

ARCH 686 | VISUAL STUDIO (C++)

AARON CHEN, AFRA KHADEMIA, BOXU ZENG, KAREN AJALA, NAIXIN CHENG

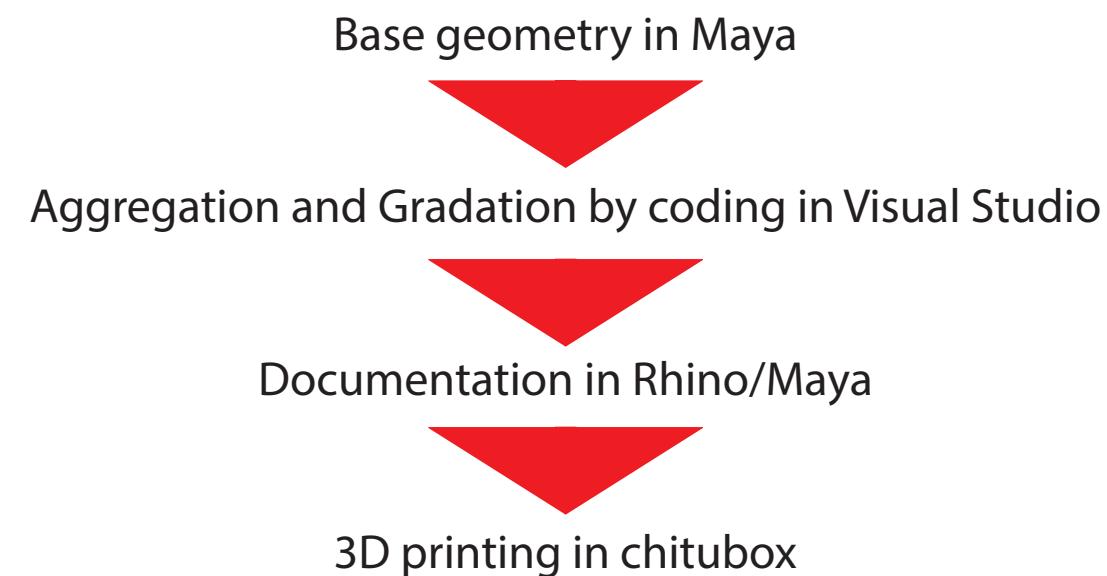
ARCH 686 | VISUAL STUDIO (C++)

OBJECTIVES

- Create a FGM file using Maya-code(c++) and 3dp (chitobox) as a road map
- Create a catalogue of various components and a study of their aggregation in Maya
- Create a catalogue of related coded aggregations (c++)
- Create a database of 3D printing statistics (estimated print time, estimated material consumption, estimated supports etc)

FGM: Functionally Graded Material refers to the gradation used in computational design and process to efficiently utilize structure, maximize materials and avoid unnecessary waste.

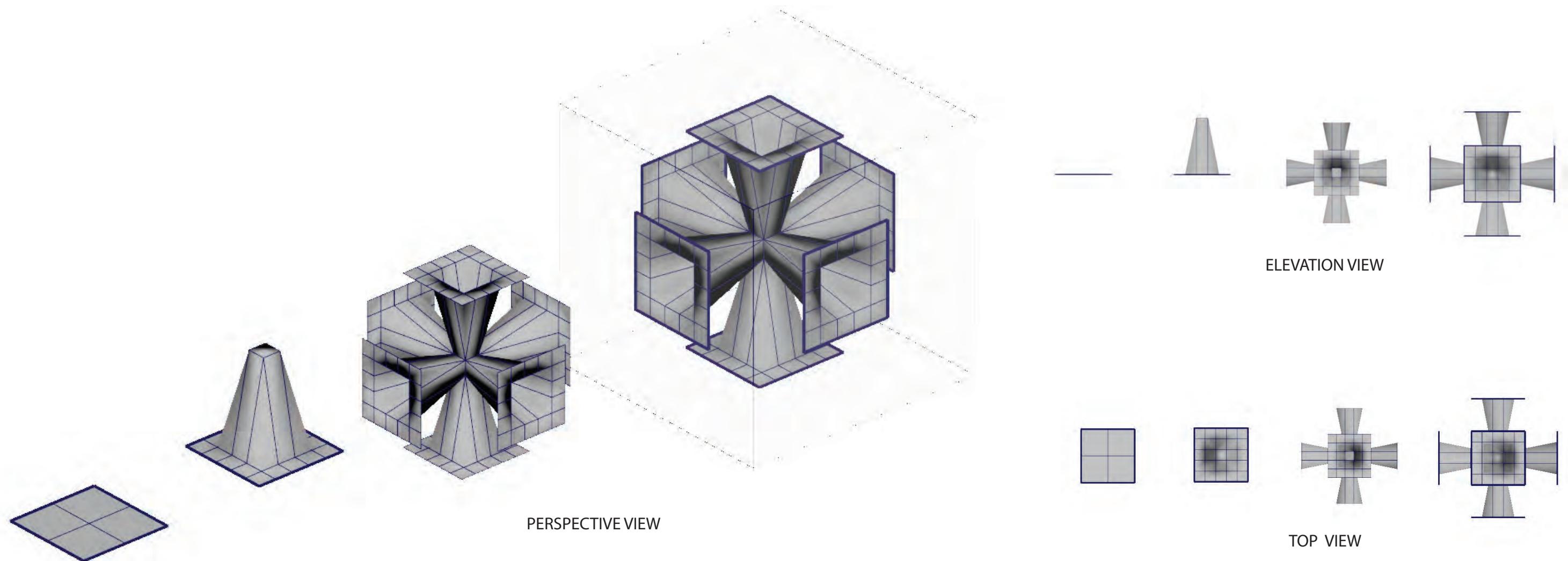
ARCH 686 | VISUAL STUDIO (C++)
UNDERSTANDING THE WORKFLOW



ARCH 686 | VISUAL STUDIO (C++)

EXPLORATIONS AND ITERATIONS CATALOGUE

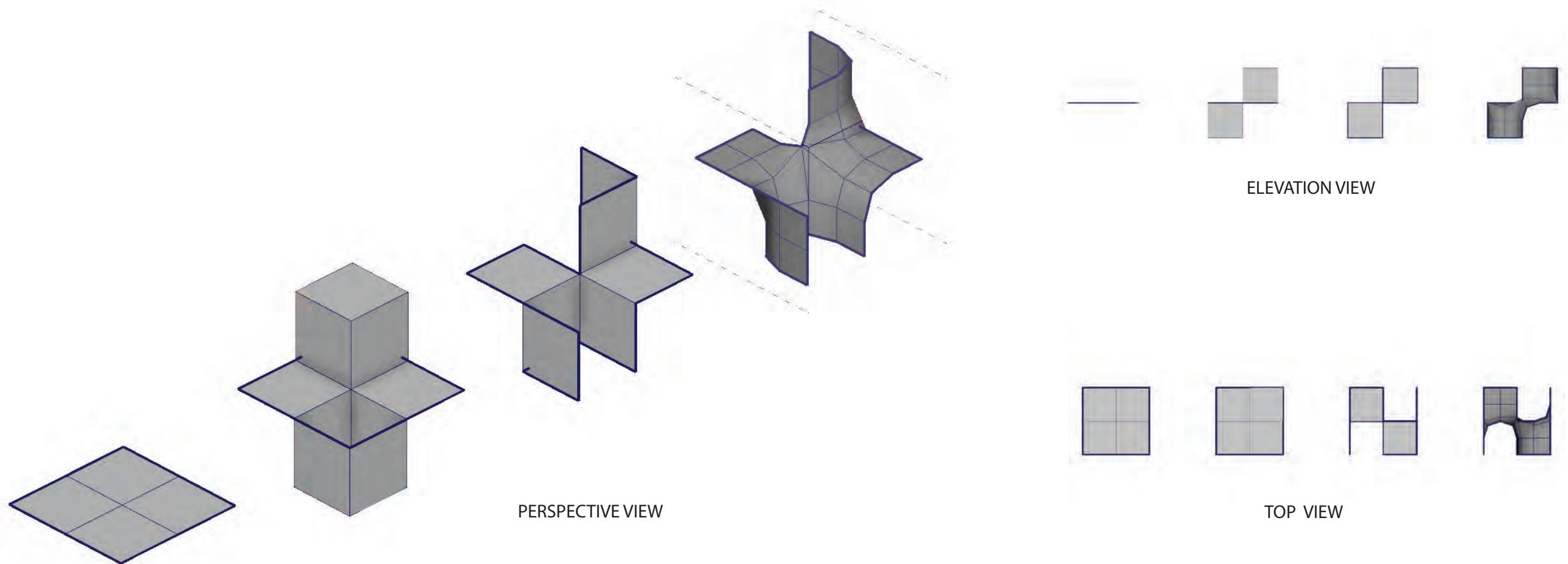
Iteration (1)



ARCH 686 | VISUAL STUDIO (C++)

EXPLORATIONS AND ITERATIONS CATALOGUE

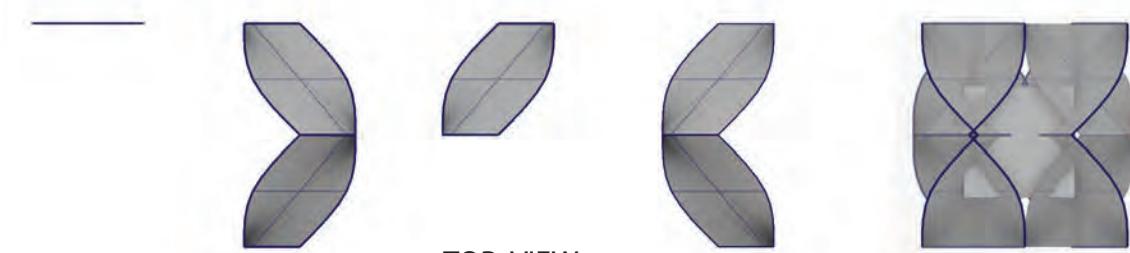
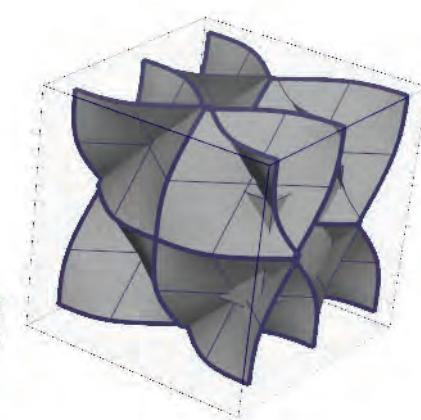
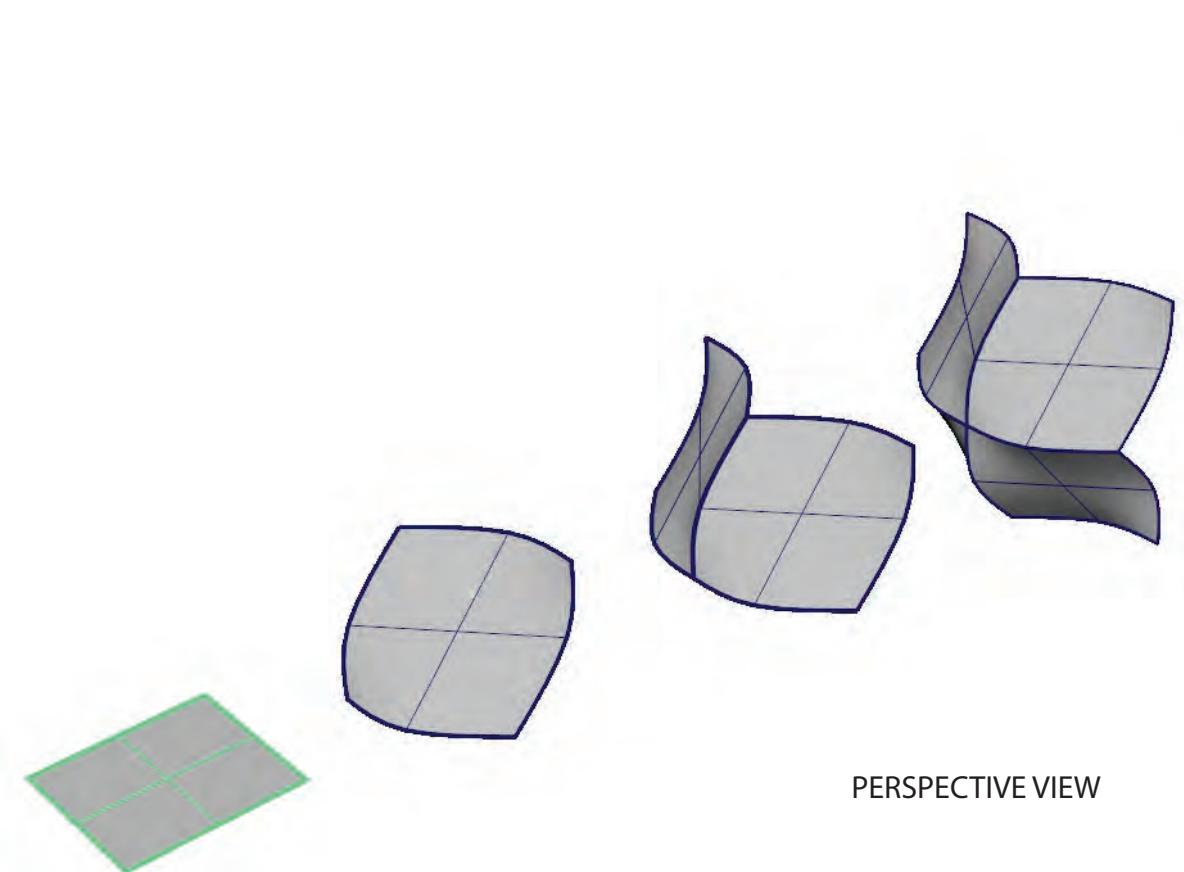
Iteration (2)



