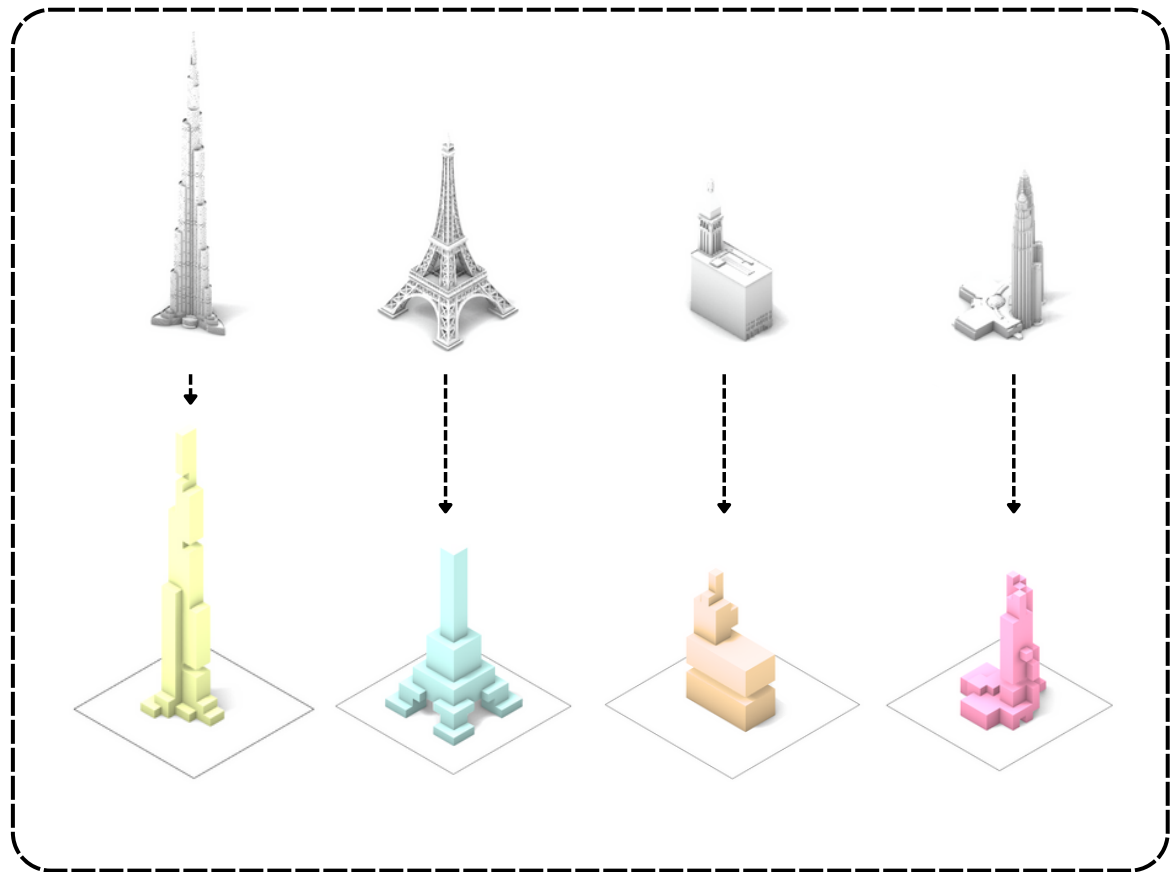
