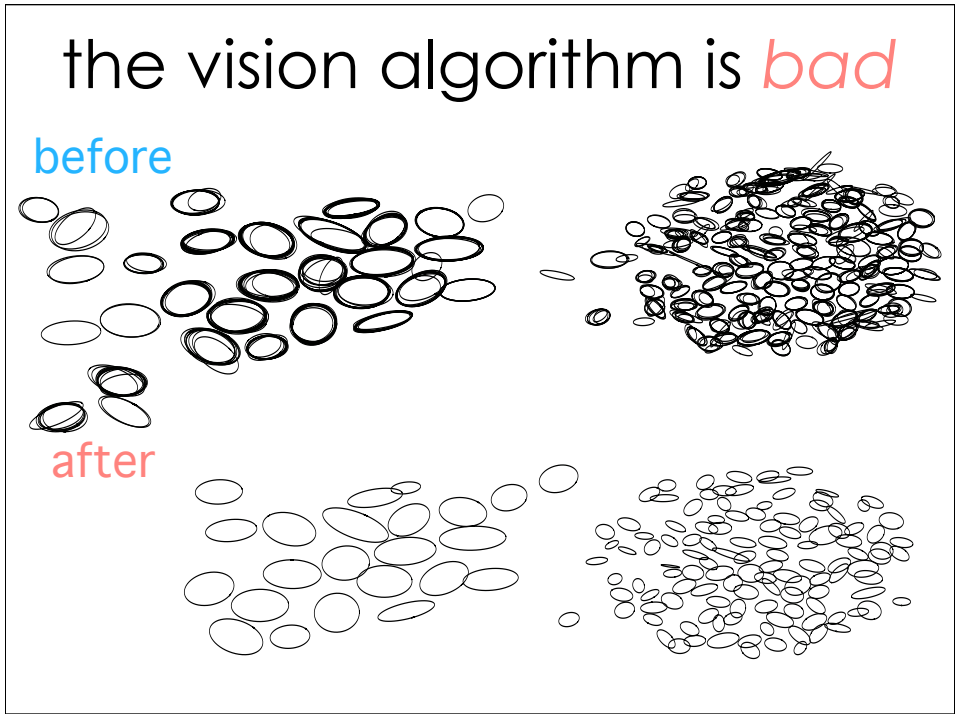
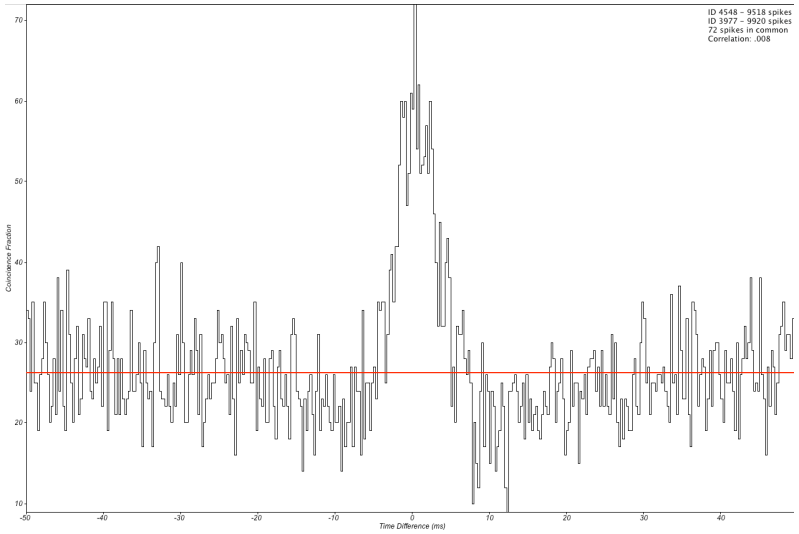
these are slides i was going to present about duplicate removal. i never had time to finish this presentation or present it, so i've included it in this directory such that it might be useful for someone who wants to improve duplicate removal