ARCH 686 | VISUAL STUDIO (C++)

EXPLORATIONS AND ITERATIONS CATALOGUE

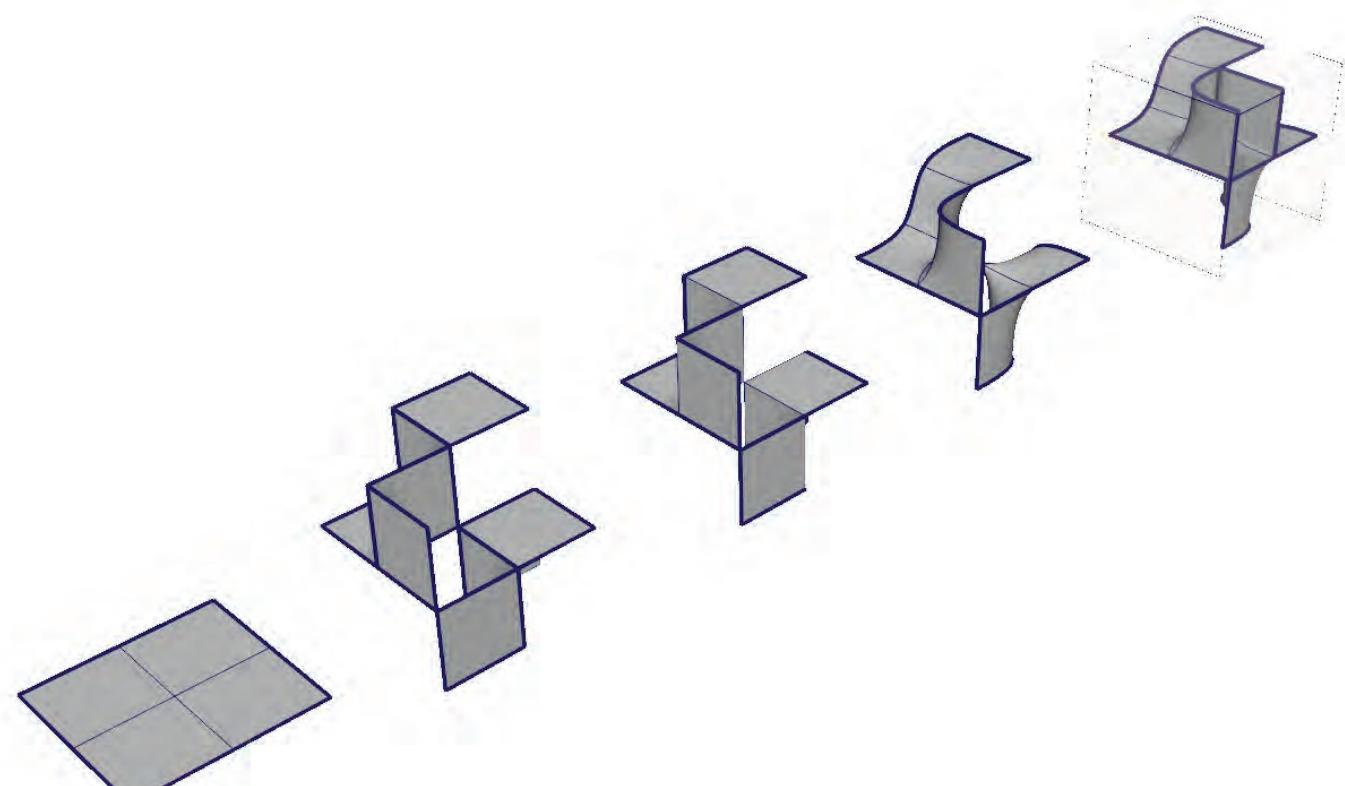
Iteration (3)



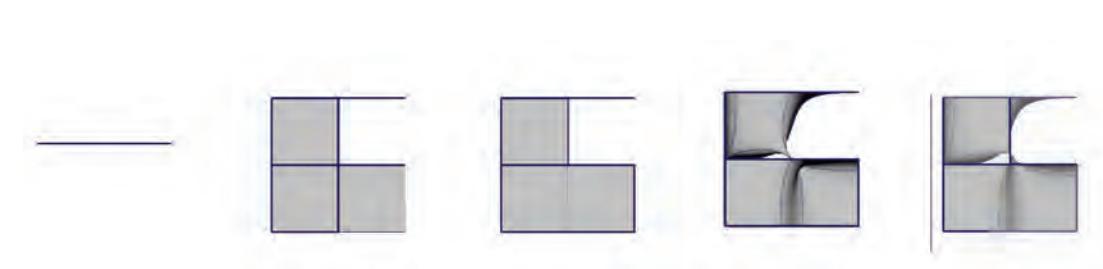
ARCH 686 | VISUAL STUDIO (C++)

EXPLORATIONS AND ITERATIONS CATALOGUE

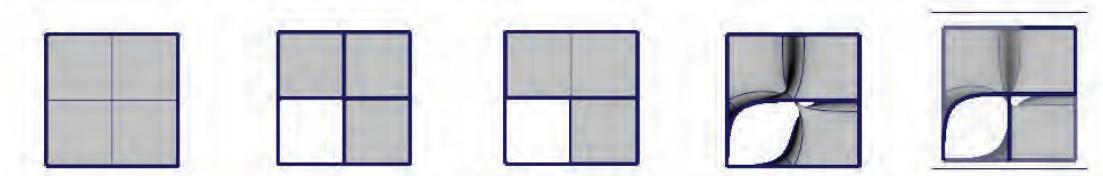
Iteration (4)



PERSPECTIVE VIEW



ELEVATION VIEW

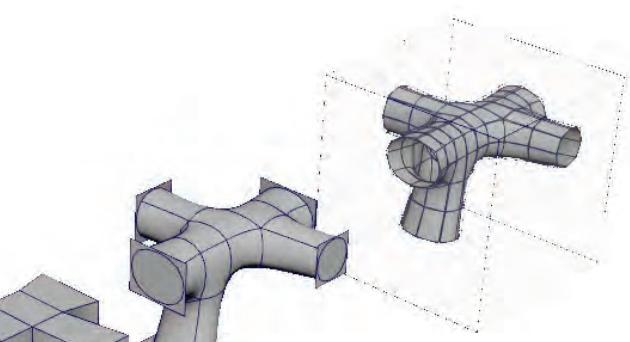
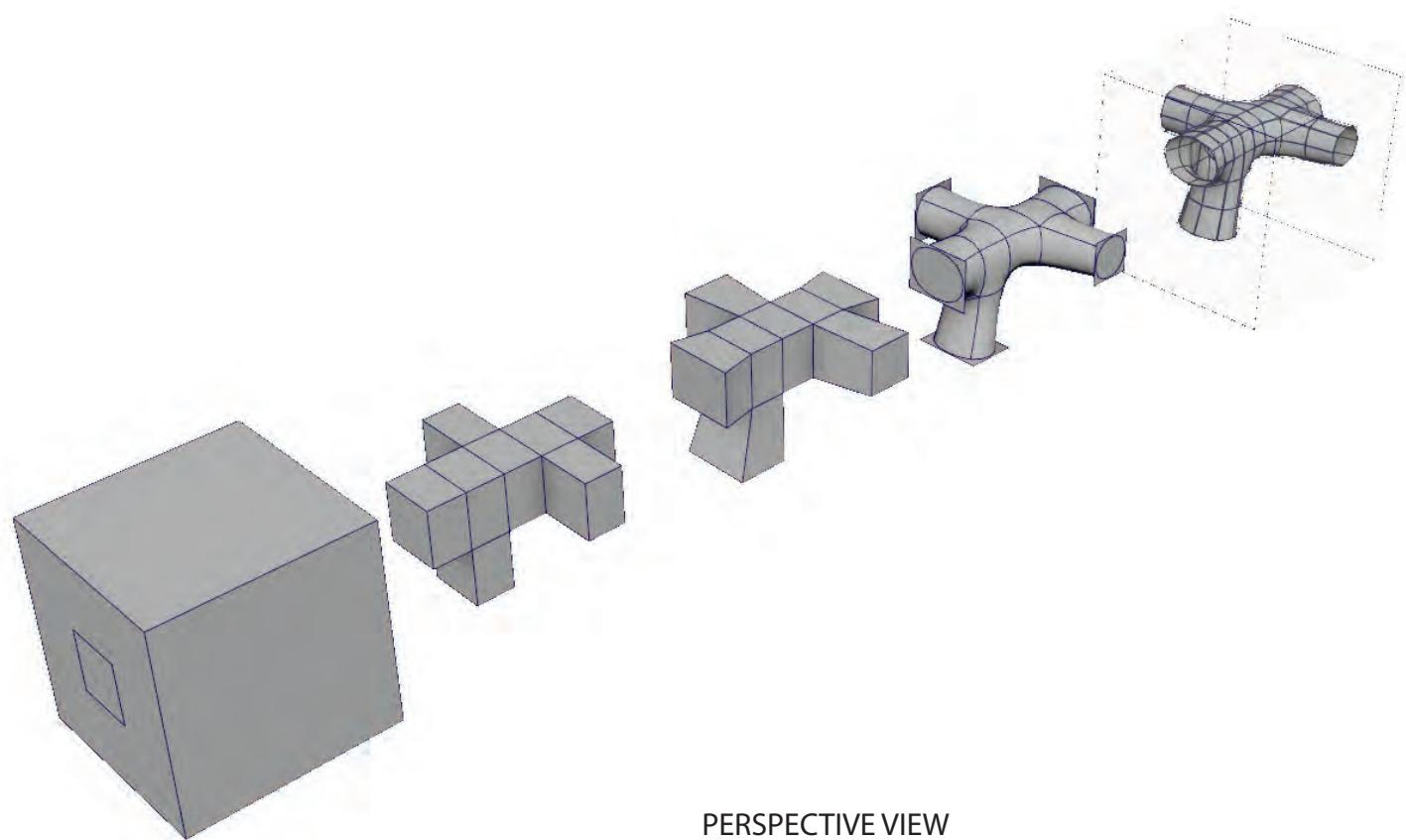


TOP VIEW

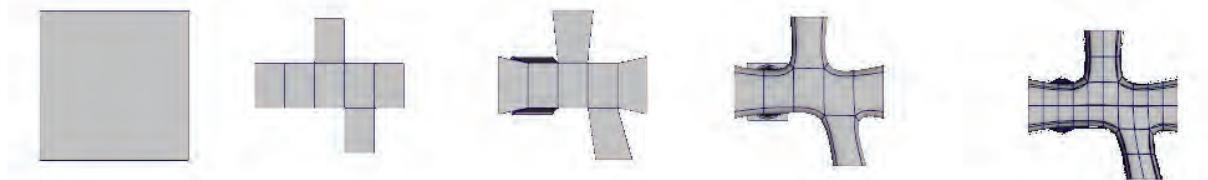
ARCH 686 | VISUAL STUDIO (C++)

EXPLORATIONS AND ITERATIONS CATALOGUE

Iteration (5)