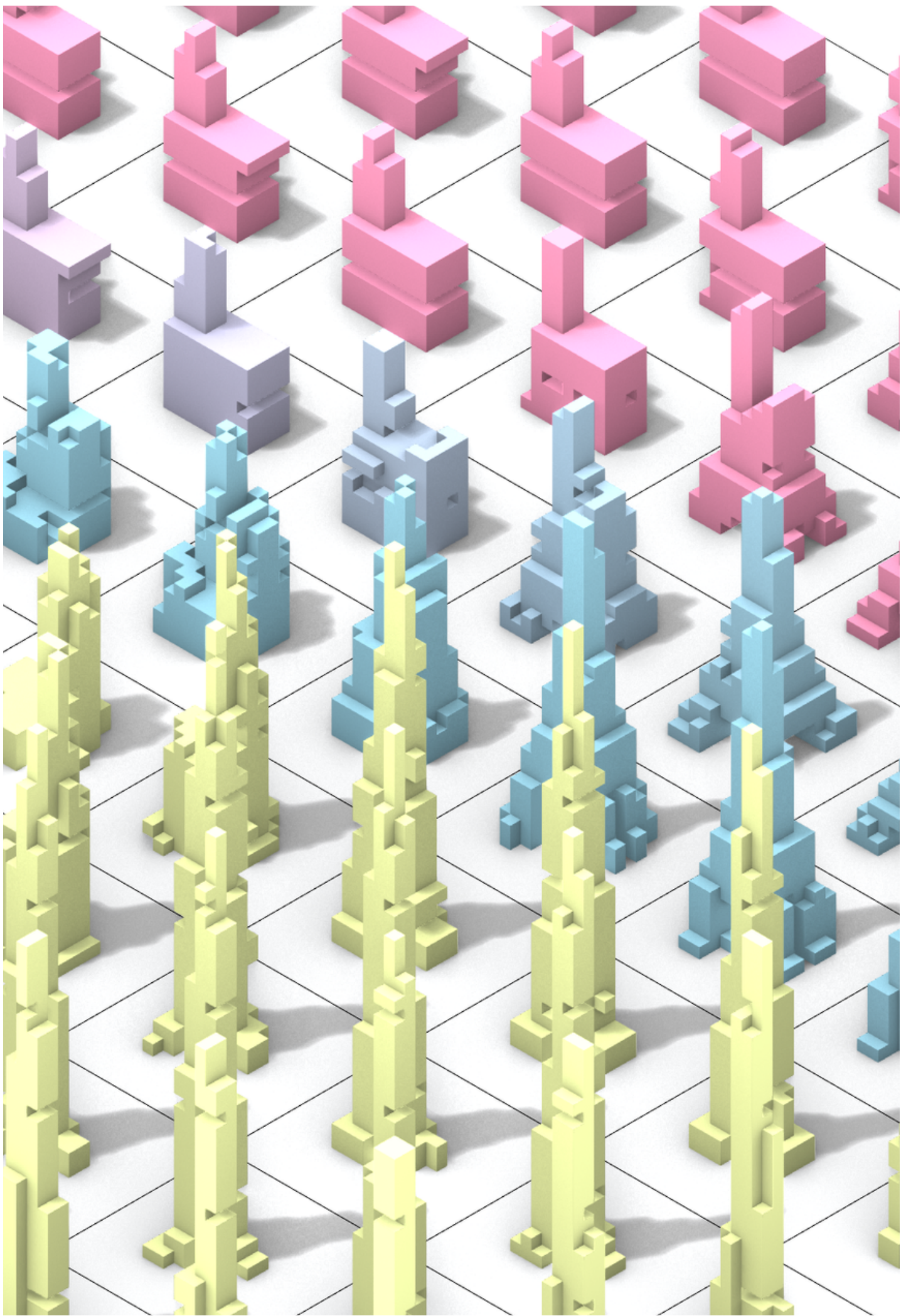