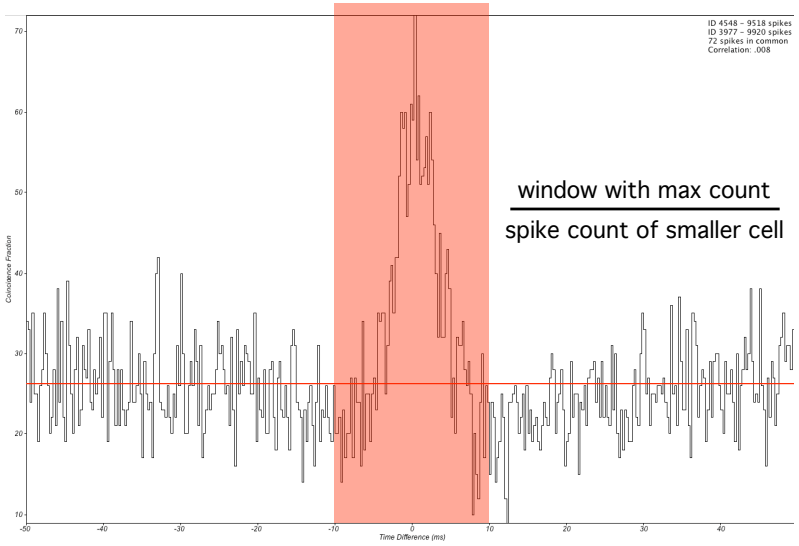
as you can see from a typical dataset like this one, there are a large number of cells that were found during spike sorting (before) that are erroneously removed during duplicate finding (after). this is due to the fact that the vision algorithm for duplicate removal is rather simple, and too aggressive at removing cells

the vision algorithm is *bad*



this figure shows the cross correlation between two cells that will be labeled “duplicates” (seen in vision when you open a .neurons file)

the vision algorithm is *bad*

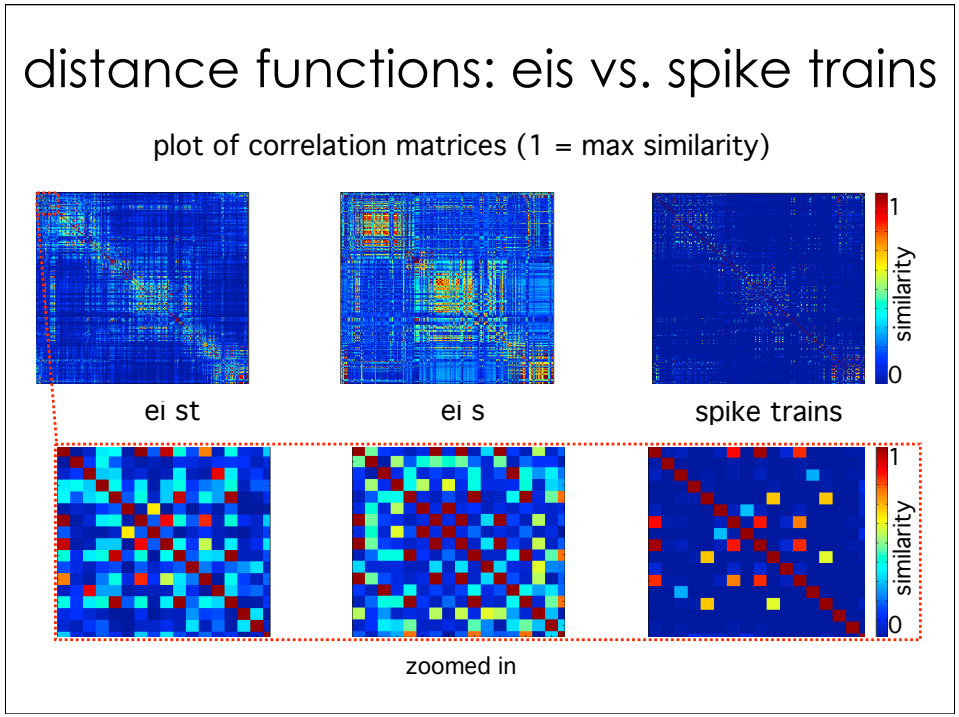


vision decides whether these cells are duplicates by binning up a cross correlation function like this one. if the number of spikes in a single bin (the default bin size is roughly the size of the red box) divided by the spike count of the smaller cell in the pair exceeds a threshold value, the cells are labeled as duplicates of each other. this is literally all vision does.

