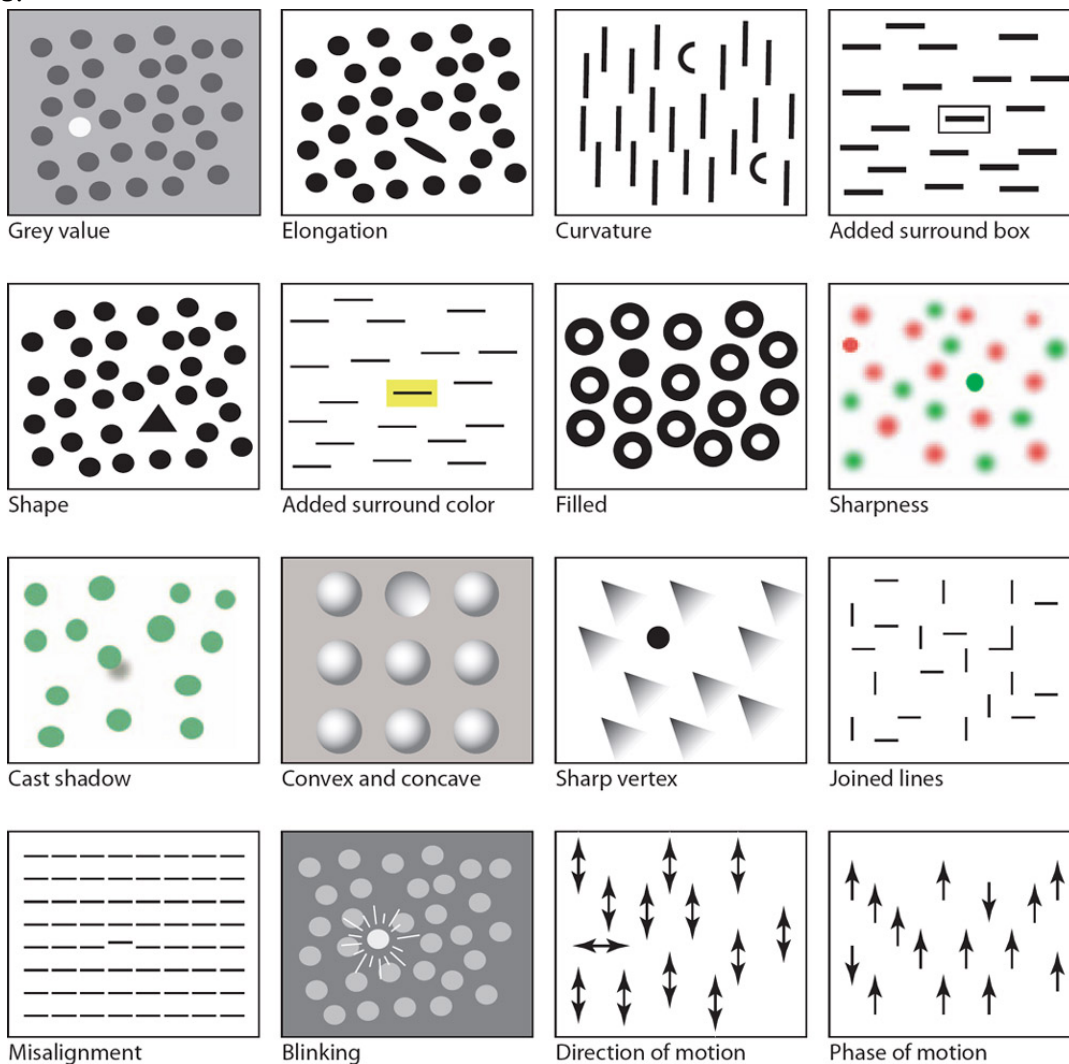


What stands out?

The following figure is from Colin Ware's book *Visual Thinking for Design*. It contains 16 visual features that have been found to trigger pre-attentive processing in your visual cortex. Pick eight of these features and replicate them in SVG using d3. At least one must be animated: blinking, different direction, or different phase. For the blinking animation, change one or more HSL features, do not make radiating lines (unless you want to). For the two that show motion, use circles moving in the way suggested by the arrows, not literal arrows.



1. Present your eight selected visual features in a grid of two rows and four columns, with each feature in its own SVG element with a black 1px border. You do not have to replicate Ware's figure exactly. Use roughly the same number of shapes in each SVG as in the

original, between 20 and 30. You may hard-code their positions, place them in a grid, or place them randomly. You may want to use a collision-avoiding force layout if you use random positions, but make sure no shapes go outside the viewable area. Each SVG should contain *one* randomly selected shape to emphasize, so that reloading the page should (probably) make a different shape stand out in each SVG. You do not need to add axes or axis labels. State under each SVG which visual feature you are implementing. You may use either CSS animation (best bet) or Javascript animation using `d3.transition()` or similar functions. Animations must loop infinitely, not a fixed number of iterations. (80 pts)

2. Create a single large SVG with width and height roughly equal to the width and height of the 2x4 grid in problem 1. Fill it with 150--200 shapes. Select two visual features from Ware's figure. Use feature 1 only to emphasize one randomly selected set of five shapes, feature 2 only to emphasize a different five shapes, and *both* features to emphasize an additional five shapes. You do not need to add axes or axis labels. State under the SVG which visual features you are implementing. You may position the points in the same way you did for problem (1). (15 pts)

3. When you are done with (2), load the page and use a stopwatch to count how long it takes you to find all the shapes in these three categories. In a `<p>` tag under the SVG element, record this number. Was this finding task easy or hard? Were the three categories equally difficult, or were some easier than others? If so, which? (5 pts)