Introduction Board

Abstract:

This project presents a custom-built morphogenetic system that utilizes a Self-Organizing Map (SOM) to interpolate between iconic architectural mesh forms and cluster them according to geometric similarity. Implemented entirely in Grasshopper using C# and Accord.NET, the system translates 3D meshes into high-dimensional feature vectors by voxelizing their geometry and measuring directional displacement from a structured grid. These vectors are then processed through an unsupervised SOM algorithm, organizing the morphogenetic outcomes into a coherent spatial grid. The output is a catalog of smooth form transitions — abstracted yet architecturally informed — that reveal latent spatial typologies and offer new generative potentials for designers.



Pseudocode :

```

* --- Preprocessing ---
DEFINE CenteredMeshes = NEW List<Mesh>
FOR EACH Mesh IN BuildingMeshes:
    CenterPoint = CalculateBoundingBoxCenter(Mesh)
    MoveVector = Vector(0,0,0) - CenterPoint
    CenteredMesh = TranslateMesh(Mesh, MoveVector)
    Add CenteredMesh to CenteredMeshes
END FOR

UnifiedBox = CalculateOverallBoundingBox(CenteredMeshes)
BaseGridPoints = GenerateGridOnXYPlane(UnifiedBox, GridX, GridY)

* --- Displacement Calculation --
DEFINE DisplacementVectorsPerMesh = NEW List<List<Vector3d>>
FOR EACH CenteredMesh IN CenteredMeshes:
    DEFINE CurrentMeshVectors = NEW List<Vector3d>
    FOR EACH Point IN BaseGridPoints:
        ClosestPointOnMesh = FindClosestPoint(Point, CenteredMesh)
        DisplacementVector = ClosestPointOnMesh - Point
        Add DisplacementVector to CurrentMeshVectors
    END FOR
    Add CurrentMeshVectors to DisplacementVectorsPerMesh
END FOR

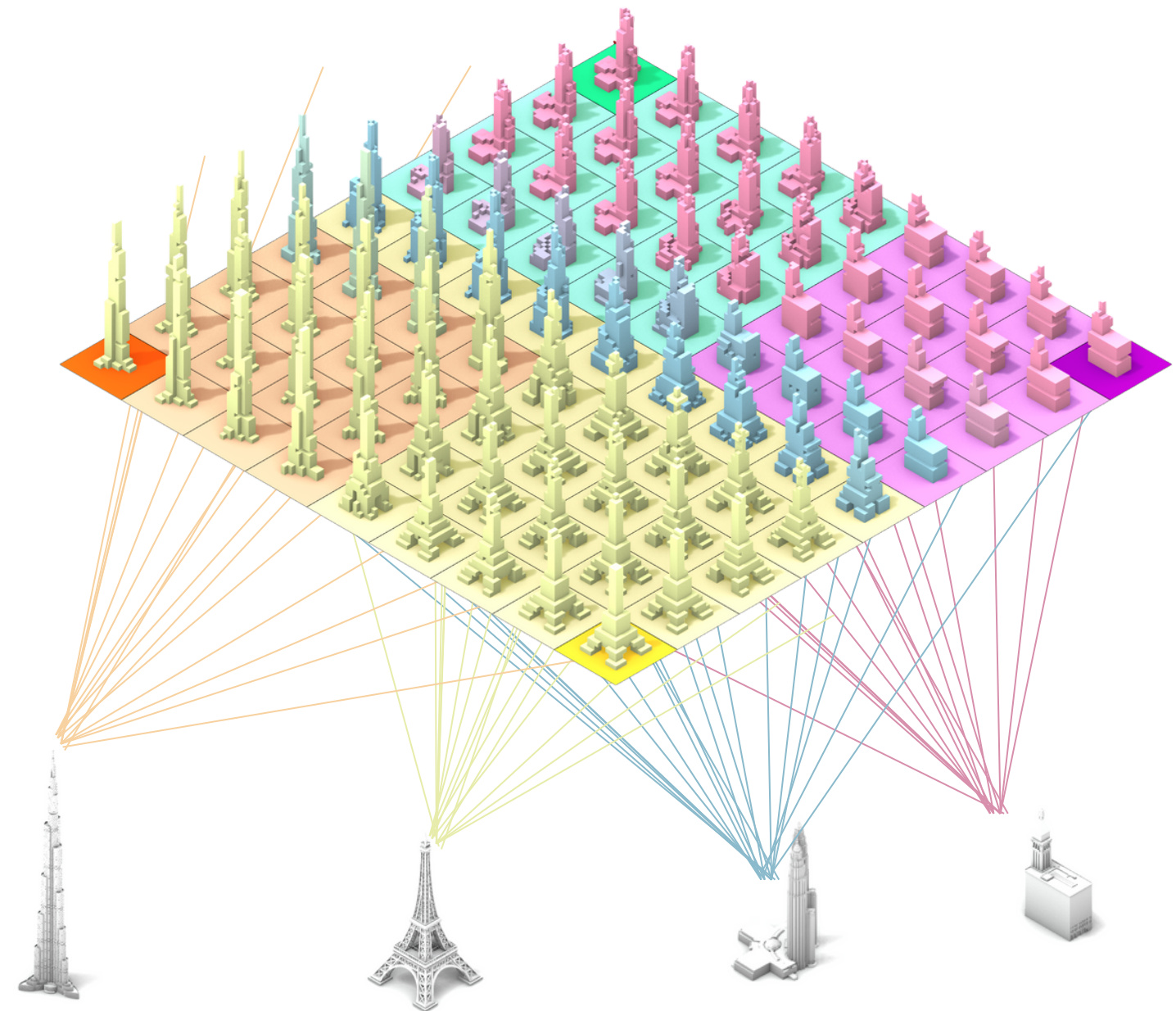
* --- SOM Morphing --
FormattedInputVectors = PrepareVectorsForSOM(DisplacementVectorsPerMesh)