ELEVATION VIEW

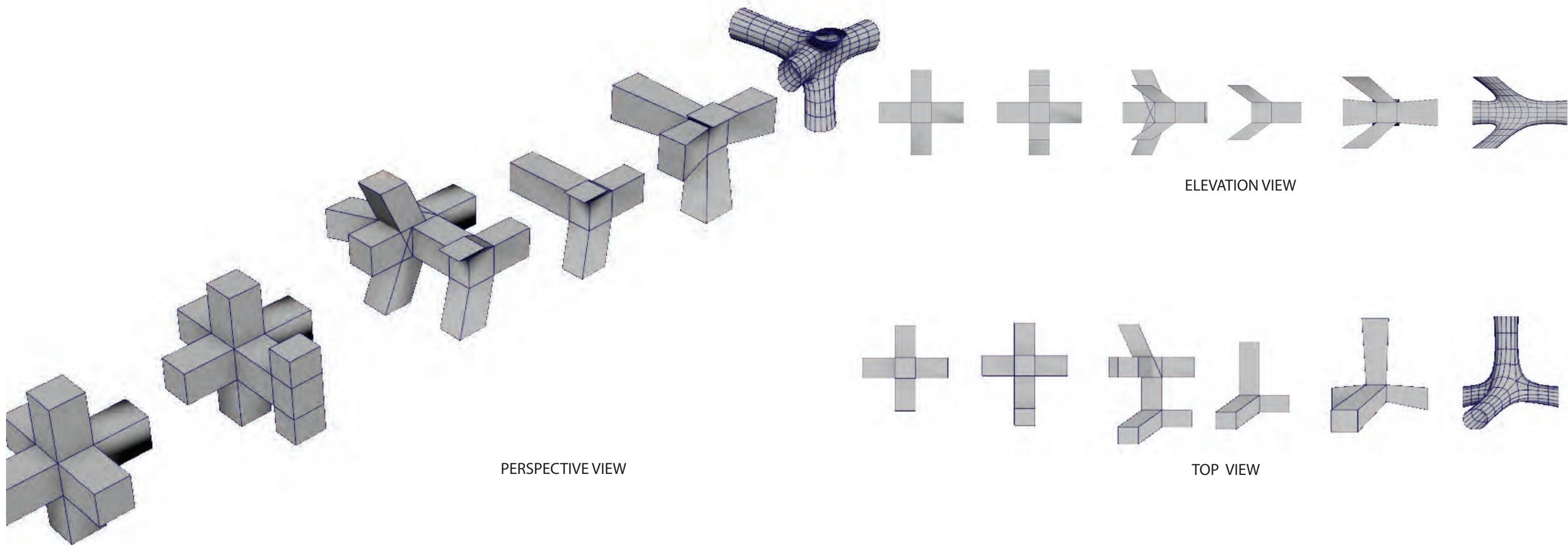


TOP VIEW

ARCH 686 | VISUAL STUDIO (C++)

EXPLORATIONS AND ITERATIONS CATALOGUE

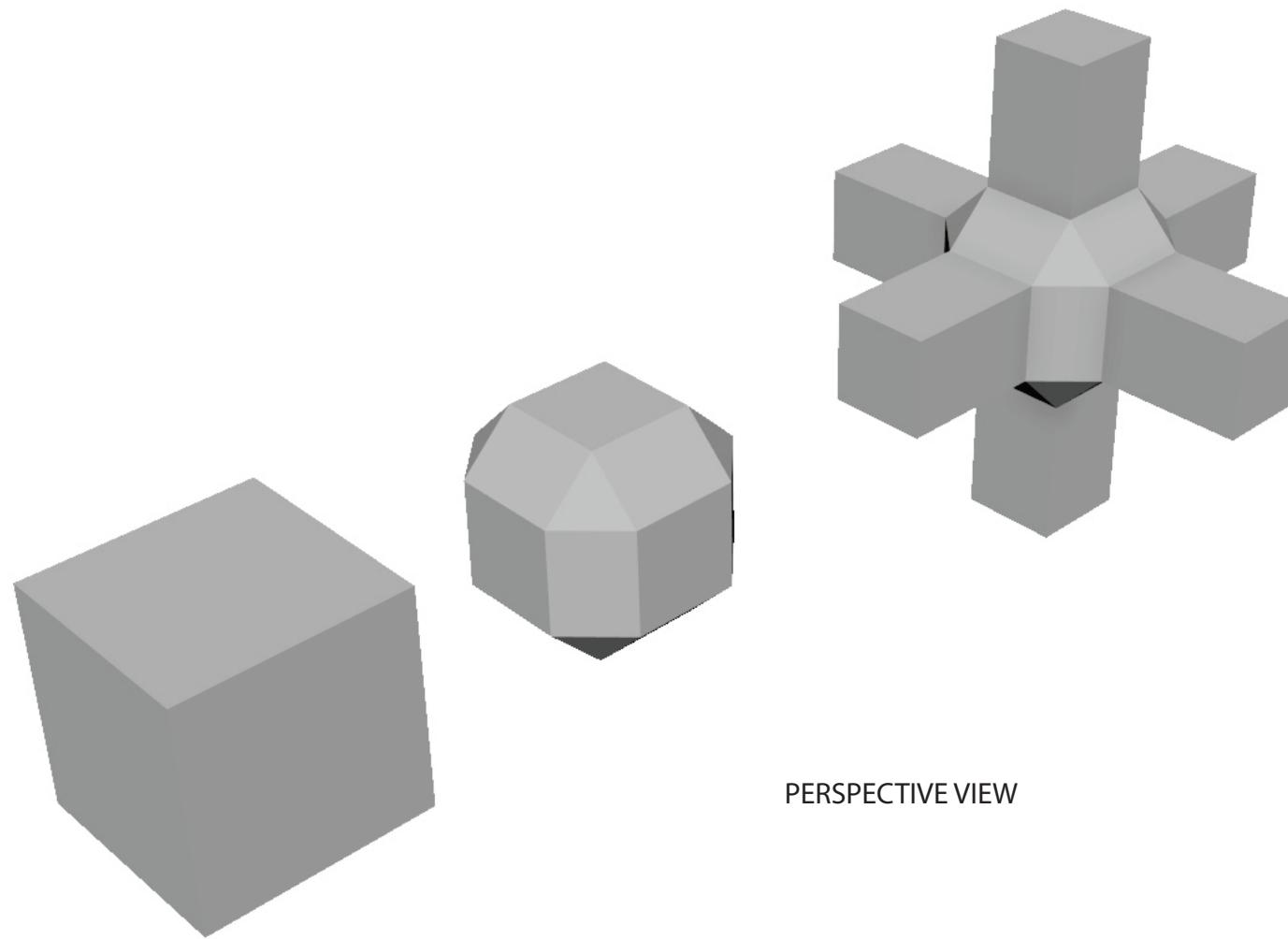
Iteration (6)



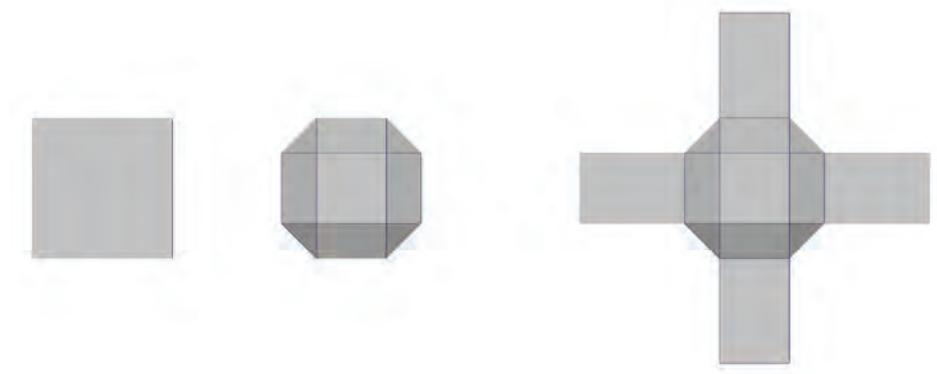
ARCH 686 | VISUAL STUDIO (C++)

EXPLORATIONS AND ITERATIONS CATALOGUE

Iteration (7)

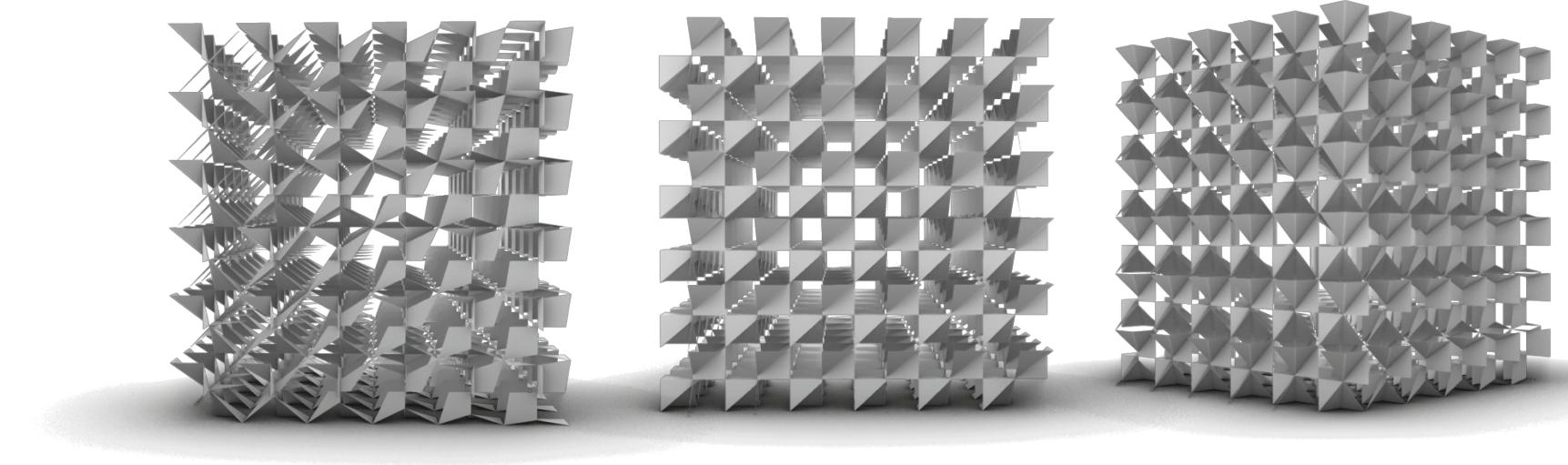
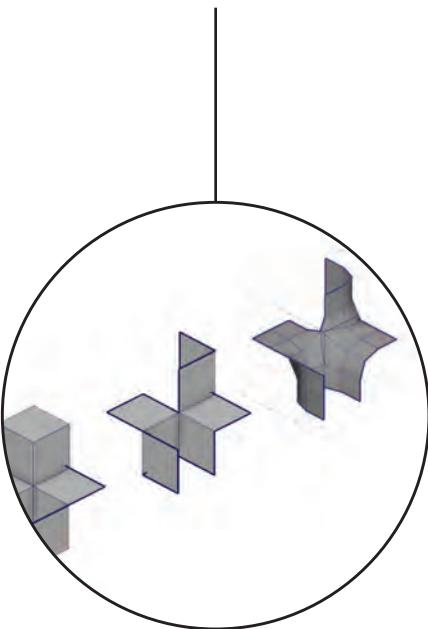
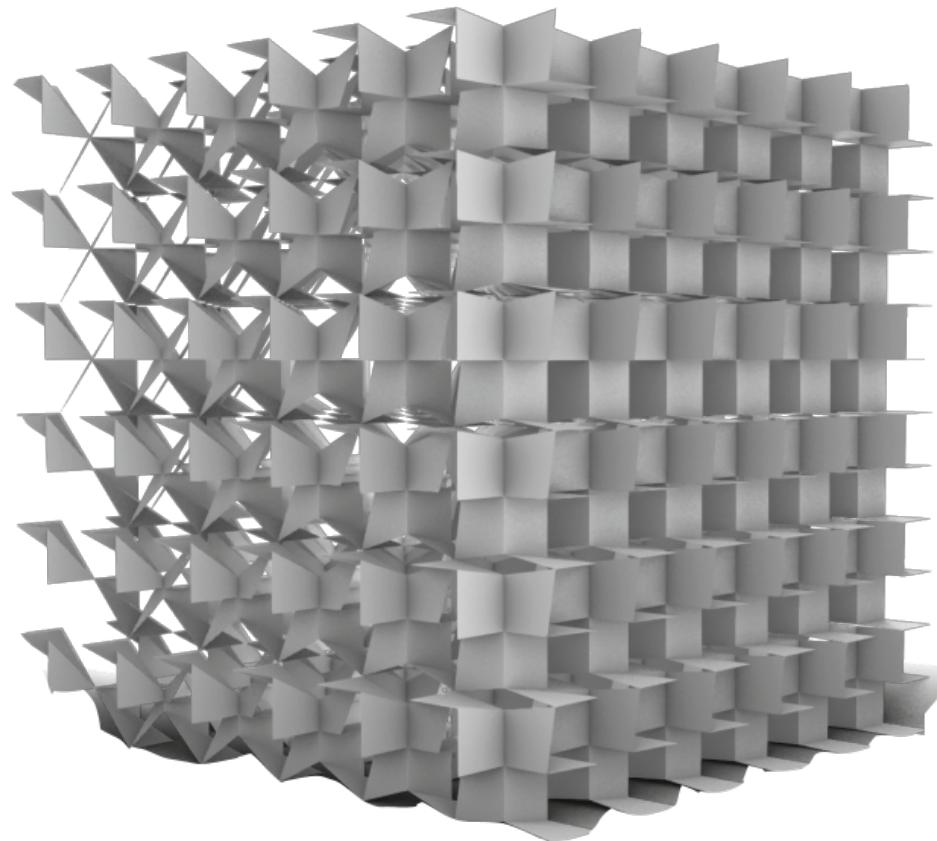


