

### 8.3 Autostereograms

Ever seen these 3D posters that work when you stare at them for a while? Did you ever care to know how they work or wanted to make one yourself? Here's your chance!

First, the theory behind autostereograms (very non-scientific); Our brain figures out the depth of objects by comparing the two angles at which our eyes see it. We can fool this mechanism by creating a pattern that is *nearly* periodic in the horizontal direction, and then look at it as if it is closer or further away than it actually is. By staring or looking cross-eyed at the pattern, we make each eye focus on a different copy of the repetitive pattern. However, since the patterns is repetitive the brain assumes that it is looking at object that is located at a different depth. To create an illusion of a 3D object we deviate from the repetitive pattern in the following precise fashion: Each dot in a copy is moved horizontally with respect to the matching dot (in the repetitive pattern). The amount in which the dot is moved is proportional to the desired depth of the illusion at that place. In terms of coding you will need to figure out how to:

- Find an interesting 2D surface from which you would like to create an autostereogram. This should be represented internally as a matrix, picture, or mathematical formula. This doesn't mean that your program should accept all of these as input, but rather that you consider the possibilities and think about converting the input into a map,  $(x, y) \rightarrow z$ .
- Create a random pattern of dots (with about 50% fill) in a narrow vertical rectangle (should be about 1" in width).
- Create a vertical strip of the autostereogram, copy the previous dot pattern and move the dots in proportion to the desired depth at that point. Remember that you need to keep copying the previous strip and not the original one.
- Plot your points to create the stereogram. It is useful to plot two additional large points below your autostereogram to indicate the spacing of repetition. This will help when trying to view the autostereogram.
- Try your program first on a simple input that you know more or less what the answer should look like (for example, a simple square that is lifted from the background). Then try it on other simple geometric objects, like pyramids and spheres. Have a 3D picture in mind and create it.
- If your height function has a large jump, you will find that a big area with no points may be created and this will hurt the quality of the result. To avoid this you can try to locate big empty spaces at each repetition and fill them up with new random points. These extra points should then be copied and moved together with the others.
- Big jumps in the height function can also cause points to "overlap" creating dense "blobs" of points. This, too can hurt the quality of the autostereogram. Removing points that have overlapped (not the points doing the overlapping) is another way to improve your autostereogram.

MIT OpenCourseWare  
<http://ocw.mit.edu>

18.S997 Introduction To MATLAB Programming  
Fall 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.