SOM_Network = InitializeSOM(SOM_X, SOM_Y, VectorDimension)
TrainedSOM = TrainSOM(SOM_Network, FormattedInputVectors, SOM_Iter)
MorphedVectorsGrid = ExtractWeights(TrainedSOM) * Grid of vectors [SOM_X x SOM_Y]

* --- Mesh Reconstruction ---
DEFINE MorphedGeometry = NEW List<GeometryBase>
FOR i from 0 to SOM_X-1:
    FOR j from 0 to SOM_Y-1:
        BasePoint = GetCorrespondingBaseGridPoint(i, j, BaseGridPoints, GridX, GridY) * Map SOM index to Base Grid index
        MorphedVector = MorphedVectorsGrid[i][j]
        TargetPoint = BasePoint + MorphedVector
        Voxel = CreateVoxelAtPoint(TargetPoint) * Or other geometry primitive
        Add Voxel to MorphedGeometry
    END FOR
END FOR

MorphedMeshes = CombineGeometry(MorphedGeometry) * E.g., Weaverbird MeshFromVoxels

RETURN MorphedMeshes
END

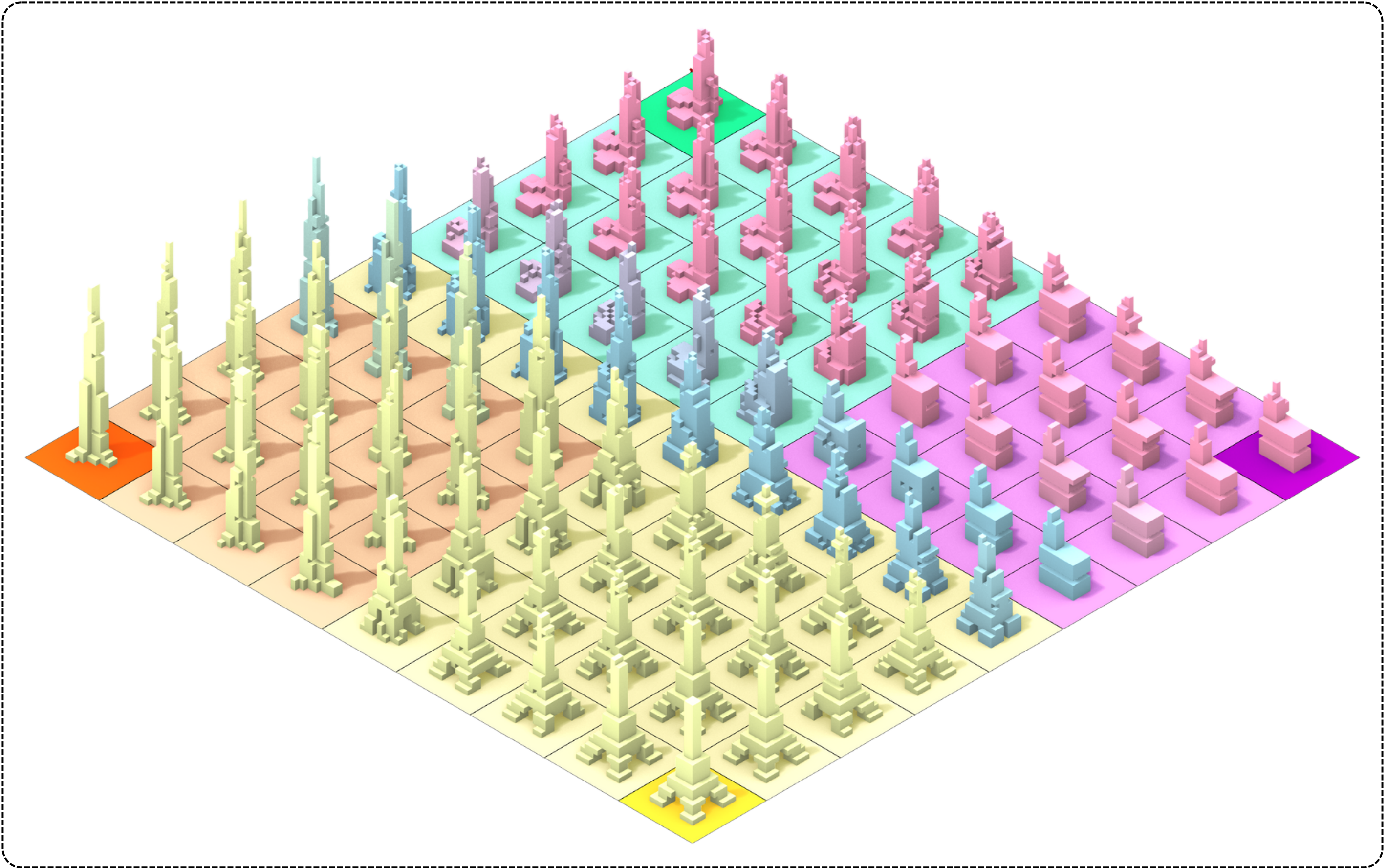
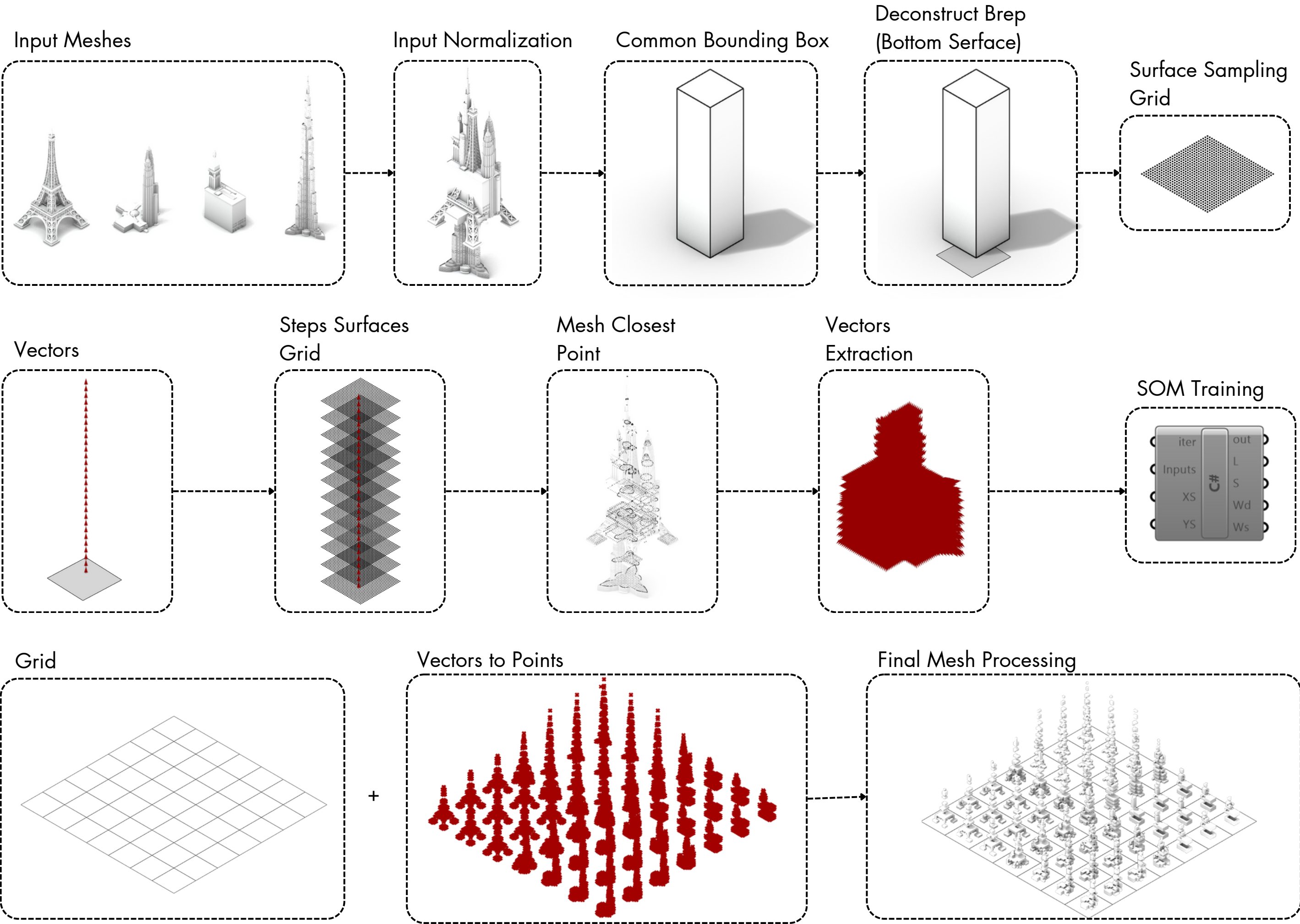
```