PERSPECTIVE VIEW



TOP & ELEVATION VIEW

ARCH 686 | VISUAL STUDIO (C++)
CODING / AGGREGATION CATALOGUE
ITERATION (2)



```
// create crss3d mesh
//createMeshFromFile("data/cross3d.obj");
createMeshFromFile("data/boxu_3.obj");
transformMesh(mat);

//----- trasnfromPartofMesh

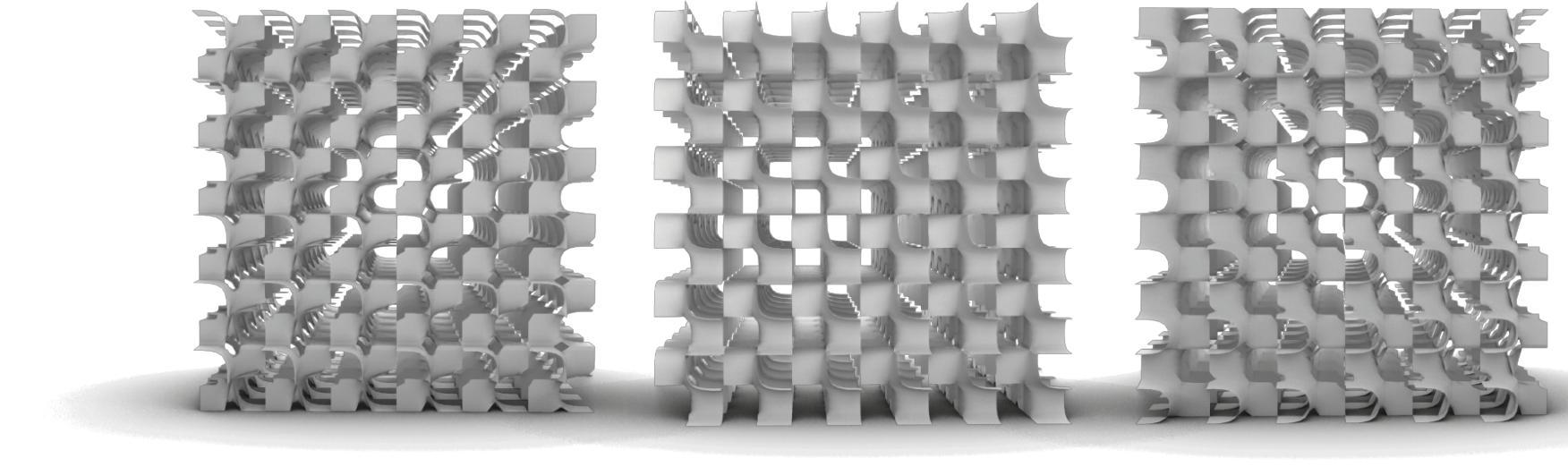
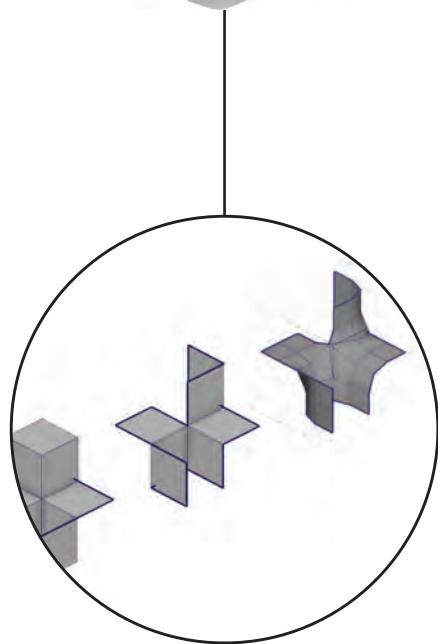
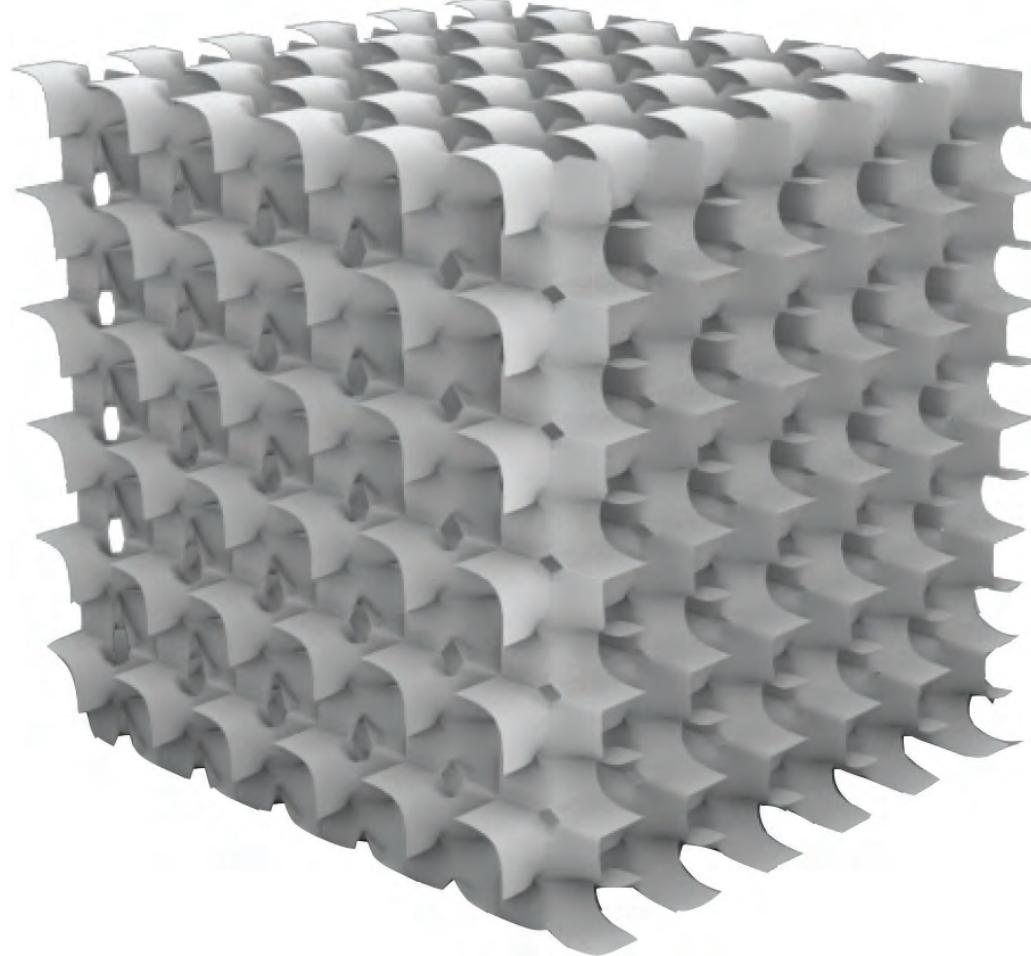
//construct array with vertex ids of the vertices that need to be trasnfromed
//int src[] = { 0, 1, 2, 3, 4, 5, 6, 7 };// cross 3d
int src[] = { 1, 3, 5, 7 };
int n = sizeof(src) / sizeof(src[0]);
zIntArray ids(src, src + n);
|
// construct a gradient parameter
zVector u, v, w, c;
//attractor = zVector(3, 3, 3);

c = cen;
float maxL = diagLength;

//float disAttractor = cen.distanceTo(attractor);

float mult = ofMap(id, 0, 215, 0.01, 1); // 4 - ofMap(cen.length(), 0, maxL, 0.1, 4);
```

ARCH 686 | VISUAL STUDIO (C++)
CODING / AGGREGATION CATALOGUE
ITERATION (2)



```
// create crss3d mesh
//createMeshFromFile("data/cross3d.obj");
createMeshFromFile("data/boxu_3.obj");
transformMesh(mat);

//----- trasnfromPArtOfMesh

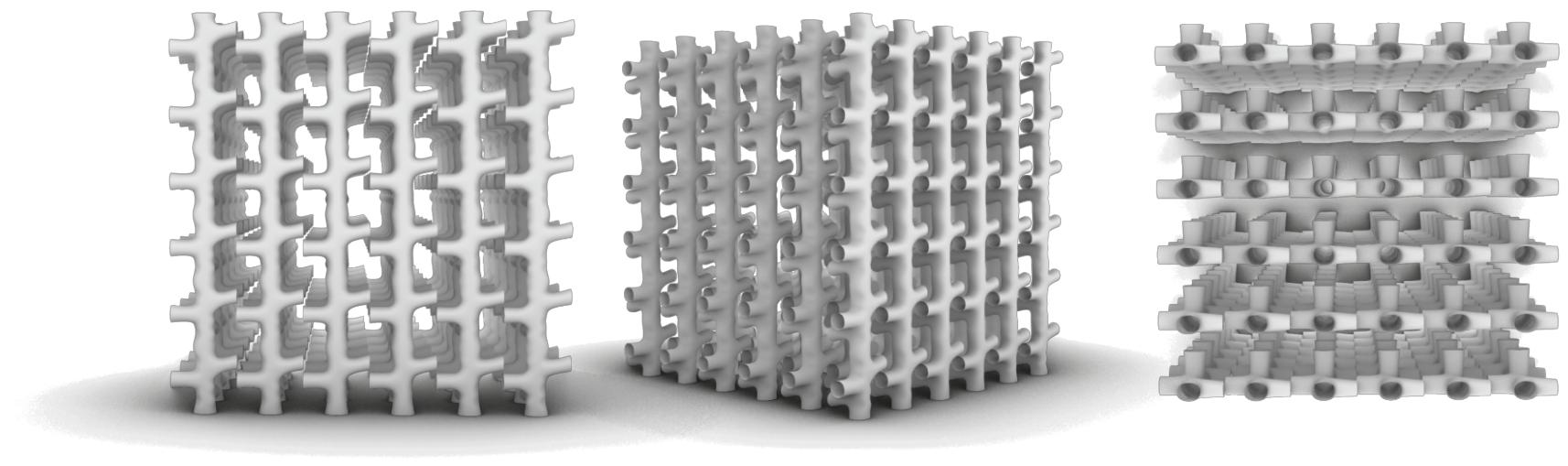
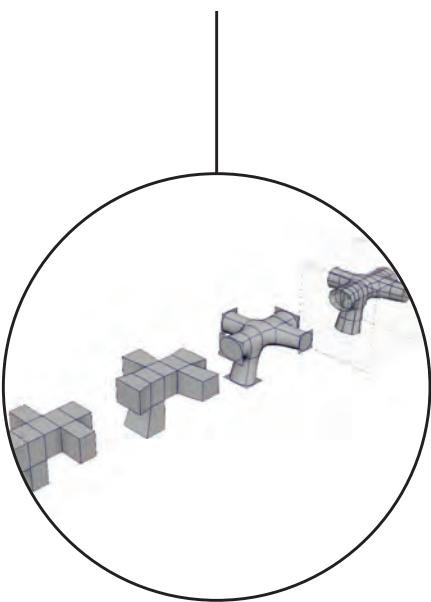
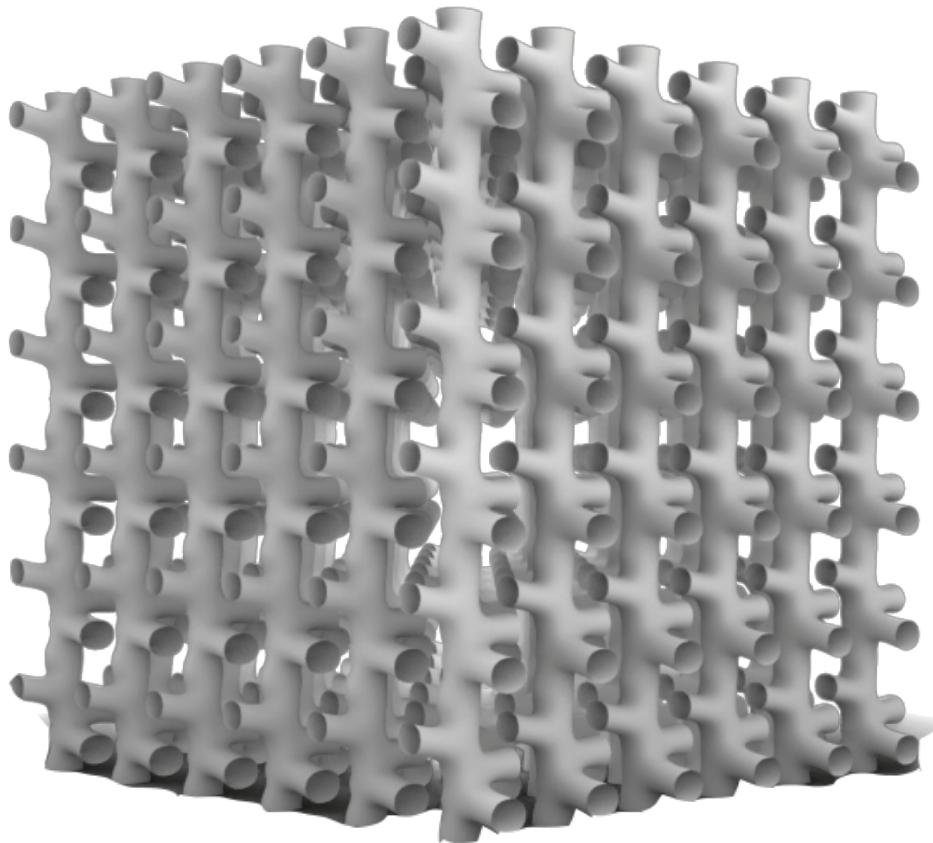
//construct array with vertex ids of the vertices that need to be trasnfomed
//int src[] = { 0, 1, 2, 3, 4, 5, 6, 7 };// cross 3d
int src[] = { 1, 3, 5, 7 };
int n = sizeof(src) / sizeof(src[0]);
zIntArray ids(src, src + n);
|
// construct a gradient parameter
zVector u, v, w, c;
//attractor = zVector(3, 3, 3);

c = cen;
float maxL = diagLength;

//float disAttractor = cen.distanceTo(attractor);

float mult = ofMap(id, 0, 215, 0.01, 1); // 4 - ofMap(cen.length(), 0, maxL, 0.1, 4);
```