two problems to fix

- deduplication algorithm
- distance function

vision's algorithm is problematic for multiple reasons: first, it is order dependent. after a duplicate pair is found, one of the two cells is immediately discarded--meaning the discarded cell can no longer be compared to any others. additionally, because the function used to compare spike trains does not create equivalence classes, the fact that the algorithm is order dependent is a big problem (since the output will significantly change if you shuffle the order of the input cells). these problems need to be fixed by a) using a better deduplication algorithm (such as one that merges cells), and b) using multiple distance metrics to evaluate the similarity between cells.



these plots compare the correlation matrices between all spike trains in a dataset when correlating: a) spatiotemporal eis, b) only the spatial component of eis, and c) only spike time information. you can quickly see that there are a lot more nonzero entries in the ei matrices, possibly indicating that there is more information. following this is a discussion of which distance function works the best.

cases:

- I. A and B are duplicates
- II. A and B are not duplicates
- III. A and B are disjoint subsets of a cell
- IV. A is from two cells; B is one of them

when we are comparing two cells A and B, a comparison between them must yield one of the cases above. a good distance metric should be able to identify all of these cases.

multiple distances needed?

- I. A and B are duplicates
  - ei distance ▸ spike distance
- II. A and B are not duplicates
  - ei distance ▸ spike distance
- III. A and B are disjoint subsets of a cell
  - ei distance ▸ spike distance
- IV. A is from two cells; B is one of them
  - ei distance ▸ spike distance

same different

perhaps we need to use multiple distance metrics in order to identify all four cases. because of the way each technique works, we expect to observe the above pattern (blue triangles indicate that the distance metric works correctly for that case)

A,B & C are duplicates

on parasol

	e	spikes
A:	02	11162
B:	18	12380
C:	10	1705

	spike times*	ei (spatial)	ei (spatiotemporal)
AB	0.9266	0.9989	0.9846
AC	0.3599	0.9987	0.9891
BC	0.3571	0.9993	0.9557

expected

*vision numbers tend to be higher

here we show correlation values in the table. blue cells indicate a “correct” answer (ie duplicate pairs are correctly identified as such). darker blues and reds indicate a more confident value (closer to 1 or 0). unlike we predicted, it seems like spike times alone didn’t do a good job identifying some of the duplicate pairs.

A,B & C are duplicates

on midget

e spikes

A: 300 / 6838

B: 274 / 9820

C: 233 / 2447

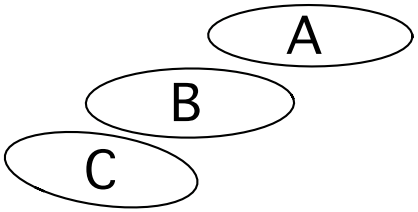
	spike times	ei (spatial)	ei (spatiotemporal)
AB	0.7710	0.9920	0.9623
AC	0.4627	0.9816	0.9805
BC	0.4606	0.9739	0.9369

expected

same dataset as previous page, but with on midget cells

A,B & C are not duplicates

on parasol



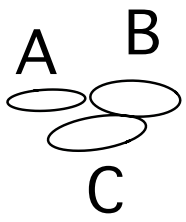
	spike times	ei (spatial)	ei (spatiotemporal)
AB	0.0678	0.1101	0.0750
AC	0.0159	0.0836	0.0008
BC	0.0719	0.0827	0.0383