References:

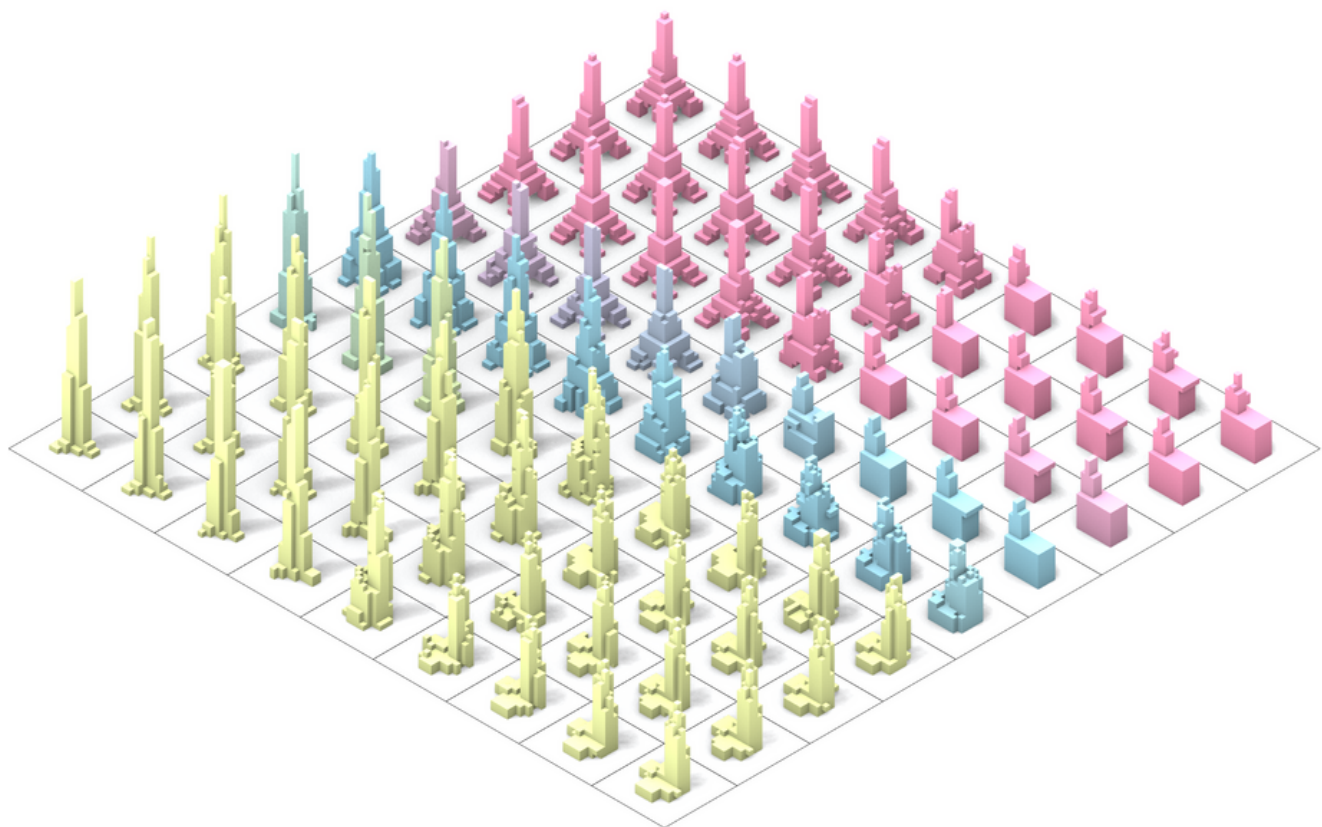
<http://accord-framework.net/>
<https://www.giulopiacentino.com/weaverbird/>
<https://www.food4rhino.com/en/app/lunchbox>
<https://arxiv.org/abs/2103.01458>

