SUMMARY FOR 'A GENETIC ALGORITHM TO FIND EFFECTIVE VISUAL STIMULI'
BY J.A.F. Heimel, S.D. Van Hooser, E. de Jong and S.B. Nelson

When exploring the functional properties of neurons in the visual system, one encounters the problem that the dimensionality of the input space is much too large for complete sampling. To reduce the complexity of the task, one often tries to find just the stimulus or stimuli that drive the neuron most effectively. If the neuron were a linear system, one could then derive the response to all stimuli, but even if this is not the case, this would still provide some information about the computation performed. Finding an `optimal' stimulus, however, still requires a search in a very high dimensional space.

In the early stages of the visual system, from retina up to V1, two traditional approaches to tackle this problem are used with reasonable success. One approach is to assume that the neuron computes a linear convolution of the visual image over space and time. After showing Gaussian white noise or basis elements of the subspace to be searched, the stimulus spike triggered average would give the convolution kernel (this is also known as the reverse correlation method). Two big problems with this approach are its linearity assumption and the question of finding a set of basis functions that sufficiently cause the neuron to fire.

Another, often used, approach is to constrain the search space heavily by picking a basic stimulus that is known to be an effective driver and to vary its properties one by one. In the striate cortex, for example, one would present a drifting grating with an a priori estimated spatial and temporal frequency at various orientations. Once the optimal orientation has been established, one would continue at this particular orientation to find the optimal spatial frequency and so forth. Although by design this method will always produce an `optimal' stimulus, this stimulus would be restricted to the class that one was presenting. One would have no idea at all about the properties that are not tested for. Also, it is by no means certain that optimizing the parameters one by one, will eventually lead to a even locally optimal set of parameters.

To try to circumvent these problems, we have designed and implemented a stochastic search through visuo-temporal space by means of a genetic algorithm. At the start of the experiment, a set of short video clips (in the order of a few hundred milliseconds) is generated from stimulus elements. The elements are shapes (e.g. ovals) of varying size, color, position, speed, orientation, eccentricity, time of onset and duration. This set of video clips is shown and the response of the neuron is recorded. The response is converted into a fitness value by taking the spike count or the peak latency. A selection from this population of stimuli is made based on these fitness values. The elements of the members of this selection are mixed using

a cross-over procedure and are randomly altered. The resulting set of new video clips is again shown and their fitnesses evaluated. This process repeats itself until the top members of the newer generation of stimuli no longer consistently have a higher fitness than their `parents'.

We have applied this genetic algorithm to several standard models of neurons. For example, for a Poisson spiking linear-nonlinear model of a center-surround type retinal ganglion cell or LGN neuron (like a cat X-cell) with a slightly delayed surround, the simulations show that within a reasonable amount of time the methods produces a stimulus that includes both the center and a surround. Likewise, for a Gabor kernel model of a V1 simple cell, the method produces a stimulus with all the main on- and off- patches represented. We also plan to try the method on a model for a complex cell, as this would be particularly interesting, because its non-linearity leaves the standard spike triggered average method useless.

Preliminary testing of our method in the visual cortex of the gray squirrel show very similar results. We have seen that in some occasions the algorithm very quickly (i.e. within 15 minutes) and consistently produces a highly effective stimulus, which roughly agrees with properties found using the more conventional methods. We will also apply the method to the neurons in the LGN and show how our method can easily be adapted to find effective stimuli for driving multiple neurons simultaneously during multielectrode recording.