expected

all metrics did well identifying these three cells as not being duplicates

A,B & C are not duplicates

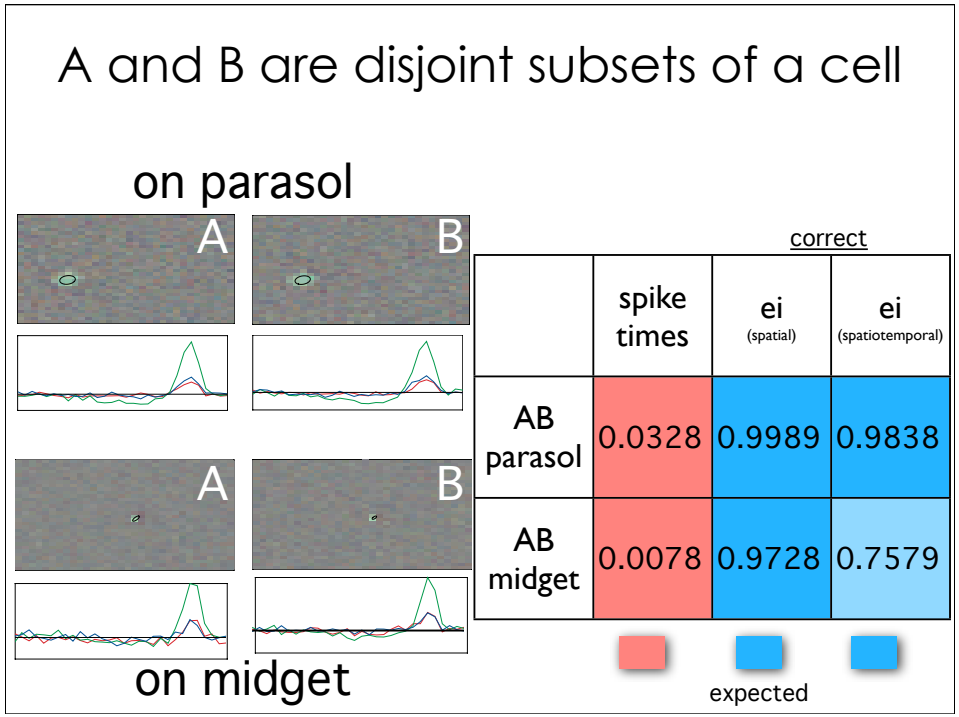
on midget



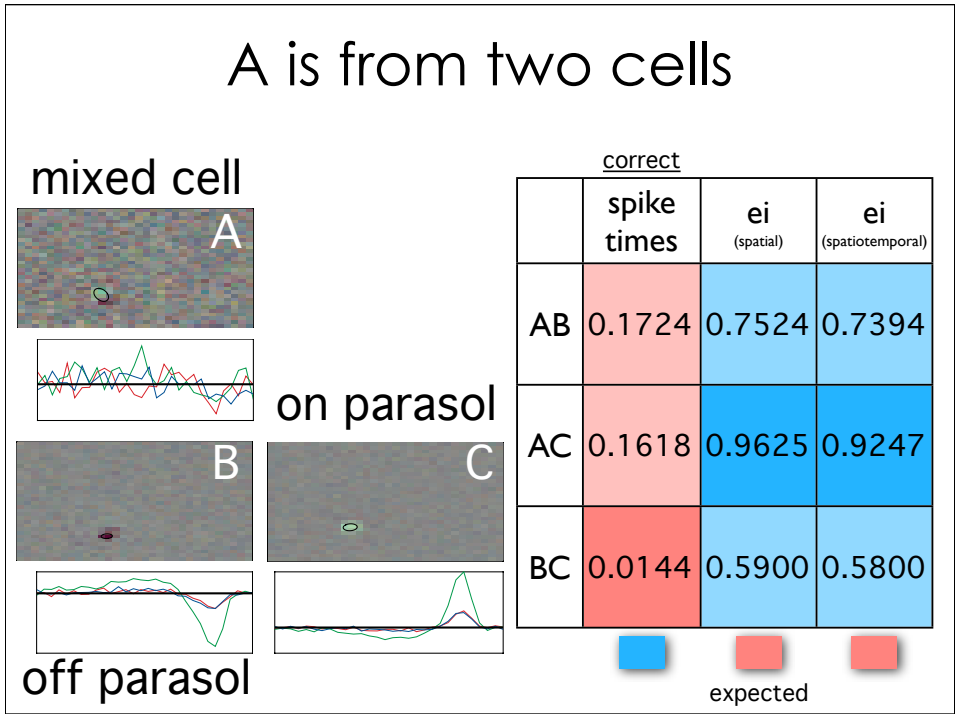
	spike times	ei (spatial)	ei (spatiotemporal)
AB	0.0351	0.3138	0.0706
AC	0.0277	0.1860	0.1241
BC	0.0513	0.5499	0.3123