Methods Board



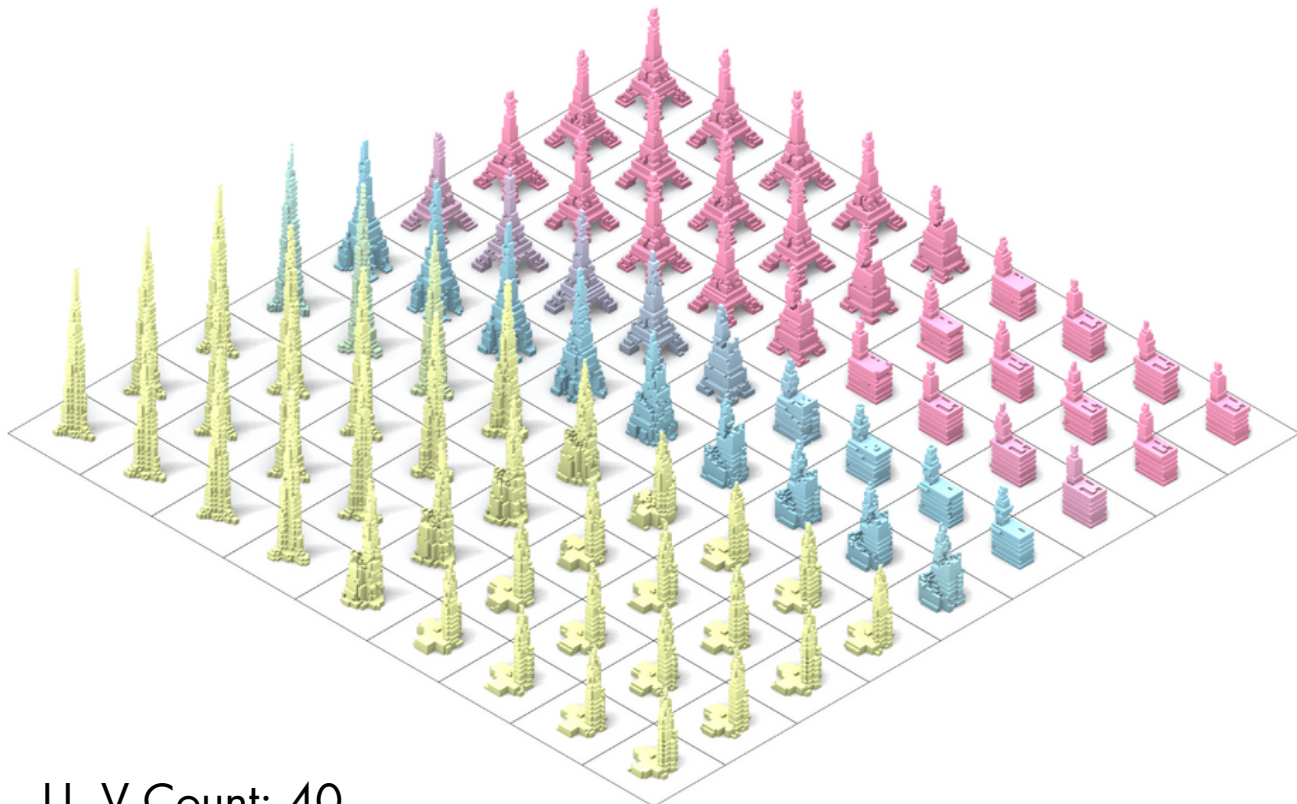
Results Board

Result 1



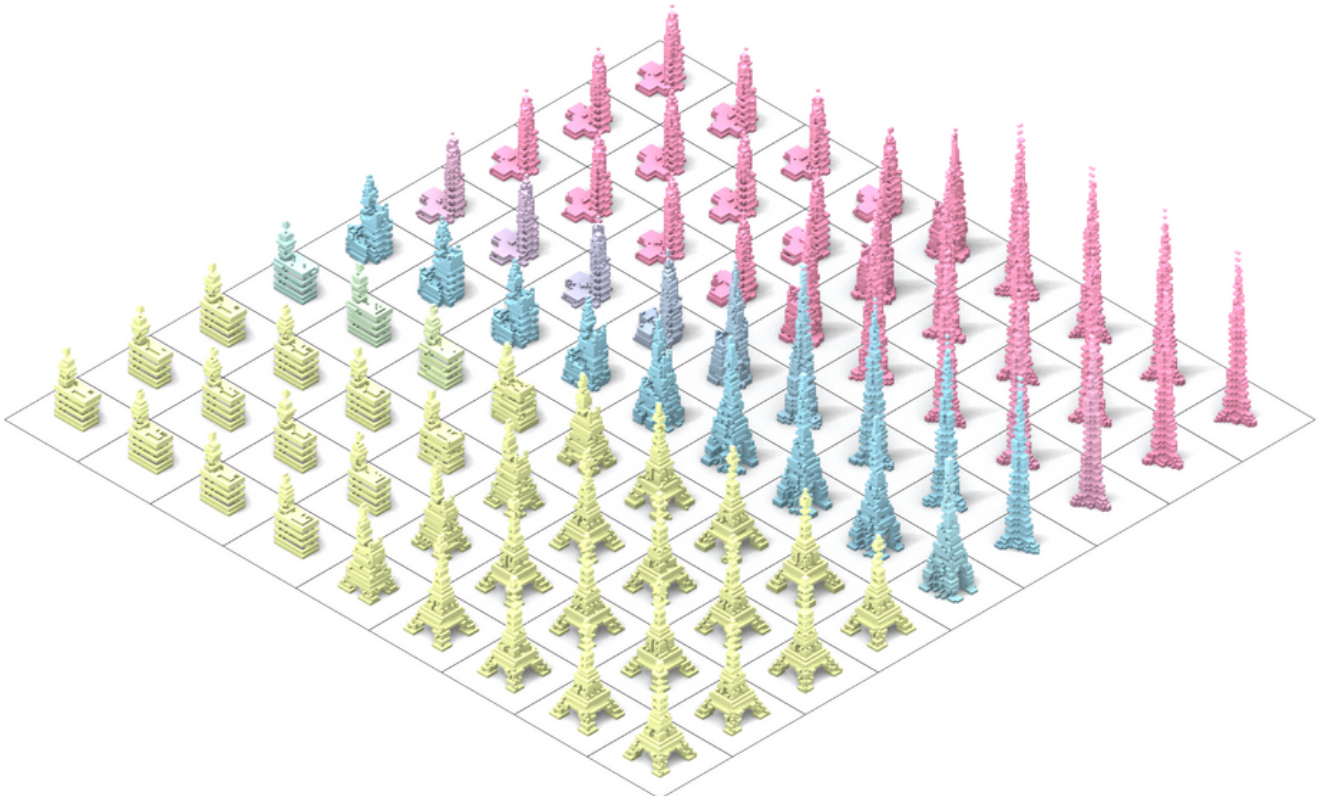
U, V Count: 40
Steps: 40
Compute Time: 432.9 Seconds
Voxel Size (x,y,z): 31