ARCH 686 | VISUAL STUDIO (C++)
 CODING / AGGREGATION CATALOGUE
 ITERATION (5)



```

//construct the transformation matrix
zTransform TM;
TM.setIdentity();
u = zVector(1, 0, 0);
v = zVector(0, 1, 0);
w = zVector(0, 0, 1);
u.normalize(); v.normalize(); w.normalize();
//u *= mult; v *= mult; w *= mult;

float tx, ty, tz;
tx = 0;
ty = 0;
tz = mult;

//assign the values to the matrix
TM.col(0) << u.x, u.y, u.z, 1;
TM.col(1) << v.x, v.y, v.z, 1;
TM.col(2) << w.x, w.y, w.z, 1;
TM.col(3) << tx, ty, tz, 1;

// use transform matrix and ids, to transform part of the mesh;
transformPartOfMesh(ids, TM);

// create crss3d mesh
//createMeshFromFile("data/cross3d.obj");
createMeshFromFile("data/Karen_Ite2.obj");
transformMesh(mat);

//----- trasnfromPartOfMesh

//construct array with vertex ids of the vertices that need to be trasnfomed
//int src[] = { 0, 1, 2, 3, 4, 5, 6, 7 };// cross 3d
int src[] = { 4, 5, 6, 7, 8, 9, 10, 11 };
int n = sizeof(src) / sizeof(src[0]);
zIntArray ids(src, src + n);

// construct a gradient parameter
zVector u, v, w, c, move, origin, attractor;
attractor = zVector(3, 3, 3);

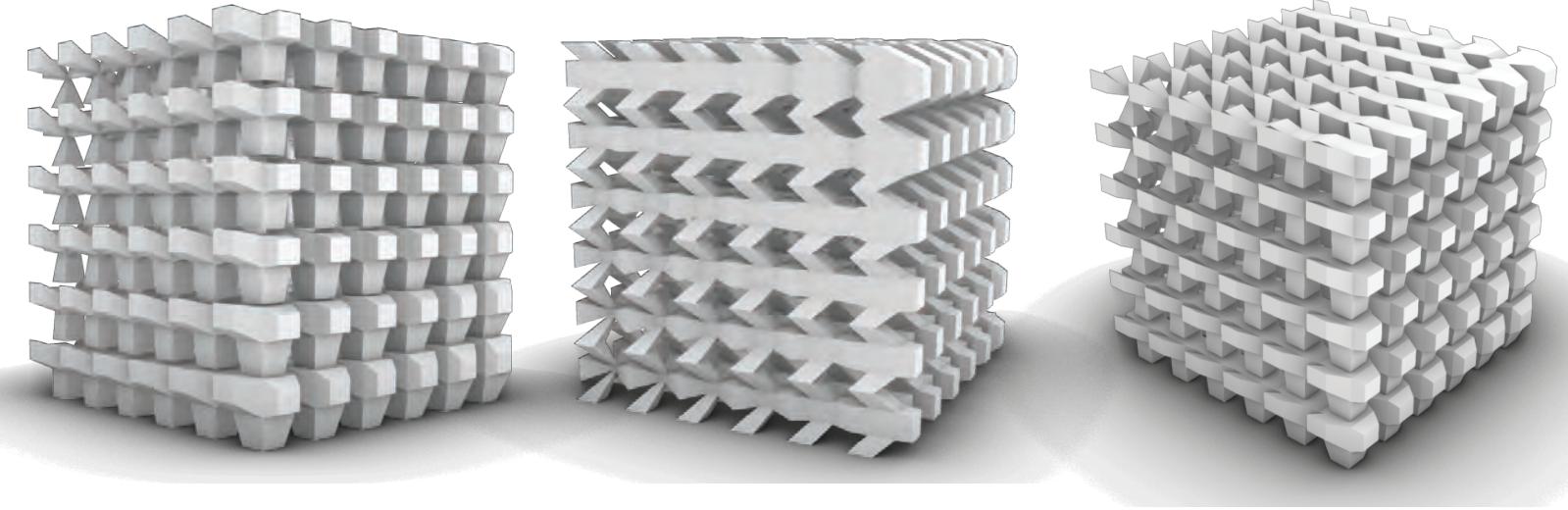
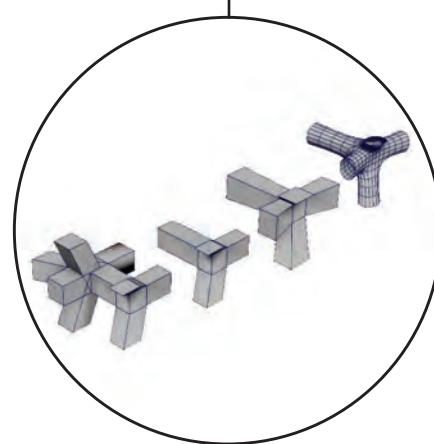
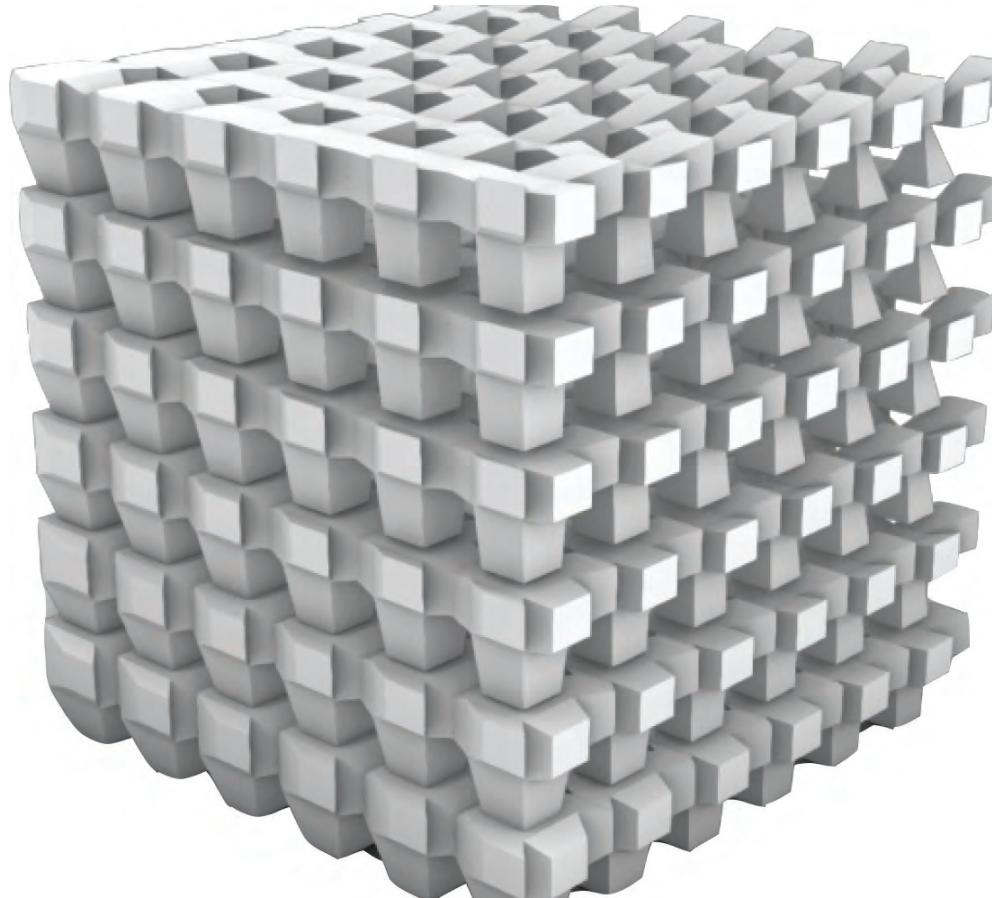
c = cen;
float maxL = diagLength;

float disAttractor = cen.distanceTo(attractor);

float mult = ofMap(cen.z, -3, 3, -0.1, 1); // 4 - ofMap(cen.length(), 0, maxL, 0.1, 4);

