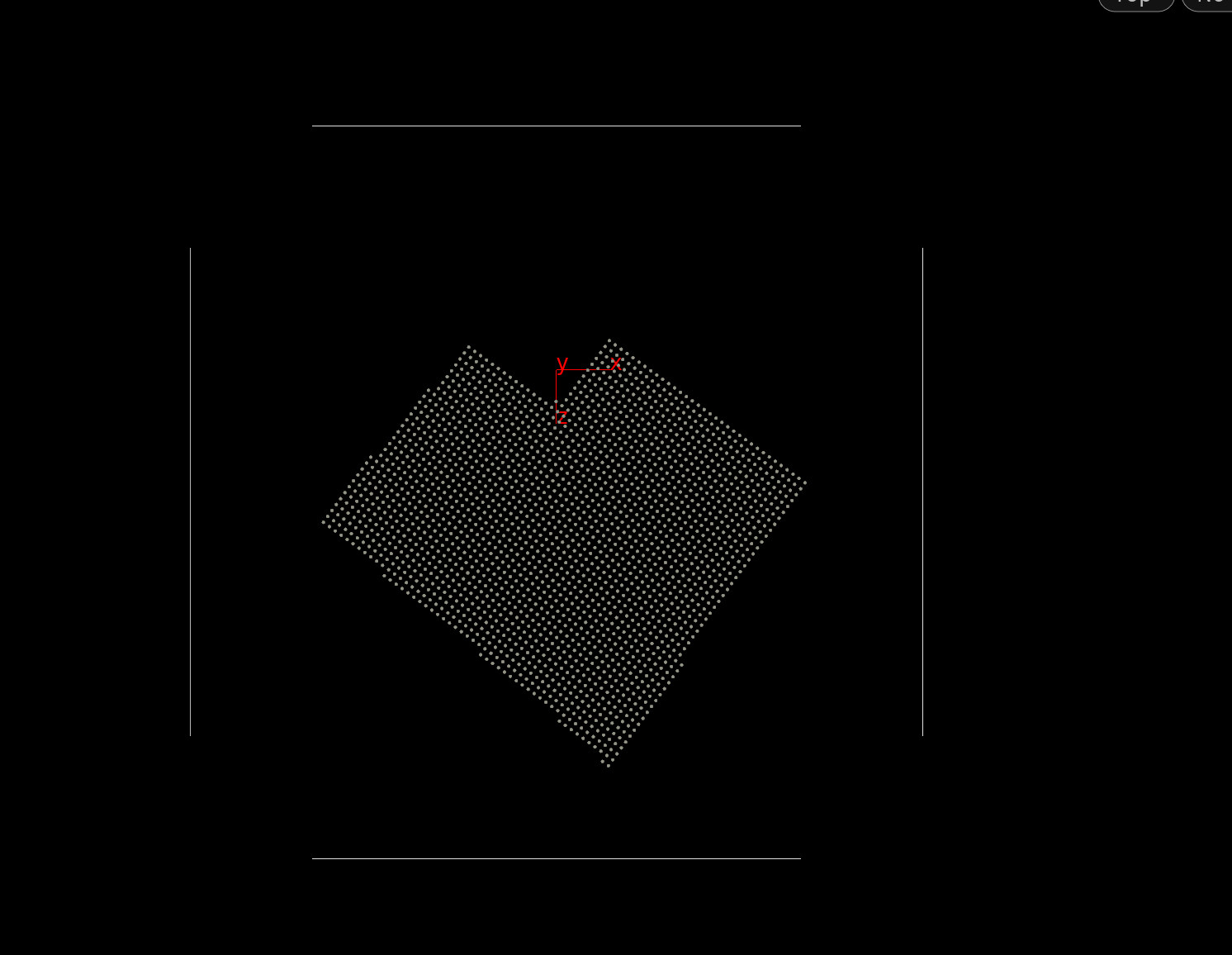
PROXIMITY TO N-E-S-W

//Date: 19-12-2019

//Authors: Jolt Wiersma, Michelle A. Menkiti, Arthur Masure

//Purpose: To create and set normalized attribute values for all voxels, based on position relative to North, East, West and South.

//The following code details the computation required to set attribute values based on their position relative to the North. The code can be very simply adapted to compute values relative to the South, East and West.



**WRANGLER 1**

//INPUT 0 = ENVELOPE POINTS

//INPUT 1 = PRIMITIVE NORTH LINE

//RUN OVER DETAIL

//getting end points of line

vector start\_point = point(1,"P",0);

vector end\_point = point(1,"P",1);

//measuring distance between point and closest point on line

float dist = ptlined(start\_point,end\_point,@P);

//set distance to new attribute

setpointattrib(0,"proximity\_north",@ptnum,dist,"set");

//for every voxel, measure the distance to every point in line

//get smallest distance

//set as new attribute

//number of voxel points

int num\_v\_points = npoints(0);

//number of line points

int num\_l\_points = npoints(1);

//going through all points

for ( int point = 0; point < num\_v\_points; point++)

{

//initialize shortest distance

float short\_dist = 10000000;

//get current point coordinates

vector point\_pos = point(0,"P",point);

//check for distance for all points

for ( int line\_point = 0; line\_point < num\_l\_points; line\_point++)

{

//get line point coordinates

vector point\_line\_pos = point(1,"P",line\_point);

//measure distance

float curr\_dist = distance(point\_pos,point\_line\_pos);

//if point smaller, replace shortest distance

if (curr\_dist < short\_dist)

{

short\_dist = curr\_dist;

}

}

//set point attribute

setpointattrib(0,"proximity\_path",point,short\_dist,"set");

}

**WRANGLER 2**

//normalize distance

int num\_points = npoints(0);

//set temporary maximum and minimum value

float max\_north = point(0,"proximity\_north",0);

float min\_north = point(0,"proximity\_north",0);

//iterate through all points

for (int point = 0; point < num\_points; point++)

{

//get north distance at that point

float north\_value = point(0,"proximity\_north",point);

//if larger or smaller than current min and max value, replace

if (north\_value > max\_north) max\_north = north\_value;

if (north\_value < min\_north) min\_north = north\_value;

}

//iterate through all points

for (int point = 0; point < num\_points; point++)

{

//get north distance at that point

float north\_value = point(0,"proximity\_north",point);

//fit value between range 0.001-0.99

north\_value = fit(north\_value, min\_north, max\_north, 0.001, 0.999);

//set new attribute value

setpointattrib(0,"proximity\_north",point,north\_value,"set");

}