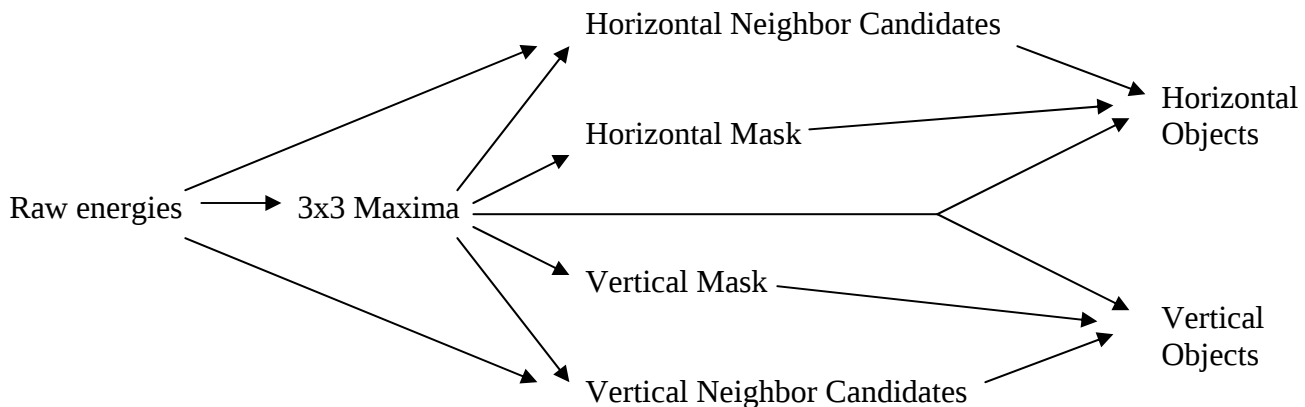


## Overview

Here's a dataflow diagram of the algorithm:



Since a cluster is defined as two adjacent pixels in the vertical or horizontal direction, it is necessary to have parallel detection in the vertical and horizontal direction – same as in previous algorithms.

The maxima table is also taken over from previous algorithms. But this table by itself is insufficient as shown below:

|   |   |   |
|---|---|---|
|   | 4 |   |
| 1 | 9 | 3 |
|   | 2 |   |

|   |   |   |
|---|---|---|
|   | 4 |   |
| 1 | 9 | 5 |
|   | 2 |   |

9 is found as a maximum and the vertical algorithm chooses 4 over 2 to combine it with the 9 and form a cluster. But it also needs to know that if there is a 5 in the horizontal direction, there can be no vertical cluster. So there must be some interaction between the vertical and horizontal detection schemes. **We solve this by introducing a table of “second maxima” – detecting the pixel that is adjacent (but not diagonally) to a maximum and is its maximum neighbor** – this is done in the V/H Neighbor Candidate tables. Finally, maxima and their best neighbors are combined to form clusters in V/H Objects. The V/H Masks exist only to prevent the following situation (they would be unnecessary if maxima were >2 pixels apart):

|   |   |   |   |   |
|---|---|---|---|---|
|   | 1 |   | 1 |   |
| 1 | 9 | 4 | 9 | 1 |
|   | 1 |   | 1 |   |

Here, the 4 is the best neighbor of both of the 9's. The algorithm would like to generate two overlapping clusters – “9 4” and “4 9”. The V/H Masks inhibit the “4 9” cluster and a “9 1” cluster is generated instead. A more detailed description of the individual steps follows.

### 3x3 Maxima

The algorithm takes the raw energies and generates a table which detects the maxima of 3x3 squares as in the previous algorithms. This involves 8 comparisons of each cell with its immediate neighbors.

|        |        |     |
|--------|--------|-----|
| $\geq$ | $>$    | $>$ |
| $\geq$ | $=$    | $>$ |
| $\geq$ | $\geq$ | $>$ |

The sharp and dull inequalities try to prevent having two maxima in the same 3x3 square.

### Horizontal and Vertical Neighbor Candidates

Since a vertical neighbor candidate can be either above or below a maximum, the existence of one is detected as the OR of these two situations:

|     |        |     |
|-----|--------|-----|
|     | $\geq$ |     |
| $>$ | M      | $>$ |
|     | $=$    |     |

|     |        |     |
|-----|--------|-----|
|     | $=$    |     |
| $>$ | M      | $>$ |
|     | $\geq$ |     |

= - the location in Vertical Neighbor Candidate table which we're looking at

M - a maximum in the 3x3 Maxima table

So in order to determine whether a pixel is a vertical neighbor candidate, 6 comparisons have to be made in the Raw Energy table and two bits from the 3x3 Maxima table need to be considered.