```

ARCH 686 | VISUAL STUDIO (C++)
CODING / AGGREGATION CATALOGUE
ITERATION (6)



```
// create crss3d mesh
//createMeshFromFile("data/cross3d.obj");
createMeshFromFile("data/Karen_Ite.obj");
transformMesh(mat);

//----- trasnfromPArtofMesh

//construct array with vertex ids of the vertices that need to be trasnfromed
//int src[] = { 0,1,2,3,4,5,6,7 };// cross 3d
int src[] = { 4,5,6,7,8,9,10,11 };
int n = sizeof(src) / sizeof(src[0]);
zIntArray ids(src, src + n);

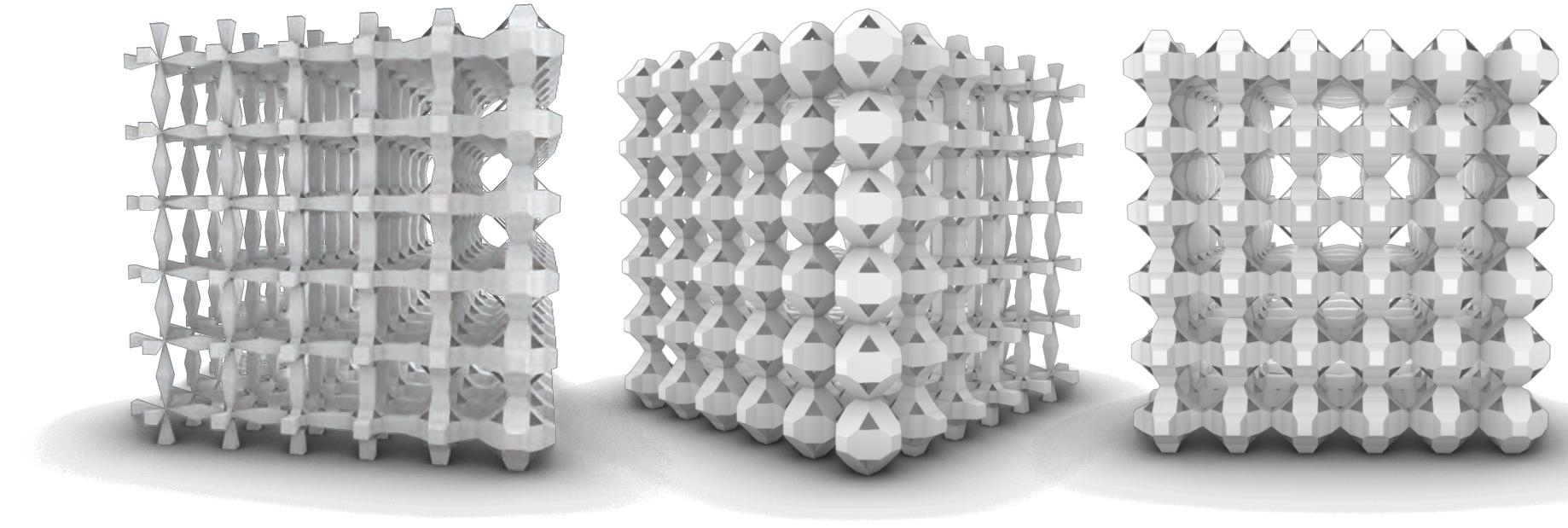
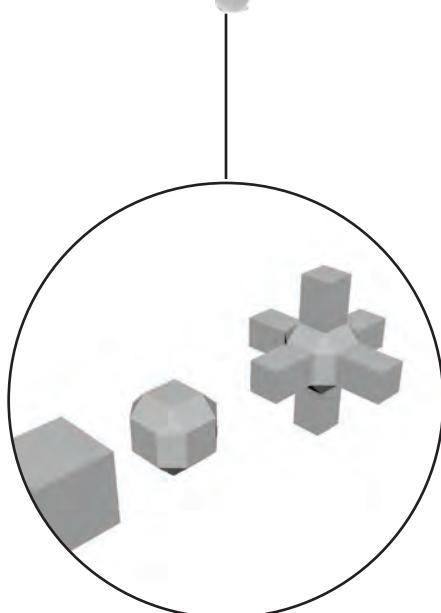
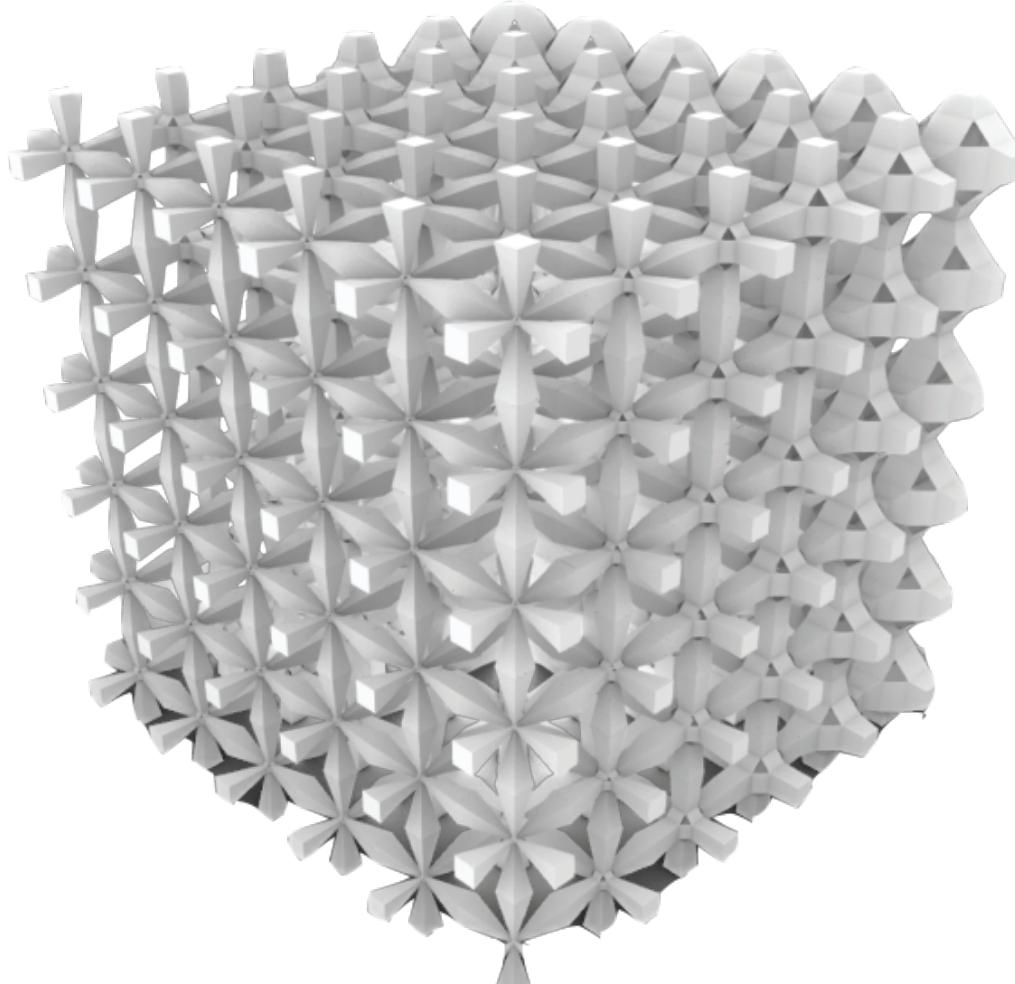
// construct a gradient parameter
zVector u, v, w, c, move, origin, attractor;
attractor = zVector(3,3,3);

c = cen;
float maxL = diagLength;

float disAttractor = cen.distanceTo(attractor);

float mult = ofMap(disAttractor, 0, 8, 0.01, 2); // 4 - ofMap(cen.length(), 0, maxL, 0.1, 4);
```

ARCH 686 | VISUAL STUDIO (C++)
CODING / AGGREGATION CATALOGUE
ITERATION (7)



```
// create crss3d mesh
//createMeshFromFile("data/cross3d.obj");
createMeshFromFile("data/Cindy_Ite.obj");
transformMesh(mat);

//----- trasnfromPArtofMesh

//construct array with vertex ids of the vertices that need to be trasnfromed
//int src[] = { 0,1,2,3,4,5,6,7 };// cross 3d
int src[] = { 8,9,13,16,6,7,10,11,17,15,12,14,0,20,21,5,1,3,4,23,22,18,19,2 };
int n = sizeof(src) / sizeof(src[0]);
zIntArray ids(src, src + n);

// construct a gradient parameter
zVector u, v, w, c;
//attractor = zVector(3,3,3);

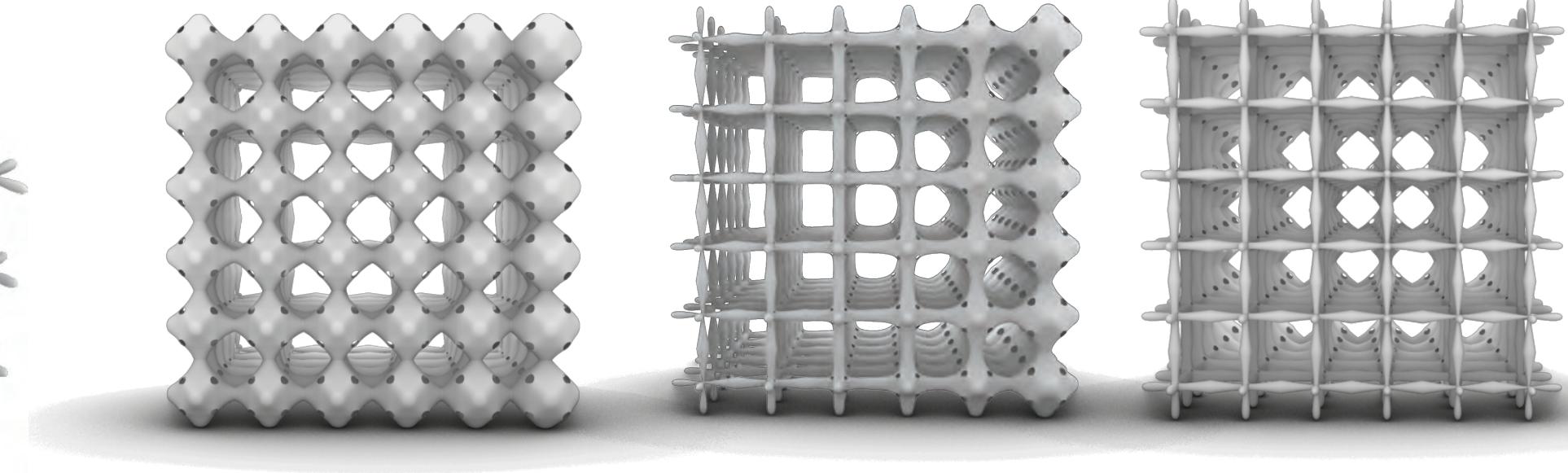
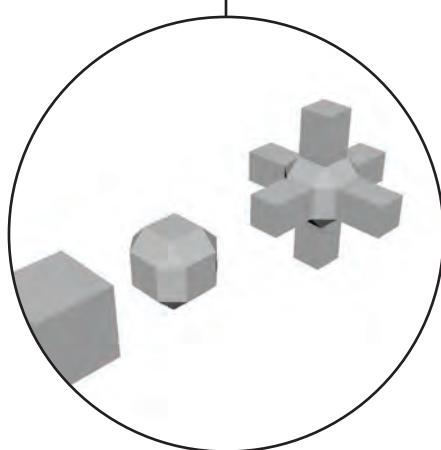
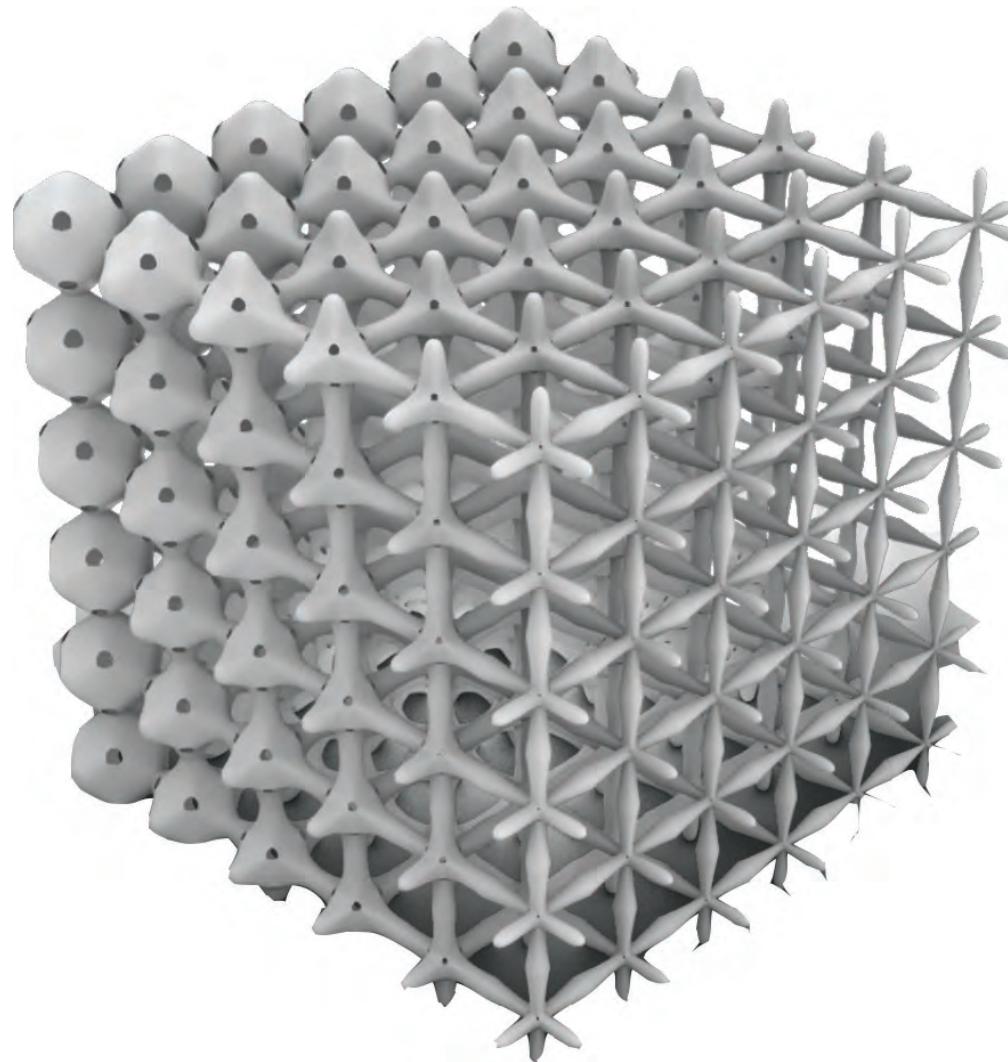
c = cen;
float maxL = diagLength;

//float disAttractor = cen.distanceTo(attractor);

float mult = ofMap(id, 0, 215, 0.1, 2); // 4 - ofMap(cen.length(), 0, maxL, 0.1, 4);
```

ARCH 686 | VISUAL STUDIO (C++)

