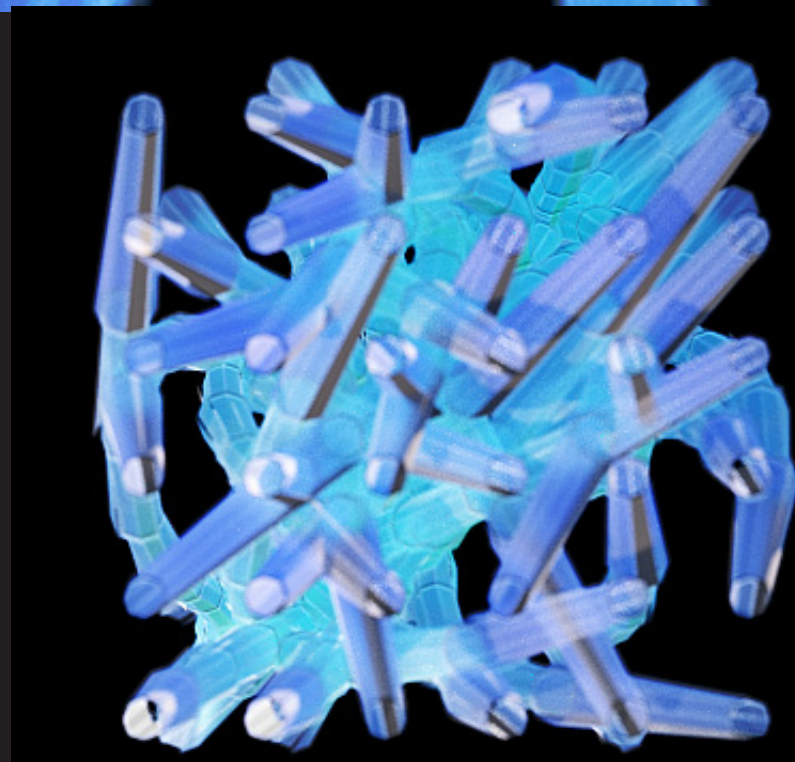
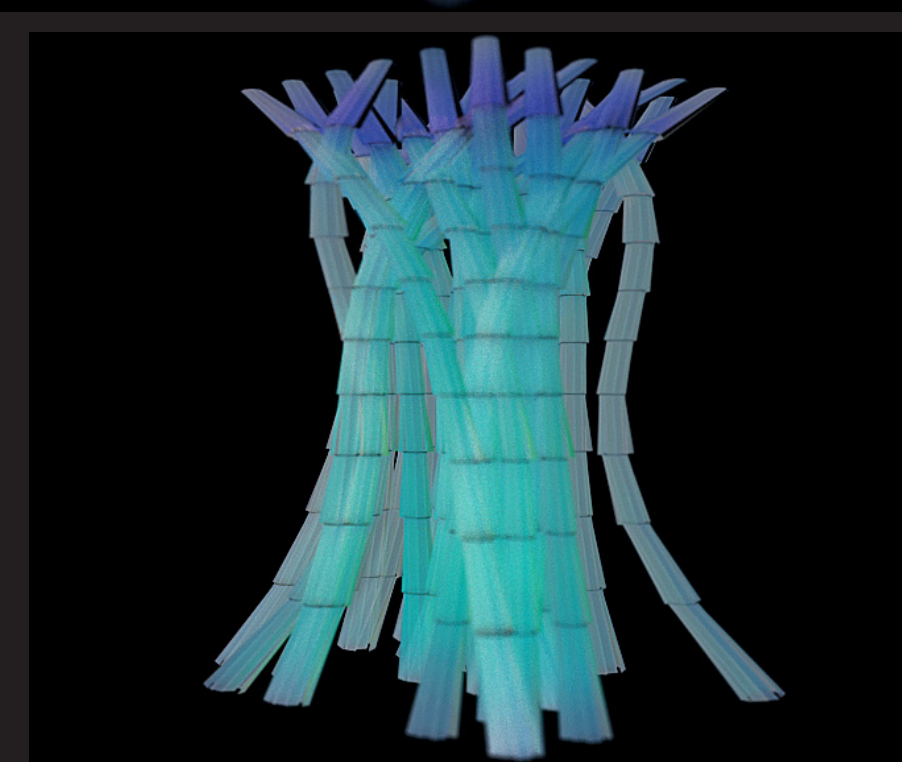


Detail



Plan



Elevation

```
import processing.opengl.*;
import igeo.p.*;
import igeo.io.*;
import igeo.geo.*;
import igeo.util.*;
import igeo.core.*;
import igeo.gui.*;

size(500,500,IG.GL);

//create 4 surfaces
ISurface surfA = new ISurface(-25, 0, 75, -25, 0, 75, 75, 0, -25, 75, 0);
ISurface surfB = new ISurface(0, 0, -50, 50, 0, -50, 50, 50, -50, 100, -50);
ISurface surfC = new ISurface(0, 0, -70, 50, 0, -70, 50, 50, -70, 0, 50, 70);
ISurface surfD = new ISurface(0, 0, -150, 50, 0, -150, 50, 50, -150, 0, 50, -150);
ISurface surfE = new ISurface(-40, -50, -150, 90, -50, -150, 90, 100, -150, -40, 100, -150);

//Random define
double randomA = IRandom.get(0.00, 0.49);
double randomB = IRandom.get(0.50, 1.00);

//Divide surf
int unum=6, vnum=6;
double uinc=1.0/unum, vinc=1.0/vnum;

//Treat SurfA
ICurve[] circlesA = new ICurve[unum][vnum];
for (int i=0; i<unum; i++) {
  for (int j=0; j<vnum; j++) {
    IVec pt1=surfA.pt((i+randomA)*uinc, (j+randomA)*vinc);
    IVec pt2=surfA.pt((i+randomB)*uinc, (j+randomA)*vinc);
    IVec pt3=surfA.pt((i+randomB)*uinc, (j+randomB)*vinc);
    IVec pt4=surfA.pt((i+randomA)*uinc, (j+randomB)*vinc);
    IVec[] pts = new IVec[4];
    pts[0] = pt1;
    pts[1] = pt2;
    pts[2] = pt3;
    pts[3] = pt4;
    ICurve profile = new ICurve(pts, 2, true).clr(1, 0, 0);
    ICurve flatProfile = IG.flatten(profile);
    circlesA[i][j] = flatProfile;
  }
}

//Treat SurfB
ICurve[] circlesB= new ICurve[unum][vnum];
for (int i=0; i<unum; i++) {
  for (int j=0; j<vnum; j++) {
    IVec pt1=surfB.pt((i+randomA)*uinc, (j+randomA)*vinc);
    IVec pt2=surfB.pt((i+randomB)*uinc, (j+randomA)*vinc);
    IVec pt3=surfB.pt((i+randomB)*uinc, (j+randomB)*vinc);
    IVec pt4=surfB.pt((i+randomA)*uinc, (j+randomB)*vinc);
    IVec[] pts = new IVec[4];
    pts[0] = pt1;