expected

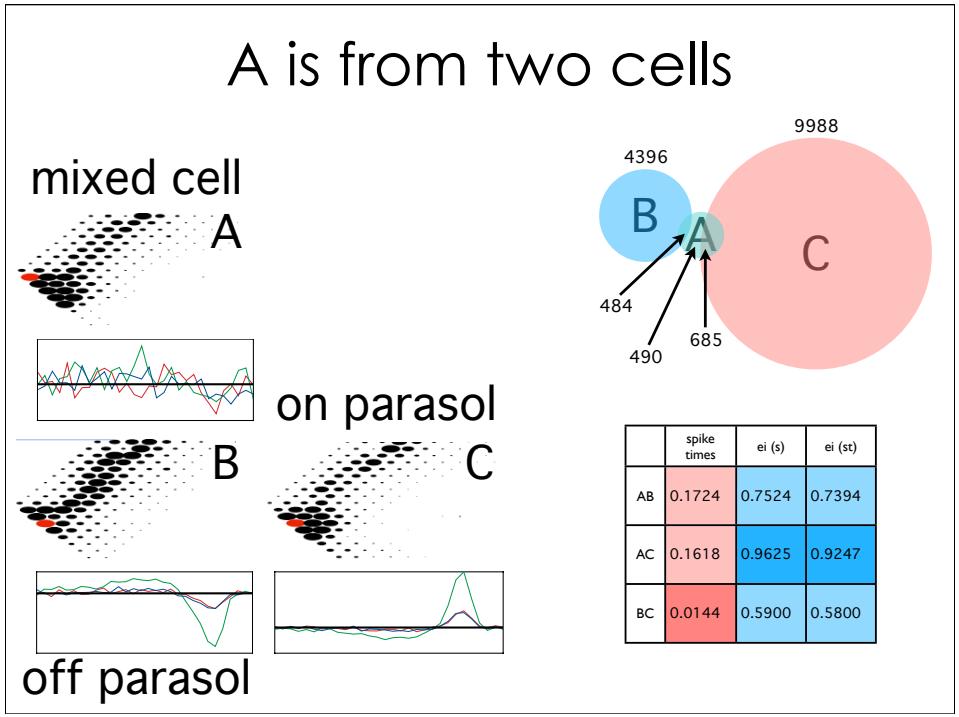
however, in these three cells, the similarity between these cells was significantly higher than you might expect when using eis. this is likely because the three cells were found on similar sets of electrodes. from this, one might expect ei duplicate removal to work more poorly on 30μm board datasets because the electrode density is higher than on the 512 boards.



as expected, when a single cell is split into two, looking for shared spike times will not reveal that these cells are duplicates of one another. comparing the eis of these two cells, it is very clear that they are duplicates of one another.



however, if we examine the case where cell A contains the spikes from two cells, comparing eis erroneously shows cells AB and AC as duplicates of each other. it also shows B and C as duplicates of each other, likely because they are simply overlapping cells found on similar sets of electrodes. comparing spike times generates more reasonable results.



here the eis of the three cells from the previous slide are shown. it's clear from this why looking at eis alone is inadequate.

this is the last slide in this document. hopefully, this gives you some ideas about the issues involved in duplicate removal. clearly multiple distance metrics (eis and spike times) must be employed, but it is unclear how to combine them optimally.