CODING / AGGREGATION CATALOGUE
ITERATION (7)



```
// create crss3d mesh
//createMeshFromfile("data/cross3d.obj");
createMeshFromfile("data/Cindy_Ite.obj");
transformMesh(mat);

//----- trasnfromPartofMesh

//construct array with vertex ids of the vertices that need to be trasnfromed
//int src[] = { 0,1,2,3,4,5,6,7 };// cross 3d
int src[] = { 8,9,13,16,6,7,10,11,17,15,12,14,0,20,21,5,1,3,4,23,22,18,19,2 };
int n = sizeof(src) / sizeof(src[0]);
zIntArray ids(src, src + n);

// construct a gradient parameter
zVector u, v, w, c;
//attractor = zVector(3,3,3);

c = cen;
float maxL = diagLength;

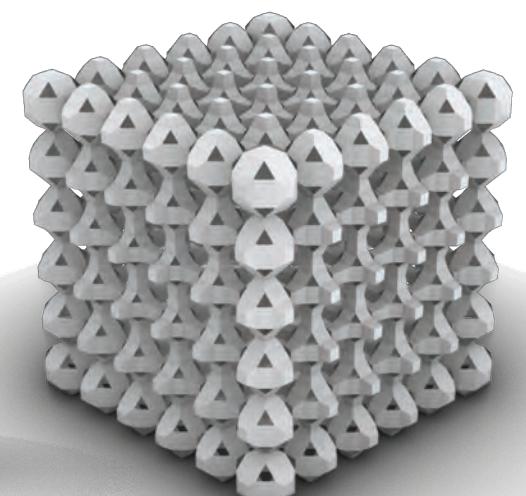
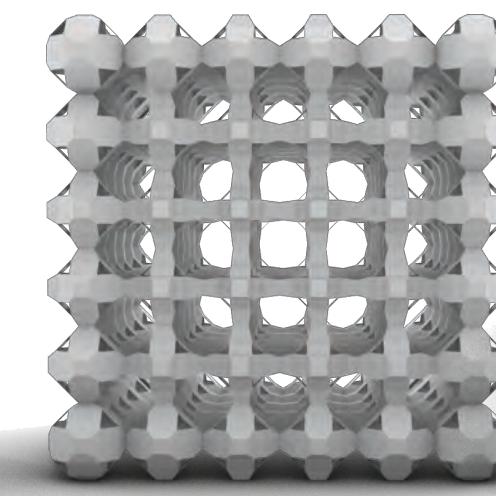
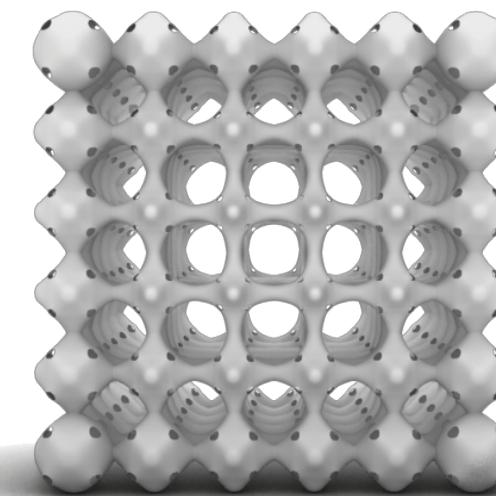
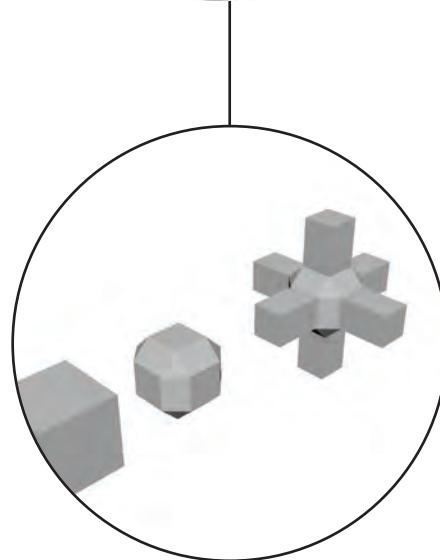
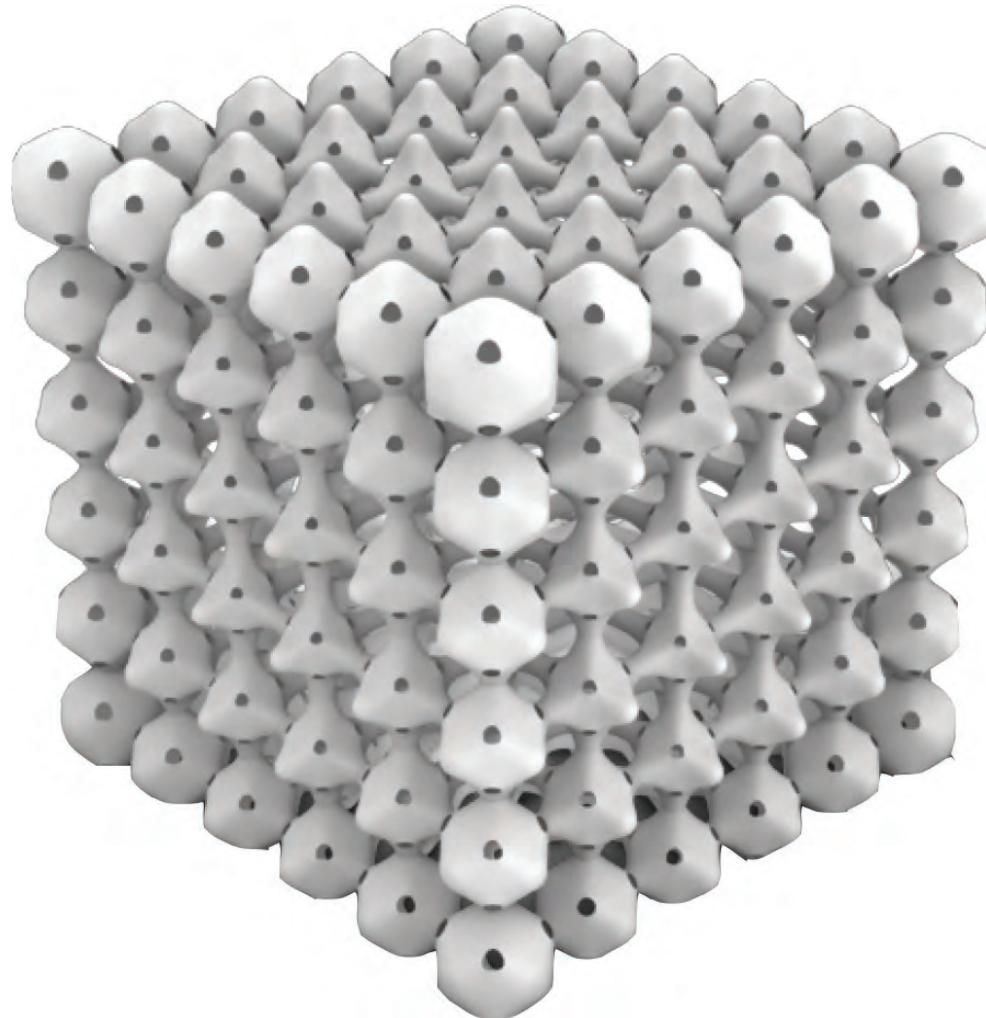
//float disAttractor = cen.distanceTo(attractor);

float mult = ofMap(id, 0, 215, 0.1, 2); // 4 - ofMap(cen.length(), 0, maxL, 0.1, 4);
```

ARCH 686 | VISUAL STUDIO (C++)

CODING / AGGREGATION CATALOGUE

ITERATION (7)



```
// create crss3d mesh
//createMeshFromfile("data/cross3d.obj");
createMeshFromfile("data/Cindy_Ite.obj");
transformMesh(mat);

//----- trasnfromPartofMesh

//construct array with vertex ids of the vertices that need to be trasnfromed
//int src[] = { 0, 1, 2, 3, 4, 5, 6, 7 };// cross 3d
int src[] = { 8, 9, 13, 16, 6, 7, 10, 11, 17, 15, 12, 14, 0, 20, 21, 5, 1, 3, 4, 23, 22, 18, 19, 2 };
int n = sizeof(src) / sizeof(src[0]);
zIntArray ids(src, src + n);

// construct a gradient parameter
zVector u, v, w, c, move, origin, attractor;
attractor = zVector(0, 0, 0);

c = cen;
float maxL = diagLength;

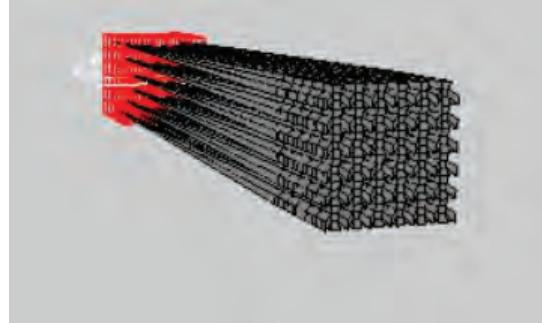
float disAttractor = cen.distanceTo(attractor);

float mult = ofMap(disAttractor, 0, 4, 0.01, 2); // 4 - ofMap(cen.length(), 0, maxL, 0.1, 4);
```

ARCH 686 | VISUAL STUDIO (C++)