Result 1



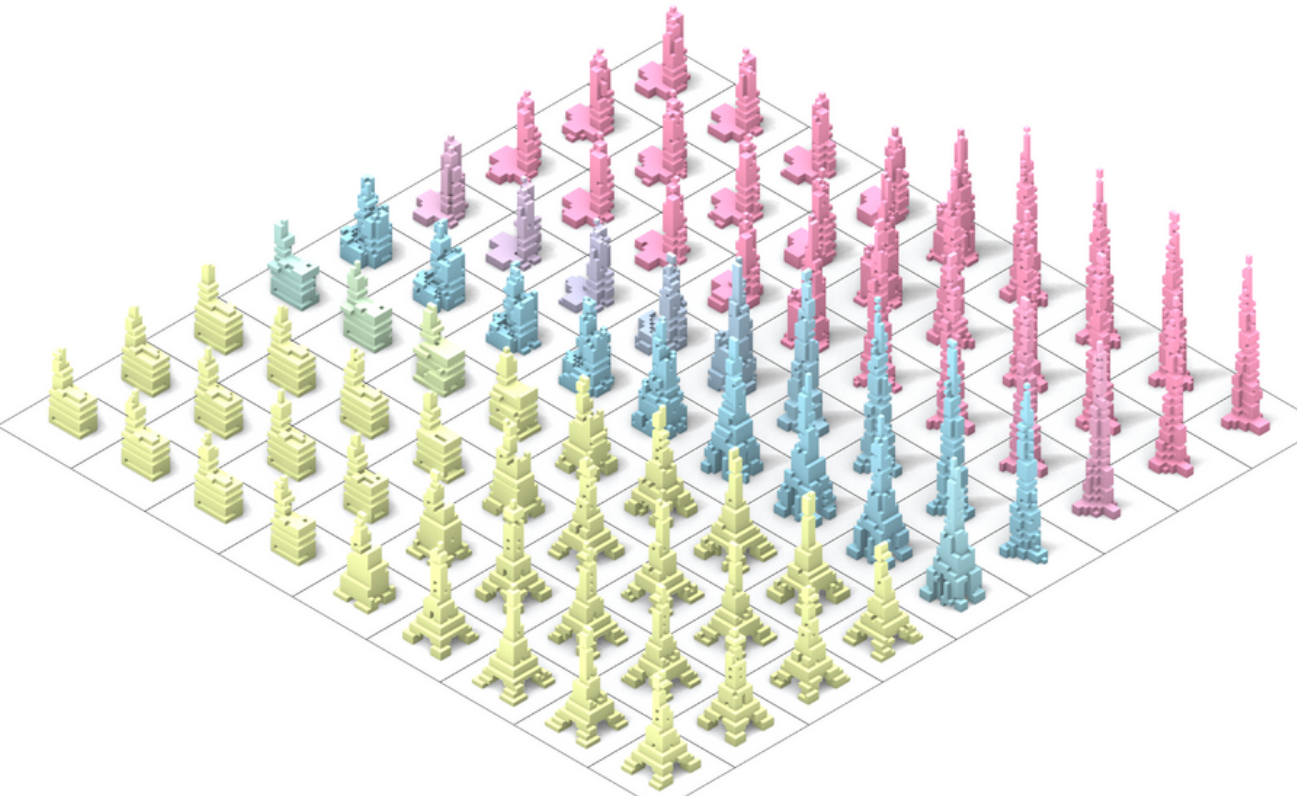
U, V Count: 40
Steps: 40
Compute Time: 75.2 Seconds
Voxel Size (x,y,z): 15

Result 1



U, V Count: 25
Steps: 25
Compute Time: 111.4 Seconds
Voxel Size (x,y,z): 15

Result 1



U, V Count: 25
Steps: 25
Compute Time: 4.8 Seconds
Voxel Size (x,y,z): 25

Top View/ 4 Inputs



Top View/ 3 Inputs