    pts[1] = pt2;
    pts[2] = pt3;
    pts[3] = pt4;
    ICurve profile = new ICurve(pts, 2, true).clr(1, 0, 0);
    ICurve flatProfile = IG.flatten(profile);
    circlesB[i][j] = flatProfile;
  }
}

//Treat SurfC
ICurve[] circlesC= new ICurve[unum][vnum];
for (int i=0; i<unum; i++) {
  for (int j=0; j<vnum; j++) {
    IVec pt1=surfC.pt((i+randomA)*uinc, (j+randomA)*vinc);
    IVec pt2=surfC.pt((i+randomB)*uinc, (j+randomA)*vinc);
    IVec pt3=surfC.pt((i+randomB)*uinc, (j+randomB)*vinc);
    IVec pt4=surfC.pt((i+randomA)*uinc, (j+randomB)*vinc);
    IVec[] pts = new IVec[4];
    pts[0] = pt1;
    pts[1] = pt2;
    pts[2] = pt3;
    pts[3] = pt4;
    ICurve profile = new ICurve(pts, 2, true).clr(1, 0, 0);
    ICurve flatProfile = IG.flatten(profile);
    circlesC[i][j] = flatProfile;
  }
}

//Treat SurfD
ICurve[] circlesD= new ICurve[unum][vnum];
for (int i=0; i<unum; i++) {
  for (int j=0; j<vnum; j++) {
    IVec pt1=surfD.pt((i+randomA)*uinc, (j+randomA)*vinc);
    IVec pt2=surfD.pt((i+randomB)*uinc, (j+randomA)*vinc);
    IVec pt3=surfD.pt((i+randomB)*uinc, (j+randomB)*vinc);
    IVec pt4=surfD.pt((i+randomA)*uinc, (j+randomB)*vinc);
    IVec[] pts = new IVec[4];
    pts[0] = pt1;
    pts[1] = pt2;
    pts[2] = pt3;
    pts[3] = pt4;
    ICurve profile = new ICurve(pts, 2, true).clr(1, 0, 0);
    ICurve flatProfile = IG.flatten(profile);
    circlesD[i][j] = flatProfile;
  }
}

//Random grid to and loft
for (int i=0; i<unum; i++) {
  for (int j=0; j<vnum; j++) {
    ICurve prof1 = circlesA[i][j];
    int rand1 = IRandom.getInt(0,unum-1);
    int rand2 = IRandom.getInt(0,vnum-1);
    IVec[] pts = new IVec[4];
    pts[0] = pt1;
    pts[1] = pt2;
    pts[2] = pt3;
    pts[3] = pt4;
    ICurve prof2 = circlesB[rand1][rand2];
    ICurve prof3 = circlesC[rand1][rand2];
    ICurve prof4 = circlesD[rand1][rand2];