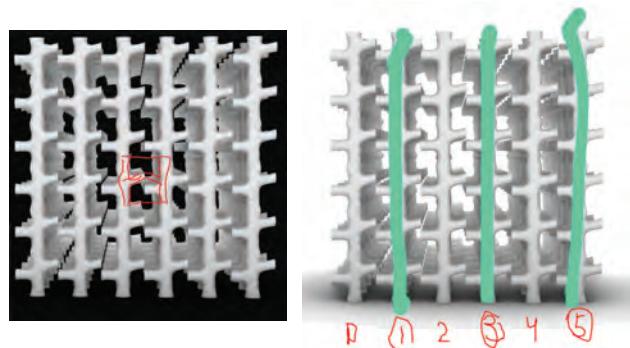
LESSONS LEARNED / TROUBLE SHOOT

PROBLEM



TROUBLESHOOT

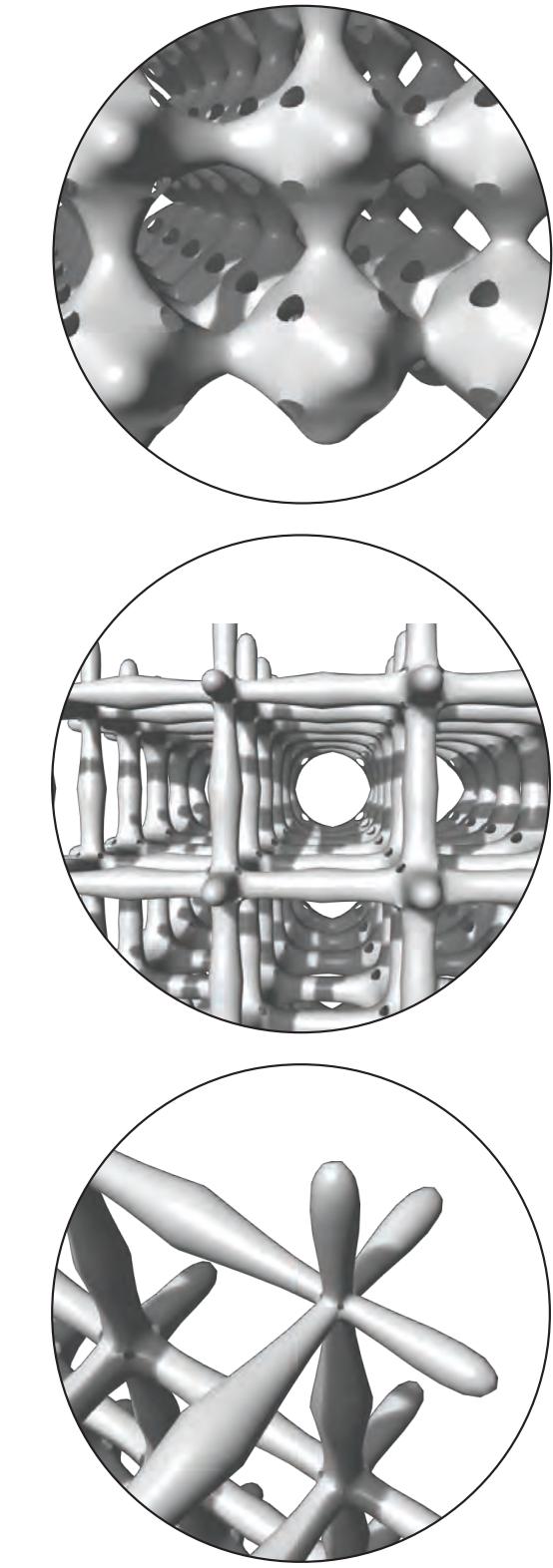
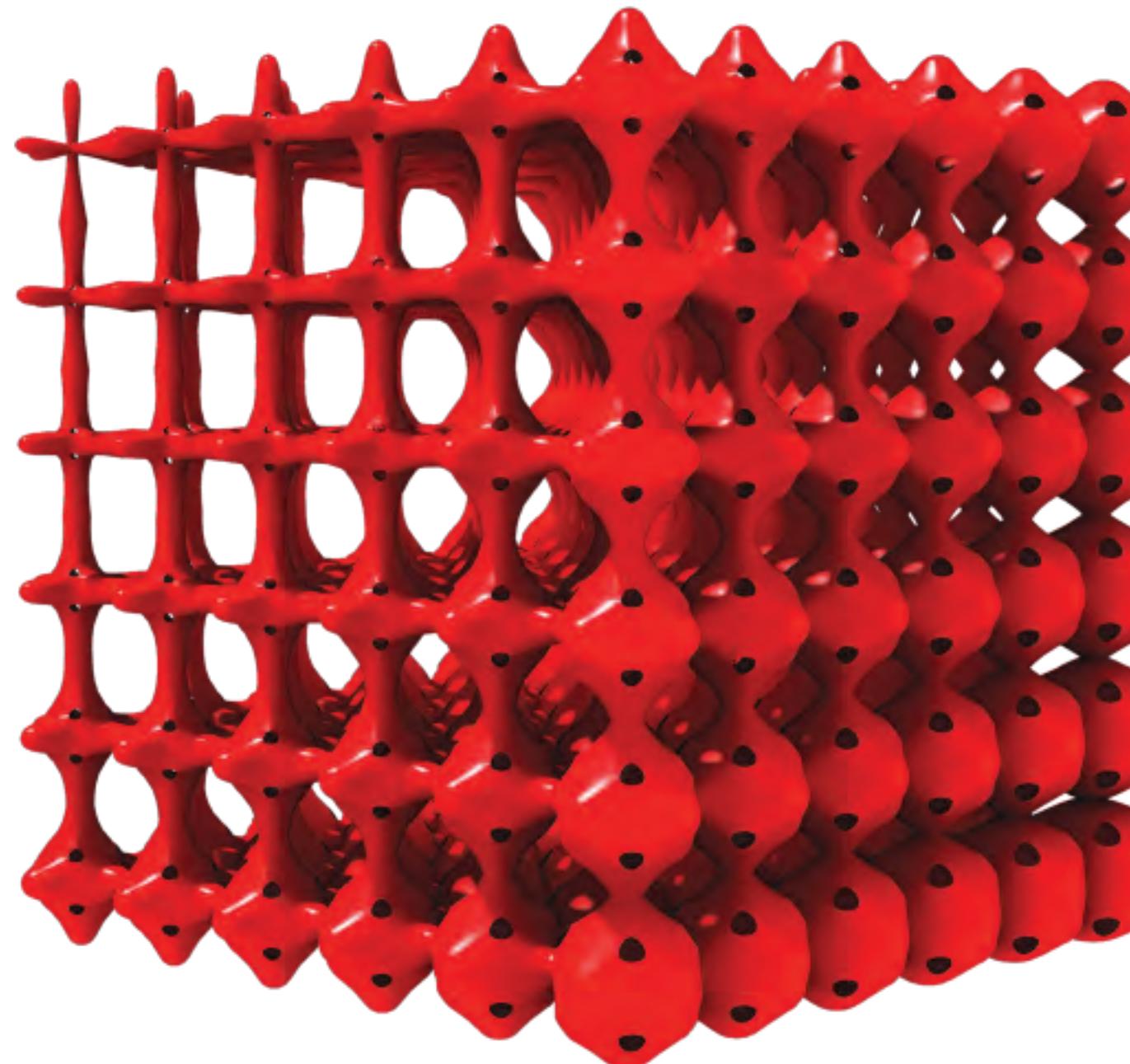
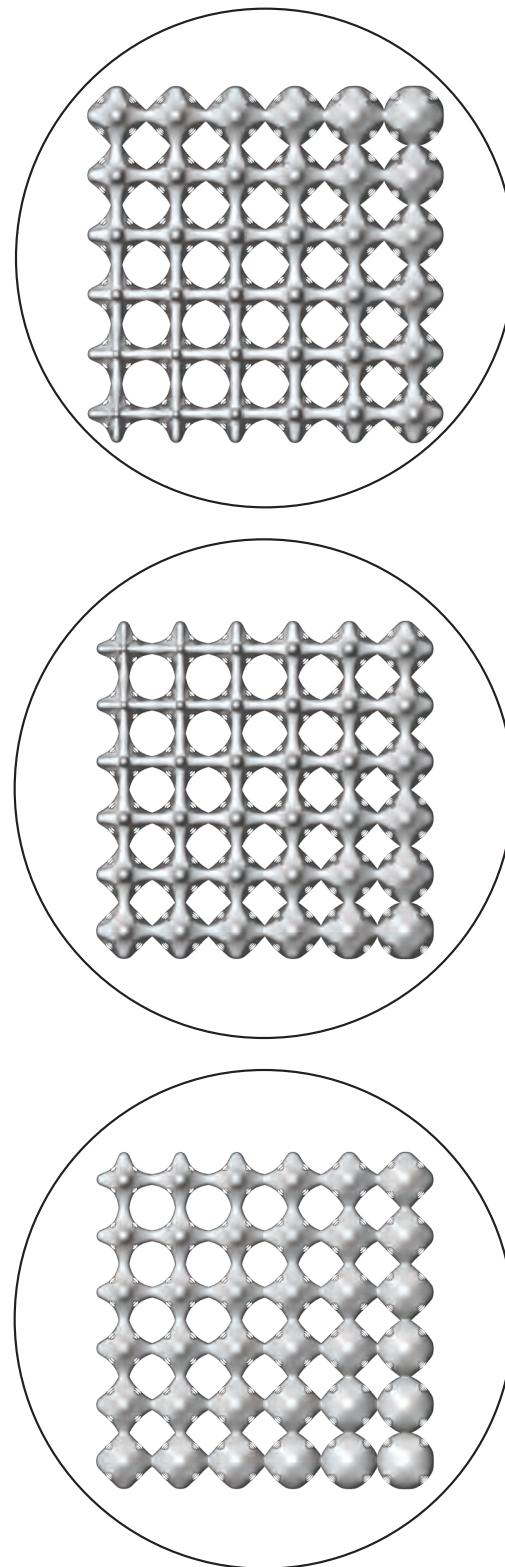
When export the initial obj to be aggregated, it should be at the origin in Maya (0,0,0)



- Identify the id of the rows (either working with multidimensional arrays or having a method that retrieves a row index based on a 1D array), if is an odd number ($id \% 2 = 0$) scale the x or y axis by -1
link to a reference on flattening / unflattening arrays here <https://stackoverflow.com/questions/7367770/how-to-flatten-or-index-3d-array-in-1d-array>.
- Design the component differently and maintain the middle area as the one you want to transform.

- The simpler the better. It is better not to exceed 6 faces in the base geometry.
- Iterations help the studies of geometry. Iterating is a productive way to do explorations.
- Creating a catalogue to document the process can provide a way to see the whole project from scratch to final production.
- Utilizing codes allows the modelling process to be much simpler than human-generated models and is able to save a lot of time. However, it is not the easiest job in the world to develop the code itself before handing the job to computers.
- Computational geometry is less precise but we could produce a series of geometry studies in a short amount of time.

ARCH 686 | VISUAL STUDIO (C++)
3D PRINTING CATALOGUE



ARCH 686 | VISUAL STUDIO (C++)

3D PRINTING CATALOGUE

```
void update( float diagLength = 2)
{
    zVector cen( mat(0, 3), mat(1, 3), mat(2, 3) );

    // create crss3d mesh
    //createMeshFromFile("data/cross3d.obj");
    createMeshFromFile("data/Cindy_Ite.obj");
    transformMesh(mat);

    //----- trasnfromPArtofMesh

    //construct array with vertex ids of the vertices that need to be
    trasnfomed
    //int src[] = { 0, 1, 2, 3, 4, 5, 6, 7 };// cross 3d
    int src[] = { 8, 9, 13, 16, 6, 7, 10, 11, 17, 15, 12, 14, 0, 20, 21, 5, 1, 3, 4, 23, 22, 18, 19, 2
    };
    int n = sizeof(src) / sizeof(src[0]);
    zIntArray ids(src, src + n);

    // construct a gradient parameter
    zVector u, v, w, c, move, origin, attractor;
    attractor = zVector(3, 3, 3);

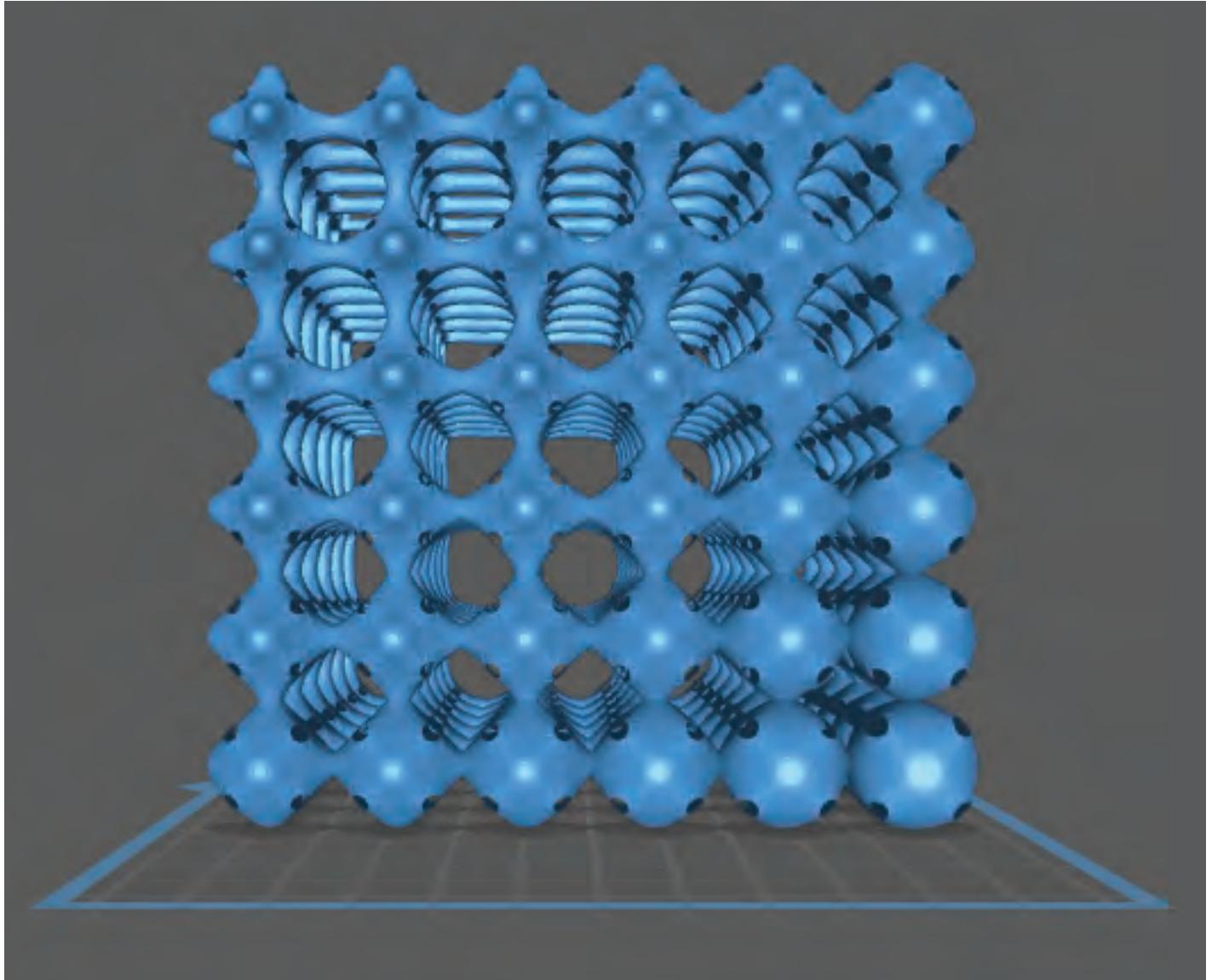
    c = cen;
    float maxL = diagLength;

    float disAttractor = cen.distanceTo(attractor);

    float mult = ofMap(disAttractor, 0, 8, 0.01, 2); // 4 -
    ofMap(cen.length(), 0, maxL, 0.1, 4);

    //cout << mult << "--mult" << endl;

    //construct the transformation matrix
    zTransform TM;
    TM.setIdentity();
    u = zVector(1, 0, 0);
    v = zVector(0, 1, 0);
    w = zVector(0, 0, 1);
    u.normalize(); v.normalize(); w.normalize();
    u *= mult; v *= mult; w *= mult;
```



ARCH 686 | VISUAL STUDIO (C++)

3D PRINTING CATALOGUE

