A Human Eye Retinal Cone Synthesizer

Michael F. Deering
deeringm@acm.org

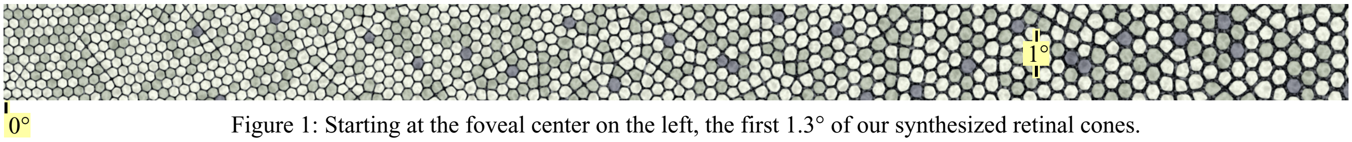

Figure 1: Starting at the foveal center on the left, the first 1.3° of our synthesized retinal cones.

1 Introduction

An important component of a new model of the human eye accurate to the photon level [Deering 2005] is an algorithm to "grow" synthetic retinal cone mosaics. The algorithm must individually model the position, size, shape, and orientation of each of the five million photoreceptor cones in the retina. While not intended to be an exact biological simulation of retinal growth, the algorithm is motivated by what is known about that process and its constraints, and the synthesized mosaics must closely mimic nature's results. Figure 1 shows a 1.3° of visual angle strip of our results.

Cones near the center of the retina actively move closer to the center from several months before birth to four years of age! It is this migration and packing pressure that forms the unique mosaic pattern. We model this by the simulated growth cycles of groups of cones. There are two halves to each growth cycle. In the first half, cone cells are represented by points whose locations change in response to energy functionals: a force toward the center of the retina, and an inverse exponential repulsive force between any two cones that approach too close. If two cones are "forced" too close together, one of them is deleted. In this first half cycle, cone cell wall information is ignored; it is the job of the second half of the growth cycle to re-create these walls from the center point information. One possible algorithm to perform this computation is that of Vornonoi cells, but it has several biologically undesirable features. First, it allows cell walls to form arbitrary far from cell center points, ignoring the constraint of maximum cone cell size. Second, its rule for forming corners in cell walls does not balance the internal pressure of the cells. And finally, it forms cell wall corners based only on three points, while real cell wall corners are frequently the point of mutual contact of four or five cells.

In order to address these limitations, a new algorithm for forming cell walls given a set of cell center points was designed, and the rest of this document will be devoted to a brief description of it.

2 Cone Cell Wall Construction Algorithm

Two cones are deemed neighbors (N[]) if the distance between their center points is less than 1.5× the sum of their birth radius targets. For each cone p, a spatially indexed data base of cones is searched to find all others cones n_j such that $N[p,n_j]$. The neighbors are sorted into clockwise order about p, as shown in Figure 2a. Then new cell wall edge vertex creation pattern matching rules are applied all around the circle of neighbors, resulting in a polygonal cell wall, as shown for the six sided case in Figure 2f.

For each successive neighbor n_i around p, rules are tried involving 5, 4, 3, or 2 consecutive neighbors (most to least complex rules). Figures 2b-e graphically show some of these rules.

In the simple case of Figure 2b, cones p, n_i, and n_{i+1} are all mutually neighbors. The new cell wall edge vertex e_j is created by simply averaging the center points of the three cones. The dashed magenta lines show where cell walls will connect to e_j; the particular direction of the walls will depend on the positions of other cell wall edge vertices. (The same convention is used in Figures 2c-e).

Figure 2c shows the contrasting case in which n_i and n_{i+1} are both (by definition) neighbors of p, but unlike in Figure 2b, not of each other. p will share a new cell wall edge vertex e_j with n_i that is created by averaging the center points of just these two cones; e_{j+1} is created similarly just between p and n_{i+1}. The cell wall edge between e_j and e_{j+1} (shown in red) will be the edge of a (usually temporary) void in the retinal cone mosaic.

Figure 2d shows what happens when three consecutive neighbors of p are mutually neighbors *and* n_i or n_{i+2} is closer to p than n_{i+1}: all four cones will share a new cell wall edge vertex e_j, the average of all four cone center points. The key here is that n_i and n_{i+2} are neighbors, and n_{i+1} is further away from p. By symmetry, all four cone cell walls will push out together to meet at a single point. The same logic applies with five mutual neighbors (not shown). In rare cases (Figure 2e), five cones will not be mutual neighbors, but a complex test shows they should share only one new cell wall edge vertex.

After forming new cell walls for each cone, cell walls marked as having voids on one side can be connected together to discover polygonal voids in the mosaic. Any voids encountered that are similar in or of larger area than that of the local cones are seeded with a new cone. In practice the vast majority of voids are just areas where the (code that estimated the) local birth density of cones was too low, and the void filling is just back-filling for this. Specifically, the interesting "breaks" in the regular hexagonal pattern of the cones are cause by the packing process, not by void filling. The last stable cell wall created for a cone becomes its final output polygon. (In biological terminology the cell *border* of animal cells.)

3 Implementation

The retinal cone synthesizer is written in Java, ~10,000 lines long. On a modern processor, it synthesizes ~100,000 cones per hour, so growing a complete 5 million cone retina takes about 2 days.

References

DEERING, M. 2005. A Photon Accurate Model of the Human Eye. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005)*.

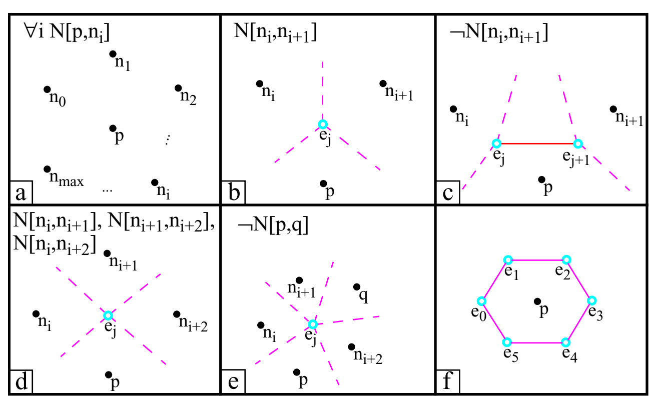

Figure 2: Patterns for new edge cell vertex e_j creation.