    IG.loft(new ICurve[] {prof1, prof2, prof3, prof4}, 3);
    IG.loft(new ICurve[] {curve6, curve7, curve8 }, 2).clr(0.2);
  }
}
surfA.del();
surfB.del();
surfC.del();
surfD.del();
IG.save("mid_for_panelization.3dm");

new ISurface(pt11, pt21, pt12, pt22).clr(0.1*uinc, 0.1*vinc, 0.3);
IVec spt1 = surf.pt(1*uinc, 1*vinc);
IVec spt2 = surf.pt((i+1)*uinc, 1*vinc);
IVec spt3 = surf.pt(1*uinc, (j+1)*vinc);
IVec spt4 = surf.pt((i+1)*uinc, (j+1)*vinc);
IVec[] ptsa = new IVec[2];
ptsa[0] = spt1;
ptsa[1] = spt2;
ICurve outline1 = new ICurve (ptsa,1);
IG.squarePipe(outline1,2).clr(0.0,1.0,3);
```

```
import processing.opengl.*;
import igeo.p.*;
import igeo.io.*;
import igeo.geo.*;
import igeo.util.*;
import igeo.core.*;
import igeo.gui.*;

size(500,500,IG.GL);
IG.open("mid_for_panelization.3dm");
IG.open("mid_presen_rhino.3dm");

//Creates vectors on the surf connect to structure outside.
ISurface[] surfs = IG.surfaces();

IImageMap map = new IImageMap("gray_scale_map1.jpg");
for(ISurface surf : surfs) {
  int unum = 10, vnum = 8;
  double uinc = 1.0/unum, vinc = 1.0/vnum;
  for(int i=0; i < unum; i++) {
    for(int j=0; j < vnum; j++) {
      // IVec pt = surf.pt(1*uinc, 1*vinc);
      double val = map.get(1*uinc, 1*vinc);
      //Paneling the surface
      IVec pt11 = surf.pt(1*uinc, 1*vinc);
      IVec pt21 = surf.pt((i+1)*uinc, 1*vinc);
      IVec pt22 = surf.pt((i+1)*uinc, (j+1)*vinc);
      IVec pt12 = surf.pt(1*uinc, (j+1)*vinc);
      new ISurface(pt11, pt21, pt12, pt22).clr(0.1*uinc, 0.1*vinc, 0.3);
      IVec spt1 = surf.pt(1*uinc, 1*vinc);
      IVec spt2 = surf.pt((i+1)*uinc, 1*vinc);
      IVec spt3 = surf.pt(1*uinc, (j+1)*vinc);
      IVec spt4 = surf.pt((i+1)*uinc, (j+1)*vinc);
      IVec[] ptsa = new IVec[2];
      ptsa[0] = spt1;
      ptsa[1] = spt2;
      ICurve outline1 = new ICurve (ptsa,1);
      IG.squarePipe(outline1,2).clr(0.0,1.0,3);
    }
  }
}
```

