Algorythm of grid-based Mls smoothing

Data: OldLevelSet - given level set values, SimVal- similarity value, MCgrid - marching cubes grid
Result: NewLevelSet
begin
  for $vertice \in MCgrid$ **do**
    verticeCurvature = curvature(vertice);
    if $|verticeCurvature| < 0.5$ **then**
      kernelOffset *= 2;
      kernelSize *= 2;
    end
    else if $|verticeCurvature| < 1.5$ **then**
      kernelSize *= 2;
    end
    samples = getNeighbors(vertice, kernelSize, kernelOffset, SimVal)
    NewLevelSet[vertice] = applyMlsCorrection(vertice, samples, kernelSize, kernelOffset)
  end
end

getNeighbors Algorythm of grid-based Mls smoothing

Data: Vertice - vertice around which neighborhood is found, SimVal- threshold of how far sdf value should be for neighbor vertices, KernelSize, KernelOffset
Result: NeighborVertices
begin
  baseSdf = getSDFvalue(Vertice);
  for $i, j, k \in -kernelSize, kernelSize$ **do**
    neighborVertice = baseCell + Vector(i * kernelOffset, j * kernelOffset, k * kernelOffset);
    sdfValue = getSDFvalue(neighborVertice);
    if $|sdfValue - baseSdf| > SimVal$ **then**
      continue;
    end
    $NeighborVertices.push_back(neighborVertice)$
  end
  return NeighborVertices
end