The Horizontal Neighbor Candidates are an OR of these two situations:

|        |        |     |
|--------|--------|-----|
|        | $\geq$ |     |
| $\geq$ | M      | $=$ |
|        | $\geq$ |     |

|     |        |        |
|-----|--------|--------|
|     | $\geq$ |        |
| $=$ | M      | $\geq$ |
|     | $\geq$ |        |

So combined, the comparisons for the horizontal and vertical direction look like this:

|     |        |     |
|-----|--------|-----|
|     | $\geq$ |     |
| $>$ | M      | $>$ |
|     | $=$    |     |
| $>$ | M      | $>$ |
|     | $\geq$ |     |

|        |        |     |        |        |
|--------|--------|-----|--------|--------|
|        | $\geq$ |     | $\geq$ |        |
| $\geq$ | M      | $=$ | M      | $\geq$ |
|        | $\geq$ |     | $\geq$ |        |

The sharp inequalities in the vertical comparison were introduced to resolve this situation:

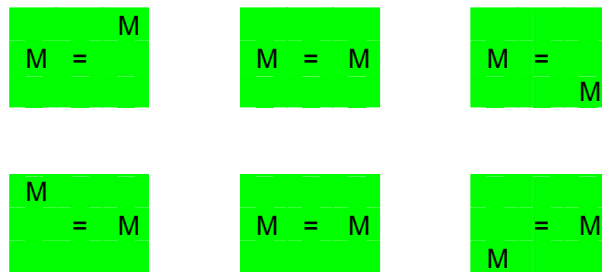
|   |   |   |
|---|---|---|
|   | 1 |   |
| 1 | 7 | 1 |
|   | 1 |   |

Here, all four ones would like to be top neighbor, but due to the sharp inequalities, **the left and right will have priority over the top and bottom ones**. To favor the vertical direction, rotate the pre-previous figure by 90 degrees and use the result as the comparison rules.

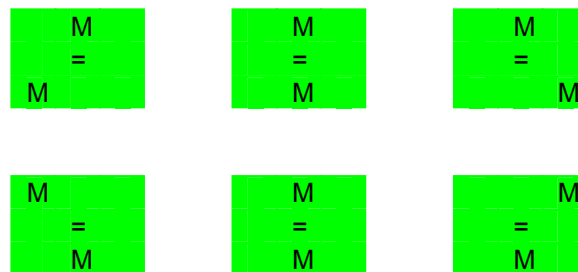
## Horizontal and Vertical Masks

As mentioned in the outline, the H/V Masks are used to detect and inhibit a condition where two maxima are too close to each other. As of now, the masks serve only this purpose, **but more generally, these masks should be used to resolve any situation where for some reason we don't want a cluster to appear.**

The horizontal mask will be zero if any of these situations happens:



Obviously this lends itself well to optimization. The vertical mask is zero if:



## Horizontal and Vertical Objects

Probably the most complicated part of the algorithm is the final detection of clusters. **The first rule is that a cluster's reference point can't occur where the mask is 0** (that's what the masks are for). So if the mask is 1, we can continue. The resulting cluster reference point is supposed to appear in the bottom or left half of the cluster, even if that is not the location of the maximum. This means that there are two basic situations where a cluster should appear (we'll only look at the vertical situation; the horizontal can be obtained by rotating things clockwise by 90 degrees):



M – maximum

C – vertical neighbor candidate

arrow - pixel being investigated

blue square - the mask for that pixel

A problem occurs if we have the following situation:

|   |   |   |   |   |
|---|---|---|---|---|
| 1 |   |   | 1 |   |
| 7 | M | C | 1 | 1 |
| 1 |   | C | 1 | 1 |

Raw Energies, 3x3 Maxima, Vertical Neighbor Candidates, Vertical Masks, Vertical Object

In this case we would get two overlapping clusters. To avoid this, we have to adjust the second condition in such a way, that it becomes active only if the vertical neighbor candidate is unique (there is no candidate on the other side of the maximum).



We still have to add one more condition to solve the following problem:

|   |   |   |   |   |
|---|---|---|---|---|
| 1 |   |   | 1 |   |
| 7 | M |   | 1 |   |
| 3 |   |   | 0 |   |
| 7 | M | C | 1 | 1 |
| 1 |   |   | 1 |   |

Raw Energies, 3x3 Maxima, Vertical Neighbor Candidates, Vertical Mask, Vertical Object

Here, the lower arrow produces a cluster, but the top arrow is inhibited by the mask being 0. So we have to make “7 1” a cluster, even though it is not the maximal combination (which would be “3 7”). **So after adding this third (complicated) condition, we get these three:**

